Fault-tolerant cycle embedding in dual-cube with node faults

Yamin Li* and Shietung Peng
Department of Computer Science,
Hosei University, Tokyo 184 8584, Japan
E-mail: yamin@k.hosei.ac.jp E-mail: speng@k.hosei.ac.jp
*Corresponding author

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

Abstract: A low-degree dual-cube was proposed as an alternative to the hypercubes. A dual-cube DC(m) has $m + 1$ links per node, where m is the degree of a cluster (m-cube) and one more link is used for connecting to a node in another cluster. There are 2^{m+1} clusters and hence the total number of nodes in a DC(m) is 2^{2m+1}. In this paper, by using Gray code, we show that there exists a fault-free cycle containing at least $2^{2m+1} - 2f$ nodes in DC(m), $m \geq 3$, with $f \leq m$ faulty nodes.

Keywords: interconnection networks; hypercube; Gray code; Hamiltonian cycle; fault-tolerant embedding.

Reference to this paper should be made as follows: Li, Y., Peng, S. and Chu, W. (2005) 'Fault-tolerant cycle embedding in dual-cube with node faults', *Int. J. High Performance Computing and Networking*, Vol. 3, No. 1, pp.45–53.

Biographical notes: Yamin Li received his PhD degree in computer science from the Tsinghua University in 1989. He is currently a professor in the Department of Computer Science at the Hosei University. His research interests include computer architecture, parallel and distributed computing, interconnection networks and fault-tolerant computing. Li is a senior member of the IEEE and a member of the IEEE Computer Society.

Shietung Peng received his PhD degree (1986) in computer science from the University of Texas, Dallas. He was a Faculty member at the University of Maryland, Baltimore County, and at the University of Aizu in Japan. He joined as the Faculty of computer and information science, at the Hosei University as a full Professor in 2000. His research interests are parallel and distributed processing, interconnection networks, fault-tolerant, and optical routing.

Wanming Chu is a Faculty member of Department of Computer Hardware, the University of Aizu, Japan. Her research interests include computer arithmetic algorithm and hardware implementation, multithreaded computer architecture, interconnection network, fault-tolerant computing, and performance evaluation.

1 INTRODUCTION

The binary hypercube has been widely used as the interconnection network in a wide variety of parallel systems such as Intel iPSC, the nCUBE (Hayes and Mudge, 1989), the Connection Machine CM-2 (Tucker and Robertson, 1988), and the SGI Origin 2000 (SGI, 1997). A hypercube network of dimension n, or n-cube, contains up to 2^n nodes and has n links per node. If unique n-bit binary addresses are assigned to the nodes of an n-cube, then a link connects two nodes if and only if their binary addresses differ in a single bit. Because of its elegant topological properties and the ability to emulate a wide variety of other frequently used interconnection networks, the hypercube has been one of the most popular interconnection networks for parallel computer systems.

However, the conventional hypercube has a major shortage, that is, the number of links per node in a system increases logarithmically as the number of nodes in the system increases. Since the number of links per node is limited to eight with current technology (SGI, 1997), the total number of nodes in a hypercube parallel computer is restricted to several hundreds. Therefore, it is interesting to develop an interconnection network, which keeps most of topological properties of the hypercube, and has more nodes in the system than the hypercube with the same number of links per node.

Several variations of the hypercube are proposed in the literature. Some variations focus on the reduction of diameter of the hypercube, such as folded hypercube (Amawy and Latifi, 1991) and crossed cube (Efe, 1992); some focus on the reduction of the number of links of the hypercube, such as cube-connected cycles (Preparata and Vuillemin, 1981) and reduced hypercube (Ziavras, 1994); and some focus on both, like hierarchical cubic network (Ghose and Desai, 1995). Generally, the variations of the hypercube that reduce the diameter, e.g., crossed cube and hierarchical cubic network, does not satisfy the following key property in the hypercube: each node can be represented by a unique binary number such that two nodes are connected by a link only if the two binary numbers differ in one bit position. This key property is at the core of many algorithmic designs for efficient routing and communications.

A new interconnection network for large parallel systems called *dual-cube* (DC) has been proposed recently (Li and Peng, 2000; Li et al., 2001). The dual-cube shares the desired properties of the hypercube (e.g., the key property of the hypercube mentioned above), and increases tremendously the total number of nodes in the system compared to the hypercube with the same number of links per node. The size of the dual-cube can be as large as 30,000 with up to eight links per node. It is practically important to refine the hypercube networks, such that the size of the network can be increased, while the number of the links per node is limited by the technology. Wu and Wu (2003) discussed some consideration on efficient VLSI layout design of the dual-cube router and Jiang and Wu (2003) provided a fault-tolerant routing in dual-cube networks.

One of the most interesting properties of the N-node hypercube is that it contains every N-node ring as a sub-graph (i.e., the hypercube is Hamiltonian). Since some parallel applications such as those in image and signal processing are originally designated on a ring or Torus architecture, it is important to have effective ring embedding in a dual-cube. In a large parallel computer system, node failure is inevitable. Therefore, the issue of embedding a cycle of maximum length in a dual-cube with faulty nodes is critical for the dual-cube to be a practical one.

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 Hamiltonian path can be obtained from a Hamiltonian cycle by removing any one link from that cycle. A graph that contains a Hamiltonian cycle is said to be *Hamiltonian.* G is *n-link Hamiltonian* if it remains Hamiltonian after removing any n links. It is clear that if graph G is n-connected then G can be at most $(n - 2)$-link Hamiltonian.

Previous results about fault-tolerant cycle embedding in networks are as follows. The n-cube is $(n - 2)$-link Hamiltonian (Latifi et al., 1992). The n-dimensional folded hypercube is $(n - 1)$-link Hamiltonian (Wang, 2001). The n-dimensional star graph is $(n - 3)$-link Hamiltonian (Tseng et al., 1997). A k-ary undirected de Bruijn graph is $(k - 1)$-link Hamiltonian (Rowley and Bose, 1994). An $(m + 1)$-connected dual-cube DC(m) is $(m - 1)$-link Hamiltonian (Li et al., 2002).

The problem of faulty-node tolerant cycle embedding is to find a cycle in a network with some faulty nodes. The cycle length depends on the number of faulty nodes. For example, an n-cube with f faulty nodes can embed a fault-free cycle containing at least $2^n - 2f$ nodes, where $f \leq n - 1$ and $n \geq 3$ (Tseng, 1996). An n-dimensional star graph with f faulty nodes can embed a fault-free cycle containing at least $n! - 2f$ nodes, where $f \leq n - 3$ (Hsieh et al., 1998). A d-ary n-dimensional undirected de Bruijn graph with f faulty nodes can embed a fault-free cycle containing at least $d^n - nf - 1$ nodes, where $f \leq d - 1$ (Rowley and Bose, 1994). This paper shows that a dual-cube DC(m) with f faulty nodes can embed a fault-free cycle containing at least $2^{2m+1} - 2f$ nodes, where $f \leq m$ and $m \geq 3$.

The rest of this paper is organised as follows. Section 2 describes the dual-cube architecture. Section 3 constructs a Hamiltonian cycle in a DC(m). Section 4 shows that there exists a fault-free cycle containing at least $2^{2m+1} - 2f$ nodes in a DC(m) with $f \leq m$ and $m \geq 3$. Section 5 concludes the paper and presents some future research directions.

2 DUAL-CUBE ARCHITECTURE

A dual-cube uses hypercubes as basic components. Each hypercube component is referred to as a *cluster.* Assume that the number of nodes in a cluster is 2^m. In a dual-cube, there are two *classes* with each class consisting of 2^m clusters. The total number of nodes in a dual-cube DC(m) is $2^m \times 2^m \times 2$, or 2^{2m+1}. Each node in a DC(m) has $m + 1$ links: m links are used within cluster to construct an m-cube and a single link is used to connect a node in a cluster of the other class. There is no link between the clusters of the same class. If two nodes are in one cluster, or in two clusters of distinct classes, the distance between the two nodes is equal to its *Hamming distance* (the number of bits where the addresses of the two nodes have different values). Otherwise, it is equal to the Hamming distance plus two: one for entering a cluster of the other class and one for leaving.

An $(m + 1)$-connected dual-cube DC(m) is an undirected graph on the node set $\{0, 1\}^{2m+1}$ and there is a link between two nodes $u = (u_{2m} \ldots u_0)$ and $v = (v_{2m} \ldots v_0)$ if and only if the following conditions are satisfied:

- u and v differ exactly in one bit position i
- if $0 \le i \le m - 1$ then $u_{2m} = v_{2m} = 0$
- if $m \le i \le 2m - 1$ then $u_{2m} = v_{2m} = 1$.

Intuitively, the set of nodes u of form $(0u_{2m-1} \ldots u_m{*} \ldots {*})$, where * means 'do not care', constitutes an m-dimensional hypercube. We call these hypercubes clusters of class 0. Similarly, the set of nodes u of form $(1{*} \ldots {*}u_{m-1} \ldots u_0)$ constitutes an m-dimensional hypercube and we call them clusters of class 1. The link connecting two nodes in two clusters of distinct classes is called *cross-link*. In the other words, $e = (u : v)$ is a cross-link if and only if u and v differ in the leftmost bit position.

Each node in a DC(m) is identified by a unique $(2m + 1)$-bit number, an *id*. Each *id* contains three parts:

- 1-bit *class_ id*
- m-bit *cluster_id*
- m-bit *node_id*.

In the following discussion, we use *id* = (*class_id*, *cluster_id*, *node_id*) to denote the node address. The bit-position of *cluster_id* and *node_id* depends on the value of *class_id*. If *class_id* = 0 (*class_id* = 1), then *node_id* (*cluster_id*) is the rightmost m bits and *cluster_id* (*node_id*) is the next (to the left) m bits.

Figure 1 depicts a DC(2) network. In each node, *class_id* is shown at the top position. For the nodes of class 0 (class 1), *node_id* (*cluster_id*) is shown at the bottom and *cluster_id* (*node_id*) is shown at the middle. Figure 2 shows a DC(3). Notice that only those cross-links connecting to cluster 0 of class 1 are shown, the other cross-links are omitted for simplicity.

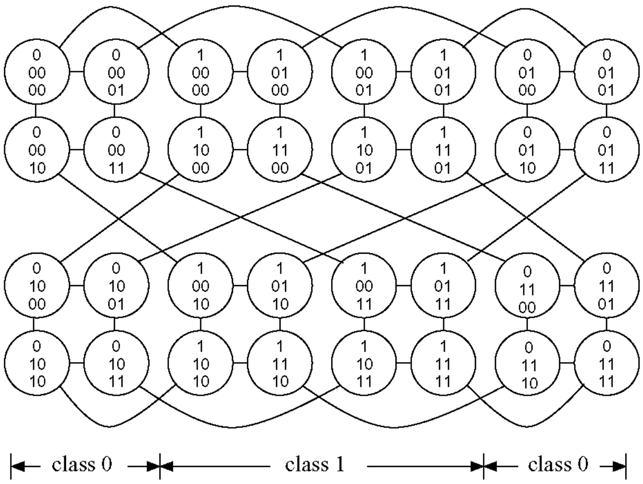

Figure 1 *A dual-cube DC(2)*

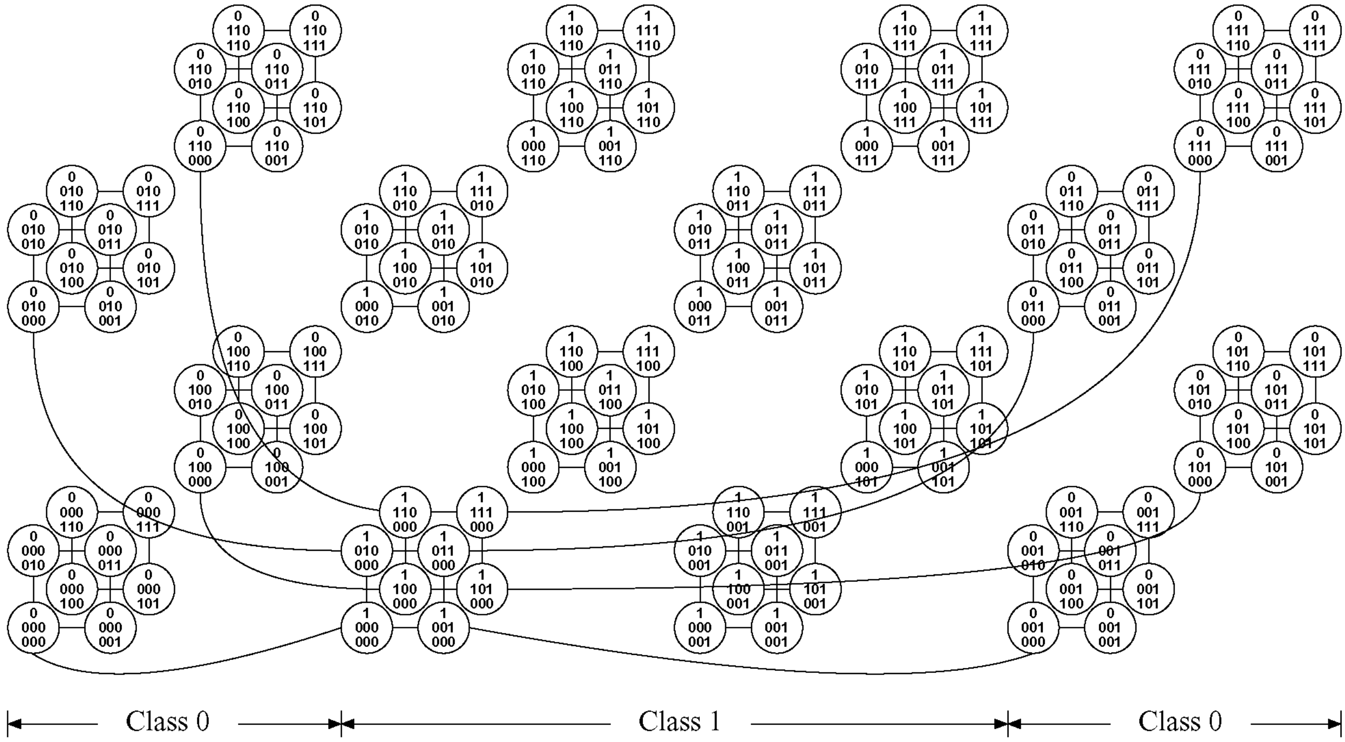

Figure 2 *A dual-cube DC(3)*

The dual-cube has a binary presentation of nodes, similar to a hypercube, in which two nodes are connected by a link only if their addresses differ in one bit position. This feature is the key for designing efficient routing and communication algorithms in dual-cube. Another important feature of a dual-cube is that, within the given bound to the number of links per node, say $m + 1$, the network can have up to 2^{2m+1} nodes. The DC(m) topological properties are given in Li and Peng (2000) and the collective communication schemes in DC(m) can be found in Li et al. (2001).

3 HAMILTONIAN CYCLE IN DUAL-CUBE

In Li et al. (2002), it was proved that the dual-cube is ($m - 1$)-link Hamiltonian. That is, if a DC(m) contains $m - 1$ faulty links, there exists a cycle that contains all the nodes. In this section, we show how to construct Hamiltonian cycles in dual-cube because it is needed for fault-tolerant cycle embedding in dual-cube with faulty nodes.

The key for constructing a Hamiltonian cycle in a DC(m) is to construct a *virtual Hamiltonian cycle* that connects all 2^{m+1} clusters in DC(m). The virtual Hamiltonian cycle in a DC(m) contains equal numbers of cube-links and cross-links; the cube-links and the cross-links are inter-leaved. To construct a fault-free Hamiltonian cycle in a DC(m) with up to $m - 1$ faulty links, we need to put some constraints on the cube-links in the virtual Hamiltonian cycle since a Hamiltonian path inside a cluster with faulty links might have fixed end nodes.

We use $0^{\{i\}}$ to denote a bit pattern $0 \ldots 0$ of i bits. The Hamiltonian cycle in an n-cube can be constructed by the *binary reflected Gray code.* A *Gray code* for binary numbers is a list of all n-bit numbers so that two consecutive numbers, including the first and last, differ in exactly one bit position. The best known example of the Gray codes is the *binary reflected Gray code*, $P(n)$, which can be described as follows. $P(1) = (0,1)$. For $n > 1$, $P(n)$ is formed by taking the list for $P(n-1)$ with each number prefixed by 0, then following that list by the reverse of $P(n-1)$ with each number prefixed by 1. For example, $P(2) = (00, 01, 11, 10)$, $P(3) = (000, 001, 011, 010, 110, 111, 101, 100)$, and so on. Since the first and last numbers of $P(n)$ also differ in one bit position, the code is in fact a cycle. For an n-cube, $P(n)$ contains all nodes and each node appears exactly once, a Hamiltonian cycle is formed.

Let $D(n)$ denote the list of dimensions (bit positions) on which the binary numbers in the reflected Gray code change their values. Then, $D(1) = 0$. For $n > 1$, $D(n)$ can be constructed recursively as follows: $D(n) = (D(n-1), n-1, D(n-1))$. For example, $D(2) = (0, 1, 0)$, $D(3) = (0, 1, 0, 2, 0, 1, 0)$, $D(4) = (0, 1, 0, 2, 0, 1, 0, 3, 0, 1, 0, 2, 0, 1, 0)$, and so on. We call $D(n)$ a *reflected dimension list.*

In what follows, we use $(u \to v)$ or $(u: \ldots : v)$ to denote a path. The following algorithm generates a Hamiltonian cycle P in an n-cube using the reflected dimension list. The $\oplus$ does bit-wise exclusive OR operation.

Algorithm 1 (cubeHC(n))

```
begin        /* build a hamiltonian cycle P in an n-cube */
    D(n) = DL(n);      /* D(n): reflected dimension list */
    w = 0;                        /* starting from node 0 */
    P = w;                     /* P is the hamiltonian cycle */
    for each dimension number i in D(n) do
        w = w ⊕ 2^i;                  /* find the next node */
        P = P : w;                   /* add the node into P */
    endfor
end
Procedure DL(n)
begin /* build a reflected dimension list for an n-cube */
    if (n == 1) return (0);
    else return (DL(n − 1), n − 1, DL(n − 1));
end
```

Note that the reflected Gray code or reflected dimension list is just one of the Gray codes. By renumbering the node numbers (exchanging bit positions of all node numbers), we have $n!$ different Gray code sequences. Furthermore, since there are 2^n links in the cycle, breaking down a different link will get a different path: there are $2^n n!$ Hamiltonian paths with different patterns in an n-cube.

Next, we add a condition to let a Hamiltonian cycle contain a given link. This is needed for constructing a fault-free Hamiltonian cycle in a dual-cube with faulty links.

Lemma 1: *Given any link $e = (u : v)$ in an n-cube, where u and v are two distinct nodes, there is a Hamiltonian cycle going through e.*

Proof: The lemma can be proved by renumbering every node in the n-cube with a mapping function $f(x)$ so that $u' = f(u) = 0^{\{n-1\}}0$ and $v' = f(v) = 0^{\{n-1\}}1$. Then a Hamiltonian cycle P can be built by Algorithm 1 with the new numbers. The Hamiltonian cycle denoted with the original node numbers can be obtained by applying $f^{-1}(x)$ to every node number in P, where $f^{-1}(x)$ is the reverse of function $f(x)$, i.e., $u = f^{-1}(u')$ and $v = f^{-1}(v')$. One possible $f(x)$ is bit-wise exclusive OR operation with u on every node number so that node u will have a new number $0^{\{n-1\}}0$, and then exchanges bit positions so that the node v will have a new number $0^{\{n-1\}}1$.

By removing $e = (u : v)$ from the Hamiltonian cycle constructed by Lemma 1, we get a Hamiltonian path from node u to node v, $(u \to v)$. We name the procedure that generates such a path as cubeHP(m, u, v).

A *virtual Hamiltonian cycle*, $V(m)$, is a cycle in DC(m) which connects all clusters and contains only two neighbouring nodes, u and v for instance, in each cluster (Figure 3). It is called to be *virtual* since the cube-link $e = (u : v)$ in the cycle will be replaced with a Hamiltonian path $(u \to v)$ in that cluster to form a Hamiltonian cycle in DC(m).

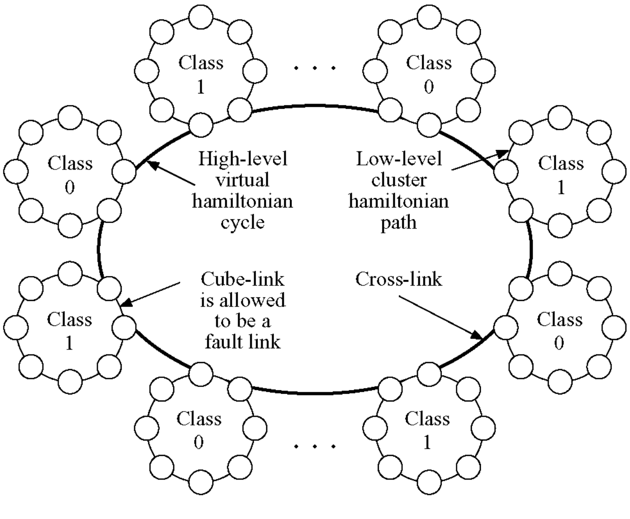

Figure 3 *Virtual Hamiltonian cycle*

The construction of the virtual Hamiltonian cycle can be done by using an *extended double-dimension list*, or $EDD(m)$, defined as follows. Let the reflected double-dimension list be $DD(m) = (DD(m-1), m-1, m-1, DD(m-1))$ if $m > 1$, and $DD(1) = (0,0)$. Then the extended double-dimension list $EDD(m) = (DD(m), m-1, m-1)$. Since there are two classes in a DC(m), $EDD(m)$ doubles each dimension number in an extended list, which consists of $D(m)$ plus the highest dimension $m-1$. For example, $EDD(2) = (0,0,1,1,0,0,1,1)$, $EDD(3) = (0,0,1,1,0,0,2,2,0,0,1,1,0,0,2,2)$, and so on. Then the virtual Hamiltonian cycle can be constructed with $EDD(m)$. For example, $V(2)$ = (00000 : 00001 : 10001 : 10101 : 00101 : 00111 : 10111 : 11111 : 01111 : 01110 : 11110 : 11010 : 01010 : 01000 : 11000 : 10000).

This virtual Hamiltonian cycle could be viewed as a high-level cycle, which connects all the clusters. Because there are two classes in a DC(m) and each class has 2^m clusters, the virtual Hamiltonian cycle contains $2^m \times 2 \times 2$, or $2^m \times 4$ nodes. If we group four nodes, whose rightmost m bits of the addresses are the same (e.g., 00001, 10001, 10101, 00101), into a *big* node, the virtual Hamiltonian cycle contains 2^m big nodes. Therefore, the algorithm to construct the virtual Hamiltonian cycle is similar to that of the hypercube. The difference is that once the next node in a cluster of a class is chosen, we need to go through the cross-link to a cluster of the other class. This is the reason why $DD(m)$ doubles each dimension number of $D(m)$. Algorithm 2 shows how to build a Hamiltonian cycle in a DC(m) and hence we have.

Theorem 1: *There is a Hamiltonian cycle in a dual-cube.*

Algorithm 2 (dualcubeHC(m))

```
begin            /* build a hamiltonian cycle P in DC(m) */
    DD(m) = DDL(m);
    EDD(m) = (DD(m), m − 1, m − 1);
    P = ∅;
    u = 0;
    for each dimension number i in EDD(m) do
        if (u is of class 0)   v = u ⊕ 2^i;
        else   v = u ⊕ 2^(m+i);
        P′ = cubeHP(m, u, v);          /* hamiltonian path */
        P = P : P′;
        u = v ⊕ 2^(2m);               /* go through cross-link */
    endfor
end
Procedure DDL(m)
begin        /* build a double-dimension list for a DC(m) */
    if (m == 1) return (0, 0);
    else return (DDL(m − 1), m − 1, m − 1, DDL(m − 1));
end
```

Lemma 2: *Given any cube-link e = (u:v) in a DC(m), there is a virtual Hamiltonian cycle going through e.*

Proof: Similar to the proof of Lemma 1.

Since there are $m2^{m-1}$ links in a cluster, by taking each of the links as link (u : v), we have $m2^{m-1}$ different virtual Hamiltonian cycles. These cycles are different but not disjointed.

Theorem 2: *There are 2^{m-1} disjoint virtual Hamiltonian cycles in a DC(m).*

Proof: We use induction to prove the theorem. For $m = 2$, two links in a cluster, for example e_0 = (00000 : 00001) and e_1 = (00011 : 00010) in the cluster 0 of class 0, connect four distinct nodes in the cluster. By Lemma 2, we can build two virtual Hamiltonian cycles going through e_0 and e_1, respectively. The two cycles are disjoint as shown as in Figure 4. The two cycles are constructed by $EDD(2)$, based on the reflected dimension list $D(2) = (0,1,0)$ with the starting nodes 00000 and 00011, respectively.

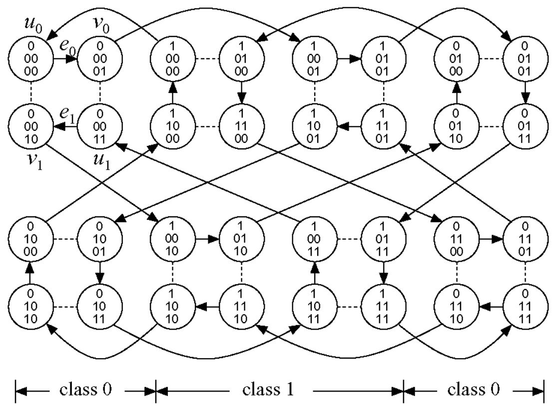

Figure 4 *Disjoint virtual Hamiltonian cycles*

Generally, there are $2^m/2$ such links in an m-dimensional cluster (m-cube): each link takes two nodes from the list of the reflected Gray codes. For $m > 2$, the $2^m/2$ virtual Hamiltonian cycles that contain e_i, $0 \le i \le 2^{m-1} - 1$, can be built based on the reflected dimension list $D(m)$ and Lemma 2. Because $D(m) = (D(m-1), m-1, D(m-1))$, by our induction hypothesis, the first half of all the cycles are disjointed. Then, all the paths that go through the $(m-1)$th dimension will still be disjoint. Similarly, the second half of all the cycles is also disjointed. Therefore, all 2^{m-1} cycles are disjoint.

Corollary 1: *Given a cycle of length n in a cluster of a dual-cube, there are n/2 disjoint virtual Hamiltonian cycles that go through a link in the given cycle.*

4 FAULT-FREE CYCLE EMBEDDING IN DUAL-CUBE WITH FAULTY NODES

In this section, we consider the problem of finding fault-free cycle of maximal length in dual-cube with faulty nodes. The following lemmas on hypercube are needed.

Lemma 3: *Given two links $e_0 = (u_0 : v_0)$ and $e_1 = (u_1 : v_1)$ in an n-cube, there is a Hamiltonian cycle going through e_0 and e_1.*

Proof: We use induction on n to prove the lemma. For $n = 2$ (a four-node ring), the lemma is true. We assume that the lemma holds for $n = k \ge 2$. Dividing an n-cube along with any dimension we can get two $(n-1)$-cubes, namely subcube0 and subcube1, respectively. For $n = k + 1$, because there are $n \ge 3$ dimensions, we can divide the n-cube along with a dimension so that the two links e_0 and e_1 are in subcube0 and/or subcube1.

If e_0 and e_1 are in a same sub-cube, subcube0 for instance, by our induction hypothesis, there is a Hamiltonian cycle going through e_0 and e_1. Select a link $(x : y)$ other than e_0 and e_1, by Lemma 1, there is a Hamiltonian cycle going through $(x' : y')^1$ in subcube1. Replacing $(x : y)$ and $(x' : y')$ with $(x : x')$ and $(y : y')$, a Hamiltonian cycle going through e_0 and e1 is obtained.

If e_0 and e_1 are in different sub-cubes, say, e_0 is in subcube0 and e_1 is in subcube1, by Lemma 1, a Hamiltonian cycle going through e_0 in subcube0 can be built. Select a link $(x : y)$ so that $(x : y) \ne e_0$ and $(x' : y') \ne e_1$, by our induction hypothesis, there is a Hamiltonian cycle going through $(x' : y')$ and e_1 in subcube1. Replacing $(x : y)$ and $(x' : y')$ with $(x : x')$ and $(y : y')$, a Hamiltonian cycle going through e_0 and e_1 is obtained.

Lemma 4: *Given three links $e_0 = (u_0 : v_0)$, $e_1^0 = (u_1 : w)$, and $e_1^1 = (w : v_1)$ in an n-cube, where $w \ne u_0$ and $w \ne v_0$, there is a Hamiltonian cycle going through e_0, e_1^0, and e_1^1.*

Proof: We use induction on n to prove the lemma. For $n = 3$, the lemma is true as shown as in Figure 5. Without loss of generality, let $u_0 = 0$ and $v_0 = 4$. For $w = 1$, all three link patterns are shown in Figure 5(a), (b), and (c), respectively. The case of $w = 2$, 5, or 6 is similar to the case of $w = 1$. For $w = 3$, all three link patterns are shown in the figure, and the case of $w = 7$ is similar to the case of $w = 3$.

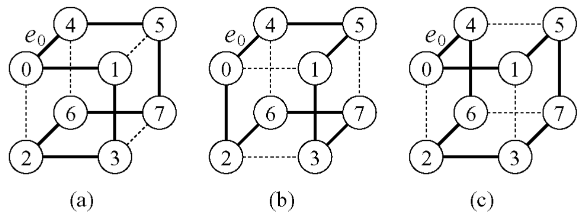

Figure 5 *Three links in Hamiltonian cycle in three-cube*

We assume that the lemma holds for $n = k \ge 3$. For $n = k + 1$, because there are $n \ge 4$ dimensions, we can divide the n-cube along with a dimension so that the three links e_0, e_1^0 and e_1^1 are in subcube0 and/or subcube1. Note that e_1^0 and e_1^1 are in a same sub-cube. Assume that e_0 is in subcube0.

If e_1^0 and e_1^1 are in subcube0, by our induction hypothesis, there is a Hamiltonian cycle going through e_0, e_1^0 and e_1^1. Select a link $(x : y)$ other than e_0, e_1^0 and e_1^1, by Lemma 1, there is a Hamiltonian cycle going through $(x' : y')$ in subcube1. Replacing $(x : y)$ and $(x' : y')$ with $(x : x')$ and $(y : y')$, a Hamiltonian cycle going through e_0, e_1^0 and e_1^1 is obtained.

If e_1^0 and e_1^1 are in subcube1, by Lemma 1, a Hamiltonian cycle going through e_0 in subcube0 can be built. Select a link $(x : y)$ other than e_0, so that $(x' : y') \ne e_1^0$ and $(x' : y') \ne e_1^1$, by our induction hypothesis, there is a Hamiltonian cycle going through $(x' : y')$, e_1^0 and e_1^1 in subcube1. Replacing $(x : y)$ and $(x' : y')$ with $(x : x')$ and $(y : y')$, a Hamiltonian cycle going through e_0, e_1^0 and e_1^1 is obtained.

Lemma 5: *Given a link $e = (u : v)$ in an n-cube with $f \le n - 2$ faulty nodes, where u and v are two non-faulty nodes, where u and v are two non-faulty nodes, there is a fault-free cycle that contains at lease $2^n - 2f$ nodes and goes through link e.*

Proof: We use induction on n to prove the lemma. For $n = 3$, the lemma is true as shown as in Figure 6, where $u = 3$ and $v = 7$. The figure shows the case of node 0 faulty. The case of node 4 faulty is similar. If node 1 is faulty, the cycle is the same as in the figure and the case in which node 2, 5, or 6 is faulty is similar to the case of node 1 faulty.

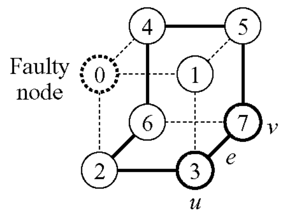

Figure 6 *Fault-free cycle in three-cubes*

We assume that the lemma holds for $n = k \geq 3$. For $n = k + 1$, without loss of generality, assume that e belongs to subcube0. Let f_0 and f_1 be the numbers of faulty nodes in subcube0 and subcube1, respectively, where $f_0 + f_1 = f \leq k - 1$. The proof of the lemma is divided into three cases.

Case 1: $f_0 < f$ and $f_1 < f$. By our induction hypothesis, there is a fault-free cycle C_0 containing at least $2^k - 2f_0$ nodes going through link e in subcube0. Mapping C_0 onto subcube1, we get C_0'. Because a faulty node in subcube1 can block at most two links in C_0', and $2^k - 2f_0 - 2f_1 = 2^k - 2f > 2^k - 2(k - 1 = 2^k - 2k + 2 > 0$, there exists a link $(x : y)$ in C_0 such that the nodes x' and y' in subcube1 are not faulty. By our induction hypothesis, in subcube1, there is a fault-free cycle C_1 that contains at least $2^k - 2f_1$ nodes and goes through link $(x' : y')$. By replacing the links $(x : y)$ and $(x' : y')$ with the links $(x : x')$ and $(y : y')$, respectively, a fault-free cycle can be built that contains at least $(2^k - 2f_0) + (2^k - 2f_1) = 2^{k+1} - 2f$ nodes and goes through link e in $(k + 1)$-cube (Figure 7).

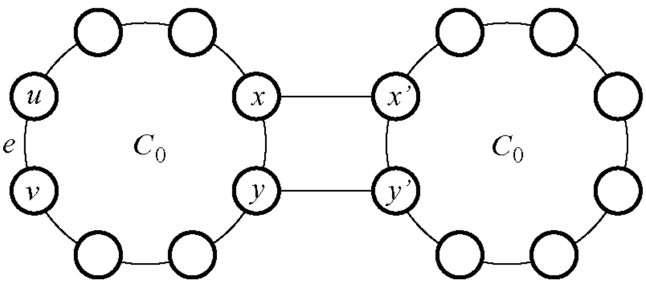

Figure 7 *Fault-free cycle in n-cube (case 1)*

Case 2: $f_0 = f$. Let w be a faulty node. Mark w non-faulty. By our induction hypothesis, there is a fault-free cycle C_0 containing at least $2^k - 2(f - 1)$ nodes and going through link e. Suppose that w appears in C_0. Let x and y be the two neighbours of w in C_0. Because no faulty node exists in subcube1, by Lemma 3, there is a Hamiltonian cycle C_1 of length 2^k that goes through links $(x' : w')$ and $(w' : y')$. By replacing the links $(x : w)$, $(w : y)$, $(x' : w')$, and $(w' : y')$ with the links $(x : x')$ and $(y : y')$, a fault-free cycle can be built that contains at least $(2^k - 2(f - 1) - 1 + (2^k - 1) = 2^{k+1} - 2f$ nodes and goes through link e in $(k + 1)$-cube (Figure 8).

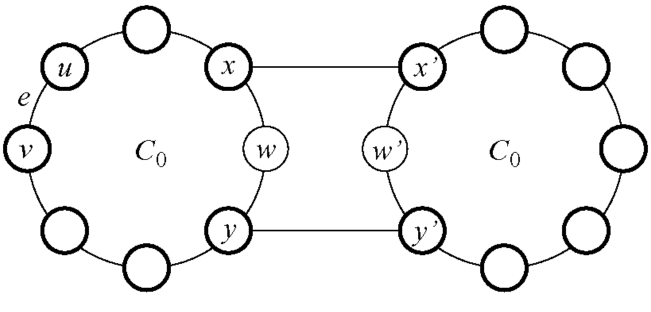

Figure 8 *Fault-free cycle in n-cube (case 2)*

Case 3: $f_1 = f$. Note that we can select a dimension to divide the n-cube so that both u' and v' in subcube1 are not faulty. Because, if u' or v' is faulty in a partition, we can re-divide the n-cube so that at least one of u' and v' is in subcube0 and apply the proof of case 1 or case 2.

Let w' be a faulty node in subcube1. Let us mark w' as non-faulty. By our induction hypothesis, there is a fault-free cycle C_1 containing at least $2^k - 2(f - 1)$ nodes. Suppose that w' appears in C_1. Let x' and y' be the two neighbours of w' in C_1. Because no faulty node exists in subcube0 and $w \neq u$ and $w \neq v$, by applying Lemma 4, there is a Hamiltonian cycle C_0 going through three links e, $(x : w)$, and $(w : y)$. By replacing the links $(x : w)$, $(w : y)$, $(x' : w')$, and $(w' : y')$ with the links $(x : x')$ and $(y : y')$, a fault-free cycle can be built that contains at least $(2^k - 1) + (2^k - 2(f - 1) - 1) = 2^{k+1} - 2f$ nodes and goes through link e in $(k + 1)$-cube (Figure 8).

Lemma 6: *There is a fault-free cycle containing at least $2^n - 2f$ nodes in an n-cube, $n \geq 3$, with $f \leq n - 1$ faulty nodes.*

Proof: We use induction on n to prove the lemma. The lemma is true for $n = 3$ as shown as in Figure 9, where two faulty nodes are denoted by dotted cycles.

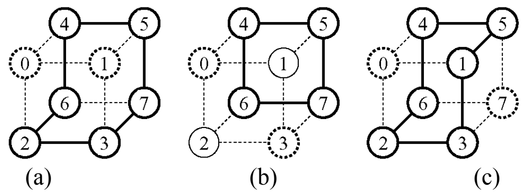

Figure 9 *Fault-free cycle in 3-cubes*

We assume that the lemma holds for $n = k \geq 3$. For $n = k + 1$, let f_0 and f_1 be the numbers of faulty nodes in subcube0 and subcube1, respectively, where $f_0 + f_1 = f \leq k$. Without the loss of generality, we assume that $f_0 \geq f_1$.

We can always divide the $(k + 1)$-cube into two k-cubes so that $f_0 \leq k - 1$. That is, there is at least one faulty node in subcube1 if $f = k$.

By our induction hypothesis, there is a fault-free cycle C_0 containing at least $2^k - 2f_0$ nodes in subcube0. Because $2^k - 2f_0 - 2f_1 = 2^k - 2f \geq 2^k - 2k > 0$ for $k > 3$, there is a link $(x : y)$ in C_0 such that the corresponding nodes x' and y' in subcube1 are not faulty.

Next, since $f_0 \geq f_1$, we have $f_1 \leq \lfloor f/2 \rfloor \leq \lfloor k/2 \rfloor \leq k - 2$ for $k \geq 3$. By applying Lemma 5, in subcube1, there is a fault-free cycle C_1 that contains at least $2^k - 2f_1$ nodes and goes through link $(x' : y')$. By replacing the links $(x : y)$ and $(x' : y')$ with the links $(x : x')$ and $(y : y')$, respectively, a fault-free cycle can be built that contains at least $(2^k - 2f_0) + (2^k - 2f_1) = 2^{k+1} - 2f$ nodes in $(k + 1)$-cube.

Now, we show that there is a fault-free cycle containing at least $2^{2m+1} - 2f$ nodes with $f \leq m$ faulty nodes in a DC(m), $m \geq 3$. In a DC(m), there are $h = 2^{m+1}$ clusters. Let f_i be the number of faulty nodes in cluster i, $0 \leq i \leq h - 1$ and $\sum_{i=0}^{h-1} f_i = f \leq m$. Let $f_x = \max(f_i | 0 \leq i \leq h - 1)$.

Case 1: $f_x \le m-2$. Theorem 2 says that there are 2^{m-1} disjoint virtual Hamiltonian cycles in a DC(m). Because $f \le m \le 2^{m-1}$, for $m \ge 3$, there exists a virtual Hamiltonian cycle that contains no faulty node. Because $f_i \le f_x < m-2$ for all i, $0 \le i \le h-1$, by Lemma 5, in each cluster i, there exists a fault-free cycle C_i that contains at least $2^m - 2f_i$ nodes and goes through a cube-link in the virtual Hamiltonian cycle. Therefore, we can construct a fault-free cycle in the DC(m) that contains at least $\sum_{i=0}^{h-1}(2^m - 2f_i) = 2^{2m+1} - 2f$ nodes.

Case 2: $f_x = m-1$. By Lemma 6, there exists a fault-free cycle C_x containing at least $2^m - 2(m-1)$ nodes in cluster x. Because $f_x = m-1$ and $f \le m$, there is at most a faulty node w not in cluster x. By Corollary 1, there are $(2^m - 2(m-1))/2 = 2^{m-1} - m + 1 > 1$, disjoint virtual Hamiltonian cycles that go through a link in C_x. Therefore, we can select a virtual Hamiltonian cycle in DC(m) that contains a link in C_x but does not contain node w. For each cluster i, $0 \le i \le h-1$ and $i \ne x$, because $f_i < m-2$, by Lemma 5, there exists a fault-free cycle containing at least $2^m - 2f_i$ nodes that goes through a link in the selected virtual Hamiltonian cycle. Therefore, a fault-free cycle in the DC(m) containing at least $(2^m - 2(m-1) - 1) + (2^{m+1} - 1)2^m - 2 = 2^{2m+1} - 2$m nodes can be built.

Case 3: $f_x = m$. Let w be a faulty node in cluster x. Mark w as non-faulty. By Lemma 6, there exists a fault-free cycle C_x that contains at least $2^m - 2(m-1)$ nodes in cluster x. If C_x does not contain w, then the construction of fault-free cycle in DC(m) is similar to Case 2. Otherwise, a new scheme for constructing virtual Hamiltonian cycle is needed. Without loss of generality, we assume $w' = 0,0 \ldots 000,0 \ldots 001$, and let $w' = 0,0 \ldots 011,0 \ldots 011$. Intuitively, the new virtual Hamiltonian cycle will contain two more cube links by passing through nodes w and w'. That is, in clusters 0 and 3 of class 0, two consecutive cube-links will be in the virtual Hamiltonian cycle. Supposing that VHC[2^{m+2}] is the selected virtual Hamiltonian cycle and w is the given faulty node, the following algorithm generates a modified virtual Hamiltonian cycle. Figure 10 illustrates a modified VHC for a DC(3). For clarity, we have put nodes w = 0000001 and w' = 0011011 in the figure.

Algorithm 3 (modified VHC(m, w))

```
begin          /* find a modified virtual hamiltonian cycle */
    x = w ⊕ 1;
    for i = 0 to 2^(m+2) − 1 do
        VHC[i] = VHC[i] ⊕ x;          /* mapping w to 1 */
    endfor
    for i = 1 to 8 do
        VHC[i] = VHC[i] ⊕ 2;     /* modifying the cycle */
    endfor
    for i = 0 to 2^(m+2) − 1 do
        VHC[i] = VHC[i] ⊕ x;            /* mapping back */
    endfor
end
```

Original VHC	Modified VHC
0000000 : 0000001 :	0000000 : 0000001 (w) : 0000011 :
1000001 : 1001001 :	1000011 : 1001011 :
0001001 : 0001011 :	0001011 : 0001001 :
1001011 : 1011011 :	1001001 : 1011001 :
0011011 : 0011010 :	0011001 : 0011011 (w') : 0011010 :
1011010 : 1010010 :	1011010 : 1010010 :
0010010 : 0010110 :	0010010 : 0010110 :
1010110 : 1110110 :	1010110 : 1110110 :
0110110 : 0110111 :	0110110 : 0110111 :
1110111 : 1111111 :	1110111 : 1111111 :
0111111 : 0111101 :	0111111 : 0111101 :
1111101 : 1101101 :	1111101 : 1101101 :
0101101 : 0101100 :	0101101 : 0101100 :
1101100 : 1100100 :	1101100 : 1100100 :
0100100 : 0100000 :	0100100 : 0100000 :
1100000 : 1000000 :	1100000 : 1000000 :

Figure 10 *Virtual Hamiltonian cycles in a DC(3)*

Then we use the modified VHC to build a fault-free cycle in DC(m) as follows. The two paths, ($x : w : y$) and ($x' : w' : y'$), in the modified VHC are replaced with the paths ($x \to y$) and ($x' \to y'$) containing $(2^m - 2(m-1) - 1)$ nodes and $(2^m - 1)$ nodes, respectively. All other cube-links in the modified VHC are replaced with Hamiltonian paths. The fault-free path in cluster x contains at least $2^m - 2(m-1) - 1$ nodes; the fault-free path in the cluster of node w' contains $2^m - 1$ nodes; and each of remaining clusters contains 2^m nodes. Therefore, the fault-free cycle in DC(m) constructed above contains at least $(2^m - 2(m-1) - 1) + (2^m - 1) + (2^{m+1} - 2)2^m = 2^{2m+1} - 2m$ nodes.

We summarise these results in the following theorem.

Theorem 3: *There is a fault-free cycle containing at least $2^{2m+1} - 2f$ nodes in a DC(m) with f faulty nodes, where $f \le m$ and $m \ge 3$.*

5 CONCLUSION AND FUTURE WORK

This paper shows that a fault-free cycle containing at least $2^{2m+1} - 2f$ nodes can be constructed in a DC(m) with $f \le m$ faulty nodes. Because the dual-cube keeps most properties of the hypercube and can link much more nodes than other variations of hypercube with the same number of links per node, it could be used as an interconnection network for large-scale parallel computers.

Recently, much of the community has moved on to lower-dimensional topologies such as meshes and tori. However, the SGI Origin2000, a fairly recent multiprocessor, does use a hypercube topology, so the dual-cube could be of use to industry. A lot of issues concerning the dual-cube require further research. Some of them are:

- evaluate the architecture complexity vs. performance of benchmarks vs. real cost
- investigate the embedding of other frequently used topologies into a dual-cube
- develop techniques for mapping application algorithms onto a dual-cube
- develop fault-tolerant routing algorithms for a dual-cube with mixed link and node failures.

REFERENCES

Amawy, A.E. and Latifi, S. (1991) 'Properties and performance of folded hypercubes', *IEEE Transactions on Parallel and Distributed Systems*, Vol. 2, No. 1, pp.31–42.

Efe, K. (1992) 'The crossed cube architecture for parallel computation', *IEEE Transactions on Parallel and Distributed Systems*, September, Vol. 3, No. 5, pp.513–524.

Ghose, K. and Desai, K.R. (1995) 'Hierarchical cubic networks', *IEEE Transactions on Parallel and Distributed Systems*, April, Vol. 6, No. 4, pp.427–435.

Hayes, J.P. and Mudge, T.N. (1989) 'Hypercube supercomputers', *Proc. IEEE*, December, Vol. 17, No. 12, pp.1829–1841.

Hsieh, S.Y., Chen, G.H. and Ho, C.W. (1998) 'Embed longest rings onto star graphs with vertex faults', *Proceedings of the International Conference on Parallel Processing*, pp.140–147.

Jiang, Z. and Wu, J. (2003) 'Fault-tolerant routing in dual-cube networks', *Proceedings of the 7th Joint Conference on Information Sciences*, Vol. 9, pp.389–392.

Latifi, S., Zheng, S.Q. and Bagherzadeh, N. (1992) 'Optimal ring embedding in hypercubes with faulty links', *Proceedings of the Twenty-Second International Symposium on Fault-tolerant Computing*, pp.178–184.

Li, Y. and Peng, S. (2000) 'Dual-cubes: a new interconnection network for high-performance computer clusters', *Proceedings of the 2000 International Computer Symposium, Workshop on Computer Architecture*, ChiaYi, Taiwan, December, pp.51–57.

Li, Y., Peng, S. and Chu, W. (2001) 'Efficient collective communications in dual-cube', *Proceedings of the Thirteen IASTED International Conference on Parallel and Distributed Computing and Systems*, Anaheim, USA, August, pp.266–271.

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

Preparata, F.P. and Vuillemin, J. (1981) 'The cube-connected cycles: a versatile network for parallel computation', *Commun. ACM*, May, Vol. 24, pp.300–309.

Rowley, R.A. and Bose, B. (1994) 'Fault-tolerant ring embedding in de bruijn networks', *IEEE Transactions on Computers*, Vol. 46, No. 2, pp.187–190.

SGI(1997) *Origin2000 Rackmount Owner's Guide*, *007-3456-003*, http://techpubs.sgi.com/library/tpl/cgi-bin/getdoc.cgi?coll=hdwr&db=bks&fname=/SGI_Admin/books/Or2000_Rack_OG/sgi_html/ch01.html&srch=origin2000

Tseng, Y.C., Chang, S.H. and Sheu, J.P. (1997)'Fault-tolerant ring embedding in a star graph with both link and node failures', *Transactions on Parallel and Distributed Systems*, Vol. 8, No. 12, pp.1185–1195.

Tseng, Y-C. (1996) 'Embedding a ring in a hypercube with both faulty links and faulty nodes', *Information Processing Letters*, August, Vol. 59, No. 4, pp.217–222.

Tucker, L.W. and Robertson, G.G. (1988) 'Architecture and applications of the connection machine', *IEEE Computer*, August, Vol. 21, pp.26–38.

Wang, D.J. (2001) 'Embedding Hamiltonian cycles into folded hypercubes with link faults', *Journal of Parallel and Distributed Computing*, Vol. 61, No. 4, pp.545–564.

Wu, C. and Wu, J. (2003) 'On self-similarity and hamiltonicity of dual-cubes', *Proceedings of Workshop on Massively Parallel Processing (in conjunction with IPDPS)*, CD-ROM, p.4.

Ziavras, S.G. (1994) 'Rh: a versatile family of reduced hypercube interconnection networks', *IEEE Transactions on Parallel and Distributed Systems*, November, Vol. 5, No. 11, pp.1210–1220.