ELSEVIER

Journal of Systems Architecture 42 (1996) 289–307

JOURNAL OF SYSTEMS ARCHITECTURE

Topological properties of hierarchical cubic networks

Wei-Kuo Chiang, Rong-Jaye Chen *

Department of Computer Science and Information Engineering, National Chiao Tung University, 1001 Ta Hsueh Road, Hsinchu 30050, Taiwan, ROC

Received 27 February 1995; revised 1 February 1996; accepted 29 April 1996

Abstract

The hierarchical cubic network (HCN), which takes hypercubes as basic clusters, was first introduced in [6]. Compared with the hypercube of the same size, the HCN requires only about half the number of links and provides a lower diameter. This paper first proposes a shortest-path routing algorithm and an optimal broadcasting algorithm for the HCN. We then show that the HCN is optimal fault tolerant by constructing node-disjoint paths between any two nodes, and demonstrate that the HCN is Hamiltonian. Moreover, it is shown that the average dilation for hypercube emulation on the HCN is bounded by 2. This result guarantees that all the algorithms designed for the hypercube can be executed on the HCN with a small degradation in time performance.

Keywords: Hypercube; Interconnection network; Shortest-path routing; Broadcasting; Fault tolerance; Hamiltonian; Emulation

1. Introduction

Advances in technology have made possible interconnection of a large number of computing elements to form a massively parallel computer system with control, processing, and information being distributed among these elements. One of the dominating factors that governs the performance of a parallel system is the underlying communication network. Hence, the choice of the topology of the interconnection network is critical in the design of massively parallel computer systems. For this reason, a lot of network topologies have been proposed in the literature [2,6,7,12,16,17], and a large amount of research has been focused on the design and evaluation of these networks [1,9,10,14].

* Corresponding author. Email: rjchen@csie.nctu.edu.tw.

PII S1383-7621(96)00019-7

The n-dimensional hypercube (or n-cube) has been one of the most popular interconnection networks because it provides the logarithmic diameter, high connectivity, symmetry, and simple routing; it is also able to efficiently emulate many other topologies such as rings, trees, meshes, butterfly and shuffle-exchange networks [11]. The hypercube has been used to design various commercial multiprocessor machines and has been extensively studied [1,14,15], while many efficient parallel algorithms for hypercubes have been implemented [3].

However, the hypercube has been considered unsuitable for building large systems since the relatively high node degree results in an additional difficulty in interconnection and an extra complexity in processor design. As the dimension n of the hypercube increases, the number of communication ports and links per processor grows rapidly so that the feasibility of higher dimensional hypercube machines becomes questionable. Therefore, there is a substantial interest in interconnection networks with hypercube characteristics but with a reduced node degree; consequently, some hierarchical topologies of the hypercube structure have been investigated in recent years [7,8,12,17].

The hierarchical cubic network (HCN) has been proposed and analyzed by Ghose and Desai in [8–10]. The HCN takes hypercubes as basic clusters, which are connected in a complete manner. An HCN that consists of 2^n basic clusters – each of which is an n-cube – is referred to as HCN(n,n). Compared with the hypercube of the same size, the HCN requires only about half the number of links and has a considerably lower diameter. However, the routing algorithm for the HCN proposed in [10] is not optimal in distance wise, and thus its diameter obtained from the path determined by the routing algorithm is just an upper bound.

This paper is organized as follows. Section 2 defines the topology of an HCN(n,n) formally. Section 3 presents a shortest-path routing algorithm for an HCN(n,n), deriving a very tight bound for the diameter, $(n+\lceil n/3\rceil+1)$. Section 4 gives a lower bound of time complexity of any broadcasting on an HCN(n,n) and develops an optimal broadcasting algorithm. In Section 5, we show that an HCN(n,n) is optimal fault tolerant by constructing node-disjoint paths between any two nodes and give an upper bound for the fault-diameter, $(2n+6)$. In Section 6, we demonstrate that an HCN(n,n) has a Hamiltonian cycle. Section 7 describes how to emulate the hypercube on the HCN with a small performance degradation. Finally, in Section 8 the conclusion of this research and topics for future research are given.

2. The hierarchical cubic network

Throughout this paper (unless stated otherwise), we use the lower-case letters a, b, c and d to denote binary bits and the capital letters A, B, C and D to denote n-bit binary sequences. For example, $A=a_{n-1}\dots a_0$ and $a_i\in\{0,1\}$ for $0\le i<n$. Let $\bar{a}_i$ denote the complement of a_i. For simplicity, let $A^i=a_{n-1}\dots\bar{a}_i\dots a_0$ and $\bar{A}=\bar{a}_{n-1}\dots\bar{a}_i\dots\bar{a}_0$.

An HCN(n,n) consists of 2^n basic clusters, each of which is an n-cube. Each node in an HCN(n,n) is assigned a $2n$-bit binary sequence A_1A_0. The most significant n bits, A_1, identify the cluster to which this node belongs, and the least significant n bits, A_0, form node addresses within the cluster. Let x be a don't-care symbol, and X be a sequence of n don't-care symbols, i.e., $X=x^n$, where the superscript is the repetition factor. That is, we can use A_1X to denote the cluster containing node A_1A_0.

Within a cluster, the edge connections are in the same fashion as those in the hypercube, i.e., node A_1A_0 is connected with nodes $A_1A_0^i$, $0\le i\le n-1$. These links are referred to as *local* edges of dimension i. The

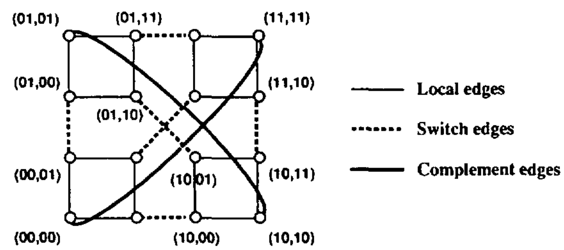


Fig. 1. An HCN(2,2).

edges between these basic clusters are formed by connecting node $A_1 A_0$ to node $A_0 A_1$ for all A_1 and A_0, with $A_1 \neq A_0$. These edges are referred to as *switch* edges, because one node can be reached from the other node by switching its most significant n bits with its least significant n bits. In the case of $A_1 = A_0$, node $A_1 A_0$ is connected to node $\bar{A}_1 \bar{A}_0$. These edges in this case are referred to as *complement* edges. The degree of each node in an HCN(n,n) is $(n+1)$; each node is connected with n local edges plus either a switch edge or a complement edge. An HCN(2,2) is shown in Fig. 1.

3. Shortest-path routing and diameter

Message routing is a central problem to the application of an interconnection network. A good interconnection network should facilitate message routing in a simple and efficient way. An efficient routing algorithm for a given network makes better use of the network and increases the overall performance of the multiprocessor system built on the network. In [10], Ghose and Desai proposed a routing algorithm for an HCN(n,n) and established an upper bound for the diameter, $(n + \lfloor n/2 \rfloor + 1)$. In this section, we will present a shortest-path routing algorithm for an HCN(n,n), and then derive a tighter bound of the diameter, $(n + \lceil n/3 \rceil + 1)$. Additionally, a conservative estimate of the average distance of an HCN(n,n) is also included.

3.1. Shortest-path routing algorithm

To make this paper self-contained, we first outline the routing algorithm for the hypercube in [14]. Let A and B be any two nodes in a n-cube, where A and B represent two n-bit binary sequences. The routing from A to B can be done by crossing successively the nodes whose labels are those obtained by modifying the bits of A one by one in order to transform A into B. The distance between A and B is equal to the number of bits that differ between A and B, i.e., to the Hamming distance H(A,B) which is defined as follows. The Hamming distance between $A = a_{n-1} \dots a_1 a_0$ and $B = b_{n-1} \dots b_1 b_0$ is defined as

$$\mathrm{H}(A,B) = \sum_{i=0}^{n-1} \mathrm{h}(a_i,b_i),$$

where

$$\mathrm{h}(a_i,b_i) = \begin{cases} 1 & \text{if } a_i \neq b_i; \\ 0 & \text{if } a_i = b_i. \end{cases}$$

The next lemma related to the Hamming distance is useful for our analysis that follows.

Lemma 3.1. $\mathrm{H}(A,B) + \mathrm{H}(A,C) \leq 2n - \mathrm{H}(B,C)$, *where A, B, and C are n-bit binary sequences.*

Proof. By the definition of the Hamming distance, we have three equations below:

$$H(A,\bar{B}) = n - H(A,B); \tag{1}$$

$$H(A,B) + H(A,C) \geq H(B,C); \tag{2}$$

$$H(\bar{A},\bar{B}) = H(A,B). \tag{3}$$

Therefore,

$$\begin{aligned} &\mathrm{H}(A,B) + \mathrm{H}(A,C) && [\text{by } (1)] \\ &= n - \mathrm{H}(A,\bar{B}) + n - \mathrm{H}(A,\bar{C}) && \\ &= 2n - \{\mathrm{H}(A,\bar{B}) + \mathrm{H}(A,\bar{C})\} && [\text{by } (2)] \\ &\leq 2n - \mathrm{H}(\bar{B},\bar{C}) && [\text{by } (3)] \\ &= 2n - \mathrm{H}(B,C). && \square \end{aligned}$$

To clarify our representation, the term "hypercube routing" is used for the routing within a cluster of an HCN(n,n). The labels of nodes coupled with two symbols $\rightarrow$ and $\Rightarrow$ are used to denote specific paths in an HCN(n,n). The symbol $\rightarrow$ is used to represent a switch edge or a complement edge. The symbol $\Rightarrow$ is used to represent a hypercube routing path within a cluster of an HCN(n,n), which is of length at most n. Let $|P|$ denote the length of path P.

Lemma 3.2. *Given node A_1A_0 and node B_1B_0 with $A_1 = B_1$ in an HCN(n,n), path P_0: $A_1A_0 \Rightarrow A_1B_0$ is the shortest path between them, which can be determined by hypercube routing.*

Lemma 3.3. *Any shortest path between any two nodes in an HCN(n,n) contains at most one complement edge.*

Proof. To prove this lemma, by contradiction, assume that there is a shortest path containing more than one complement edge between two nodes in an HCN(n,n). Let $P: AA \rightarrow \overline{AA} \ldots \overline{BB} \rightarrow BB$ be a subpath between any two complement edges in this shortest path, which is of length k. By the assumption, P is a shortest path between AA and BB. Now, if we complement the labels of all the nodes on P, then the resulting path is $\bar{P}: \overline{AA} \rightarrow AA \ldots BB \rightarrow \overline{BB}$, which is also of length k. So, there is a shorter path of length $(k-2)$ between AA and BB. This is a contradiction. Therefore, a shortest path between two nodes in an HCN(n,n) contains at most one complement edge. $\square$

Lemma 3.4. *If a shortest path between two nodes in an HCN(n,n) does not traverse any complement edge, then the path contains at most two switch edges.*

Proof. To prove it, we need to show that given two nodes A_1A_0 and B_1B_0, any path containing three switch edges between them can be reduced to a shorter path containing only one switch edge. Let $P: A_1A_0 \Rightarrow A_1C \rightarrow CA_1 \Rightarrow CD \rightarrow DC \Rightarrow DB_1 \rightarrow B_1D \Rightarrow B_1B_0$ be a path containing three switch edges, where $|P| = \mathrm{H}(A_0,C) + 1 +$

$H(A_1,D) + 1 + H(C,B_1) + 1 + H(D,B_0) \geq 3 + H(A_0,B_1) + H(A_1,B_0)$. Then P can be reduced to $P^*: A_1A_0 \Rightarrow A_1B_1 \rightarrow B_1A_1 \Rightarrow B_1B_0$ which is of length $H(A_0,B_1) + 1 + H(A_1,B_0) < |P|$ and traverses only one switch edge. Hence, any shortest path which does not traverse any complement edge contains at most two switch edges. □

Lemma 3.5. *Given node A_1A_0 and node B_1B_0 with $A_1 \neq B_1$ in an HCN(n,n), path $P_\alpha: A_1A_0 \Rightarrow A_1B_1 \rightarrow B_1A_1 \Rightarrow B_1B_0$ is the only path containing one switch edge.*

Lemma 3.6. *Given node A_1A_0 and node B_1B_0 with $A_1 \neq B_1$ in an HCN(n,n), path $P_\beta: A_1A_0 \rightarrow A_0A_1 \Rightarrow A_0B_1 \rightarrow B_1A_0 \Rightarrow B_1B_0$ is a shortest path among all the routing paths containing two switch edges.*

Proof. Let $P: A_1A_0 \Rightarrow A_1C \rightarrow CA_1 \Rightarrow CB_1 \rightarrow B_1C \Rightarrow B_1B_0$ be a path containing two switch edges. Since $|P| = H(A_0,C) + 1 + H(A_1,B_1) + 1 + H(C,B_0) \geq 2 + H(A_1,B_1) + H(A_0,B_0) = |P_\beta|$, P can be replaced by P_β with no more length. □

Lemma 3.7. *If a shortest path between two nodes in an HCN(n,n) traverses a complement edge, each of the two subpaths separated by the complement edge contains at most one switch edge.*

Proof. From Lemma 3.4, each subpath separated by the complement edge contains at most two switch edges. Here, we only need to show that there is no shortest path containing one complement edge such that one of its subpaths separated by the complement edge contains two switch edges. By contradiction, assume that there exists such a shortest path. Let $P: AA \Rightarrow AC \rightarrow CA \Rightarrow CB_1 \rightarrow B_1C \Rightarrow B_1B_0$ be a separated subpath of the shortest path, implying that P is the shortest path between AA and B_1B_0. The length of path P is $|P| = H(A,C) + 1 + H(A,B_1) + 1 + H(C,B_0)$. However, $P^*: AA \Rightarrow AB_1 \rightarrow B_1A \Rightarrow B_1B_0$ is a shorter path than P, which produces a contradiction. □

Let $P_C: A_1A_0 \Rightarrow A_1C \rightarrow CA_1 \Rightarrow CC \rightarrow \overline{CC} \Rightarrow \overline{C}B_1 \rightarrow B_1\overline{C} \Rightarrow B_1B_0$ be a path containing one complement edge and two switch edges, then $|P_C| = 3 + H(A_0,C) + H(A_1,C) + H(B_1,\overline{C}) + H(B_0,\overline{C})$. Let $f(C) = |P_C|$. If we choose a node C^*C^* such that $f(C^*) = \min f(C)$, then the resulting path $P_{C^*}: A_1A_0 \Rightarrow A_1C^* \rightarrow C^*A_1 \Rightarrow C^*C^* \rightarrow \overline{C}^*\overline{C}^* \Rightarrow \overline{C}^*B_1 \rightarrow B_1\overline{C}^* \Rightarrow B_1B_0$ is a shortest path among all paths containing one complement edge and two switch edges.

Let x denote a don't-care symbol. If some bit in a sequence corresponds to symbol x, then this bit can be assigned 0 or 1 arbitrarily. As an example, a sequence $0x00x$ can be assigned as 00000, 01000, 00001, or 01001. Two sequences are said to match each other if their corresponding bits except don't-care symbols are identical. For instance, $0100x$ matches $0x001$.

The search for a chosen node C^*C^* can be achieved by defining an operation $\oplus$ as follows:

$$C^* = \oplus\left(A_1, A_0, \overline{B}_1, \overline{B}_0\right),$$

where

$$\begin{cases} c_i^* = 0 \text{ if } a_{1i} + a_{0i} + \overline{b}_{1i} + \overline{b}_{0i} \leq 1; \\ c_i^* = x \text{ if } a_{1i} + a_{0i} + \overline{b}_{1i} + \overline{b}_{0i} = 2 \text{ for } 0 \leq i \leq n-1; \\ c_i^* = 1 \text{ if } a_{1i} + a_{0i} + \overline{b}_{1i} + \overline{b}_{0i} \geq 3. \end{cases}$$

Lemma 3.8. *Given node $A_1 A_0$ and node $B_1 B_0$ with $A_1 \neq B_1$ in an HCN(n,n), path $P_\gamma : A_1 A_0 \Rightarrow A_1 C^* \rightarrow C^* A_1 \Rightarrow C^* C^* \rightarrow \bar{C}^* \bar{C}^* \Rightarrow \bar{C}^* B_1 \rightarrow B_1 \bar{C}^* \Rightarrow B_1 B_0$ is a shortest path among all the routing paths containing one complement edge and two switch edges, where $C^* = \oplus(A_1, A_0, \bar{B}_1, \bar{B}_0)$.*

Example 3.1. A shortest path with one complement edge and two switch edges between node (0001,1000) and node (1011,1101) in an HCN(4,4), based on Lemma 3.8, is [(0001,1000),(0001,0000),(0000,0001),(0000,0000), (1111,1111),(111 1,1011),(1011,1111),(1011,1101)], which is a path of length 7. Note that the chosen node is (0000,0000).

Lemma 3.9. *Given node $A_1 A_0$ and node $B_1 B_0$ with $A_1 \neq B_1$ in an HCN(n,n), if A_1 (or $\bar{B}_1$) matches C^*, then let $C^* = A_1$ (or $\bar{B}_1$) such that P_δ (or P_μ) is of length $(|P_\gamma| - 1)$ where*

$$C^* = \oplus\left(A_1, A_0, \bar{B}_1, \bar{B}_0\right),$$

$$P_\delta : A_1 A_0 \Rightarrow A_1 A_1 \rightarrow \bar{A}_1 \bar{A}_1 \Rightarrow \bar{A}_1 B_1 \rightarrow B_1 \bar{A}_1 \Rightarrow B_1 B_0,$$

and

$$P_\mu : A_1 A_0 \Rightarrow A_1 \bar{B}_1 \rightarrow \bar{B}_1 A_1 \Rightarrow \bar{B}_1 \bar{B}_1 \rightarrow B_1 B_1 \Rightarrow B_1 B_0.$$

Proof. When A_1 (or $\bar{B}_1$) matches C^*, P_δ (or P_μ) contains only one switch edge; that is, we can reduce one switch edge while keeping the property of minimum $f(C)$. □

Notice that P_δ (or P_μ) is a shorter path than P_γ only when A_1 (or $\bar{B}_1$) matches C^*; in other cases, P_δ (or P_μ) is a longer path than P_γ. However, either P_δ or P_μ is the shortest path with one complement edge and one switch edge between $A_1 A_0$ and $B_1 B_0$ in an HCN(n,n).

Example 3.2. The routing path from (0001,1000) to (1011,1101) in an HCN(4,4), by using P_δ in Lemma 3.9, is [(0001,1000),(0001,0000),(0001,0001),(1110,1110),(1110,1111),(1110,1011),(1011,1110),(1011,1111),(1011, 1101)], which is a path of length 8. This path is longer than that path using P_γ in Example 3.1.

Lemma 3.10. *Given node $A_1 A_0$ and node $B_1 B_0$ with $A_1 \neq B_1$ in an HCN(n,n), if $A_1 = \bar{B}_1$, then path $P_\lambda : A_1 A_0 \Rightarrow A_1 A_1 \rightarrow \bar{A}_1 \bar{A}_1 \Rightarrow B_1 B_0$ is the only path containing one complement edge and no switch edge between them.*

Theorem 3.1. *A shortest path between node $A_1 A_0$ and node $B_1 B_0$ with $A_1 \neq B_1$ in an HCN(n,n) is the path whose distance is the smallest of the following six paths (if exists):*

$$P_\alpha : A_1 A_0 \Rightarrow A_1 B_1 \rightarrow B_1 A_1 \Rightarrow B_1 B_0,$$

$$P_\beta : A_1 A_0 \rightarrow A_0 A_1 \Rightarrow A_0 B_1 \rightarrow B_1 A_0 \Rightarrow B_1 B_0,$$

$$P_\gamma : A_1 A_0 \Rightarrow A_1 C^* \rightarrow C^* A_1 \Rightarrow C^* C^* \rightarrow \bar{C}^* \bar{C}^* \Rightarrow \bar{C}^* B_1 \rightarrow B_1 \bar{C}^* \Rightarrow B_1 B_0,$$

if A_1 matches C^*,

$$P_\delta : A_1 A_0 \Rightarrow A_1 A_1 \rightarrow \bar{A}_1 \bar{A}_1 \Rightarrow \bar{A}_1 B_1 \rightarrow B_1 \bar{A}_1 \Rightarrow B_1 B_0,$$

if $\overline{B}_1$ *matches* C^*,

$$P_\mu : A_1 A_0 \Rightarrow A_1 \overline{B}_1 \rightarrow \overline{B}_1 A_1 \Rightarrow \overline{B}_1 \overline{B}_1 \rightarrow B_1 B_1 \Rightarrow B_1 B_0,$$

if $A_1 = \overline{B}_1$, *then*

$$P_\lambda : A_1 A_0 \Rightarrow A_1 A_1 \rightarrow \overline{A}_1 \overline{A}_1 \Rightarrow B_1 B_0,$$

where

$$C^* = \oplus\left(A_1, A_0, \overline{B}_1, \overline{B}_0\right).$$

Proof. Previous results, from Lemmas 3.3, 3.4 and 3.7, show a shortest path between any two nodes in an HCN(n,n) traverses at most two switch edges and at most one complement edge. The only path with one switch edge and no complement edge is given in Lemma 3.5 (P_α). Although there may be a few paths with two switch edges, a shortest path among them is found in Lemma 3.6 (P_β). A shortest path with one complement edge and two switch edges is chosen in Lemma 3.8 (P_γ). Lemma 3.9 gives a shortest path with one complement edge and one switch edge (P_δ or P_μ). A path with one complement edge and no switch edge for a special case $A_1 = \overline{B}_1$ is listed in Lemma 3.10 (P_λ). Based on the above discussion, we obtain this theorem. □

Based on Theorem 3.1, we present a message routing strategy for an HCN(n,n). First, a procedure Route-Decision for determining the shortest path is given in Fig. 2, which is executed at the source node. In addition, a distributed routing algorithm Distributed-Route is given in Fig. 3, which is invoked by the node receiving the message. There are five arguments in this algorithm which are denoted as *msg* representing the message to be transmitted, character x indicating the name of path determined by procedure Route-Decision, *Sourc, Dest and Choice* indicating the labels of the source node, the destination nodes, and the chosen node.

Procedure Route-Decision($x, A_1A_0, B_1B_0, C^*C^*$);
{* *This procedure invoked by* A_1A_0 *determines the shortest path from* A_1A_0 *to* B_1B_0. *}
{* x: *path name;* A_1A_0: *source;* B_1B_0: *destination;* C^*C^* *exists if* $x = \gamma$ *}

if $A_1 = B_1$ **then** $x := 0$; {* *Send msg using hypercube routing only.* *}
else $L1 := H(A_0, B_1)$; $L2 := H(A_1, B_0)$; $L3 := H(A_1, B_1)$;
$L4 := H(A_0, B_0)$; $L5 := H(B_1, B_0)$; $L6 := H(A_0, A_1)$;
$C^* := \oplus(A_1, A_0, \overline{B}_1, \overline{B}_0)$;
/* use R_x denote the length of path P_x, i.e., $R_x = |P_x|$ */
$R_\alpha := 1 + L1 + L2$;
$R_\beta := 2 + L3 + L4$;
$R_\gamma := 3 + H(A_0, C^*) + H(A_1, C^*) + H(B_1, \overline{C}^*) + H(B_0, \overline{C}^*)$;
$R_\delta := 2n + 2 - L1 - L3 + L5$;
$R_\mu := 2n + 2 - L2 - L3 + L6$;
$R_\lambda := n + 1 + L6 - L2$;
$R_{y^*} := \min\{ R_y \mid y = \alpha, \beta, \gamma, \delta, \mu, \lambda \}$
$x := y^*$
Send [$msg, x, A_1A_0, B_1B_0, C^*C^*$] *along path* P_x.
end.

Fig. 2. A procedure for determining the shortest routing path in an HCN(n,n).

```
Algorithm Distributed-Route(msg,x,Sourc,Dest,Choice);
{* This algorithm is invoked by node C_1C_0 receiving the message. *}
{* x: path name; Choice exists if x = γ *}

  A_1A_0 := Sourc;
  B_1B_0 := Dest;
  Msg := [msg,x,Sourc,Dest,Choice]
  if C_1 = B_1, C_0 = B_0 then Receive Msg; stop.
  case x = 0:  Send Msg to next node using hypercube routing;
  case x = α:  if C_1 = A_1, C_0 ≠ B_1   then Send Msg to A_1B_1 using hypercube routing;
               if C_1 = A_1, C_0 = B_1   then Send Msg to B_1A_1 via the switch edge;
               if C_1 = B_1, C_0 ≠ B_0   then Send Msg to B_1B_0 using hypercube routing;
  case x = β:  Send [msg,α,A_0A_1,B_1B_0,–] to A_0A_1 via the switch edge;
  case x = γ:  C*C* := Choice;
               if C_1 = A_1, C_0 ≠ C*    then Send Msg to A_1C* using hypercube routing;
               if C_1 = A_1, C_0 = C*    then Send Msg to C*A_1 via the switch edge;
               if C_1 = C*, C_0 ≠ C*     then Send Msg to C*C* using hypercube routing;
               if C_1 = C*, C_0 = C*     then
                     Send [msg,α,C̄*C̄*,B_1B_0,–] to C̄*C̄* via the complement edge;
  case x = δ:  if C_1 = A_1, C_0 ≠ A_1   then Send Msg to A_1A_1 using hypercube routing;
               if C_1 = A_1, C_0 = A_1   then
                     Send [msg,α,Ā_1Ā_1,B_1B_0,–] to Ā_1Ā_1 via the complement edge;
  case x = μ:  if C_1 = A_1, C_0 ≠ B̄_1   then Send Msg to A_1B̄_1 using hypercube routing;
               if C_1 = A_1, C_0 = B̄_1   then
                     Send [msg,λ,B̄_1A_1,B_1B_0,–] to B̄_1A_1 via the switch edge;
  case x = λ:  if C_1 = A_1, C_0 ≠ A_1   then Send Msg to A_1A_1 using hypercube routing;
               if C_1 = A_1, C_0 = A_1   then Send Msg to Ā_1Ā_1 via the complement edge;
               if C_1 = Ā_1, C_0 ≠ B_0   then Send Msg to Ā_1B_0 using hypercube routing;
end.
```

Fig. 3. A distributed routing algorithm for an HCN(n,n).

Note that the chosen node *Choice* exists only if P_γ is used for the routing; otherwise, symbol – is used for this argument.

3.2. Diameter bounds

The distance between two nodes in a network is the length of a shortest path joining them. The diameter of a network is the maximum distance among all pairs of nodes. Here, an upper bound and a lower bound for the diameter of HCN(n,n) are derived based on Theorem 3.1.

Theorem 3.2. *The diameter D_n of HCN$(n,n) \leq n + \lceil n/3 \rceil + 1$, where $\lceil p \rceil$ denotes the smallest integer not less than p.*

Proof. Consider the routing path between node $A_1 A_0$ and node $B_1 B_0$ in an HCN(n,n) under our shortest-path routing strategy. Theorem 3.1 gives six possible routing paths between them. Four of these six paths are used to derive an upper bound for the diameter, and their lengths are indicated below:

$$|P_\alpha| = \mathrm{H}(A_0, B_1) + 1 + \mathrm{H}(A_1, B_0),$$
$$|P_\beta| = 1 + \mathrm{H}(A_1, B_1) + 1 + \mathrm{H}(A_0, B_0),$$
$$|P_\delta| = \mathrm{H}(A_0, A_1) + 1 + \mathrm{H}(\bar{A}_1, B_1) + 1 + \mathrm{H}(\bar{A}_1, B_0),$$

and

$$|P_\mu| = \mathrm{H}(A_0,\bar{B}_1) + 1 + \mathrm{H}(A_1,\bar{B}_1) + 1 + \mathrm{H}(B_1,B_0).$$

The sum of the lengths of three paths P_α, P_β, and P_δ is

$$\begin{aligned}\mathrm{SUM1} &= 2n+5+\mathrm{H}(B_1,A_0)+\mathrm{H}(A_0,B_0)+\mathrm{H}(A_1,A_0) \quad [\text{by Lemma 3.1}]\\ &\le 4n+5+\mathrm{H}(A_1,A_0)-\mathrm{H}(B_1,B_0).\end{aligned}$$

On the other hand, the sum of the lengths of three paths P_α, P_β, and P_μ is

$$\begin{aligned}\mathrm{SUM2} &= 2n+5+\mathrm{H}(A_1,B_0)+\mathrm{H}(A_0,B_0)+\mathrm{H}(B_1,B_0) \quad [\text{by Lemma 3.1}]\\ &\le 4n+5+\mathrm{H}(B_1,B_0)-\mathrm{H}(A_1,A_0).\end{aligned}$$

Since $\mathrm{SUM1}+\mathrm{SUM2} \le 8n+10$, either SUM1 or SUM2 is less than or equal to $4n+5$. If the sum of the lengths of three paths is at most $4n+5$, the length of the shortest path among these three paths is at most $\lfloor (4n+5)/3 \rfloor = n + \lceil n/3 \rceil + 1$. Thus, the length of the shortest path among these four possible paths is less than or equal to $n+\lceil n/3\rceil+1$. □

Theorem 3.3. *The diameter D_n of HCN$(n,n) \ge n + \lfloor (n+1)/3 \rfloor + 1$, where $\lfloor q \rfloor$ is the largest integer not exceeding q. More precisely, a lower bound for the diameter of HCN(n,n) is found as follows:*

$$D_n \ge \begin{cases} 4k+1 & \text{if } n=3k;\\ 4k+2 & \text{if } n=3k+1;\\ 4k+4 & \text{if } n=3k+2.\end{cases}$$

Proof. Based on Theorem 3.1, there are six possible routing paths between any two nodes. The length of the shortest path among these six paths (if exists) is the distance between these two nodes. Let D, E, and F represent three distinct sequences assigned under different situations in the following, and X denote a sequence of all don't-care symbols. Now, consider the routing from $(DEF,\overline{D}\overline{E}F)$ to $(\overline{D}E\overline{F},D\overline{E}\overline{F})$. Since $C^* = XXF$ and DEF matches C^*, P_δ is of length $(|P_\gamma|-1)$; therefore, we omit the value of $|P_\gamma|$. Additionally, P_λ does not exist because $A_1 \ne \overline{B}_1$. So, we only evaluate the lengths of four paths P_α, P_β, P_δ and P_μ.

(1) $n=3k$: Let D, E, and F represent three k-bit binary sequences. The lengths of the four possible paths are indicated below.

$$|P_\alpha| = 2k+1+2k = 4k+1,$$

$$|P_\beta| = 1+2k+1+2k = 4k+2,$$

$$|P_\delta| = 2k+1+k+1+k = 4k+2,$$

$$|P_\mu| = k+1+k+1+2k = 4k+2.$$

The shortest path among these four paths is of length $4k+1$; hence, the diameter of HCN$(3k,3k)$ is not less than $4k+1$.

(2) $n=3k+1$: Let D and F be two k-bit binary sequences, and E be a $(k+1)$-bit binary sequence. Similar to the preceding discussion, the diameter of HCN$(3k+1,3k+1)$ is not less than $4k+2$.

(3) $n = 3k + 2$: Let D be a k-bit binary sequence, E and F be two $(k+1)$-bit binary sequences. Similarly, the diameter of HCN$(3k+2, 3k+2)$ is not less than $4k+4$. □

The difference between these two bounds in Theorems 3.2 and 3.3 is at most one, implying that the upper bound for the diameter in Theorem 3.2 is very close to the exact value.

3.3. Average distance bound

As the diameter reflects only the worst case communication time, the average distance conveys the actual performance of the network in practice. The average distance in a symmetric network is defined as the ratio of the sum of the distances of all its nodes from a given node to the total number of nodes. The value of this measure for the n-cube is equal to the following:

$$\frac{1}{2^n}\sum_{k=0}^{n}(k)\binom{n}{k}=\frac{n}{2},$$

where

$$\binom{n}{k}$$

is the number of nodes at distance k from each hypercube node.

Since the HCN(n,n) is asymmetric, the average distance of HCN(n,n) is much more difficult to obtain than the diameter is. Like the discussion over diameter, we would like to find an upper bound for the average distance of HCN(n,n).

Theorem 3.4. *The average distance in an HCN(n,n) is less than or equal to $(n+1)$.*

Proof. Let A_1A_0 be the reference node and X be a sequence of n don't-care symbols. The length of the shortest path from A_1A_0 to one of the nodes within cluster B_1X is at most $\mathrm{H}(A_0,B_1)+1$, since there exists a path $P: A_1A_0 \Rightarrow A_1B_1 \rightarrow B_1A_1$. Moreover, the mean value of distances from node B_1A_1 to all the nodes within cluster B_1X is $n/2$ because each cluster is an n-cube. Therefore, the total sum of the distances from A_1A_0 to all the nodes of cluster B_1X is at most $(2^n)(\mathrm{k}+(n/2)+1)$ where $k=\mathrm{H}(A_0,B_1)$. An upper bound of the average distance in an HCN(n,n) is given by:

$$\frac{1}{2^{2n}}\sum_{k=0}^{n}(2^n)\binom{n}{k}\left(k+\frac{n}{2}+1\right)=\frac{1}{2^{2n}}(2^n)\left\{\sum_{k=0}^{n}k\binom{n}{k}+\left(\frac{n}{2}+1\right)\sum_{k=0}^{n}\binom{n}{k}\right\}=(n+1).\quad □$$

4. Optimal broadcasting

One of the common processes in parallel computer systems is the sending of a message from one node of a network to all the other nodes as quickly as possible. Now, we consider the problem of broadcasting on the

HCN under the following constraint: during each unit of time a node which already knows the message can only inform one of the nodes to which it is connected directly by an edge.

Since initially only one node has the message to be broadcasted, and at any point in time an additional step in the broadcasting will at most double the number of nodes that have received the message. As a result, at least $2n$ steps are necessary to complete the one-to-all broadcasting since the HCN(n,n) has 2^{2n} nodes. However, the degree of each node in an HCN(n,n) is $(n+1)$, which is less than $2n$. Hence, we will consider the problem of minimum time for broadcasting on the HCN(n,n) carefully.

In [4], Bermond et al. proposed general lower bounds on the time required to broadcast in bounded degree graphs. Let b(G,d) denote the broadcast time on a network G of degree d. Let m(d,t) denote the maximum number of nodes that can be informed in time smaller than or equal to t. A lower bound on b(G,d) can be obtained from the calculation of an upper bound on m(d,t). That is, $\mathrm{m}(d,t) < |G|$ if and only if $\mathrm{b}(G,d) > t$, where $|G|$ denotes the number of nodes in G. To achieve this maximum m(d,t), we may assume that a node does not remain idle if it has already been informed, but has not yet informed all of its neighbours. Hence, if there is one informed node at time $t=0$, then after time $t+d-1$, all the nodes that were informed by time t have informed all of their neighbours and must become idle. Thus, an upper bound on m(d,t) is the solution of the following recurrence:

$$\mathrm{m}(d,t) = 2^t \quad \text{for } 0 \le t \le d,$$

$$\mathrm{m}(d,t) = 2\mathrm{m}(d,t-1) - \mathrm{m}(d,t-d-1) \quad \text{for } t > d.$$

Theorem 4.1. *The broadcasting time for an HCN(n,n) is at least* $(2n+1)$.

Proof. Based on the preceding analysis, a lower bound of the broadcast time on the HCN(n,n) is evaluated as follows. Since the HCN(n,n) with 2_n^2 nodes has degree $(n+1)$, we have the following equations:

$$\mathrm{m}(n+1,2n) = 2^{2n} - (n-1)^{n-2} < 2^{2n},$$

and

$$\mathrm{m}(n+1,2n+1) = 2^{2n+1} - n2^{n-1} > 2^{2n}.$$

Therefore, the value of this bound is $(2n+1)$. □

Now, we present an optimal algorithm of complexity $(2n+1)$ for broadcasting in an HCN(n,n). The basic idea of our broadcasting algorithm in the HCN(n,n) is stated as follows. Assume the source node is A_1A_0. Our algorithm consists of two phases. The objective of the first phase is to broadcast the message from A_1A_0 to all other nodes of cluster A_1X, just like the broadcasting in an n-cube. When all the nodes of cluster A_1X have already received the message, these nodes other than A_1A_1 immediately transfer the message to nodes in other clusters through the switch edges. It is clear that each cluster of the HCN(n,n) contains at least one node being informed the message after time $(n+1)$. In the second phase, the informed node in each cluster broadcasts the message to other uninformed nodes as does the first phase. Notice that the complement edges are not used to broadcast the message in our scheme.

```
Algorithm Distributed-Broadcast(msg,count);
{* This algorithm is invoked by each informed node. *}
{* Set count = 0 in the source node initially. *}
    case count ≤ n:
        while count < n do
            count := count + 1;
            d := count − 1;
            Send [msg,count] along the edge of dimension d;
        endwhile;
        if there is a switch edge incident to the current node
            then count := count + 1; Send [msg,count] through the switch edge;
    case count ≥ (n + 1):
        while count < 2n + 1 do
            count := count + 1;
            d := count − (n + 2);
            Send [msg,count] along the edge of dimension d;
        endwhile;
end.
```

Fig. 4. A distributed broadcasting algorithm for an HCN(n,n).

Formally, a distributed algorithm for broadcasting on the HCN(n,n) is given in Fig. 4. There are two arguments in this algorithm which are denoted as *msg* representing the message to be broadcast, while *count* indicates the number of time step in the broadcasting.

5. Node-disjoint paths and fault-diameter

A set of paths is said to be node-disjoint if no node except the source node and the destination node appears in more than one path. It is important to have node-disjoint paths between any two nodes in an interconnection network to speed up transfers of large amounts of data and provide alternative routes in cases of node and/or link failures. Furthermore, node-disjoint paths admit very robust communication, because a message sent along several node-disjoint paths can arrive intact even though some paths have faults that block the message or that even alter the message. In this section, we address the problem of constructing node-disjoint paths between any two nodes in an HCN(n,n).

The following result related to hypercubes, which was previously addressed in [14], is useful for our discussion that follows.

Theorem 5.1. [14] *Let A and B be two nodes in an n-cube such that* $\mathrm{H}(A,B) = k$. *There are n node-disjoint paths between them. These paths are composed of k paths of length k, and* $(n-k)$ *paths of length* $(k+2)$. *Therefore, there are n node-disjoint paths of length* $\leq n+1$ *between any source-destination pair in an n-cube.*

To construct $(n+1)$ node-disjoint paths between node $A_1 A_0$ and node $B_1 B_0$ in an HCN(n,n), we consider two cases: (1) both nodes within a cluster, i.e., $A_1 = B_1$, and (2) two nodes belonging to distinct clusters, i.e., $A_1 \neq B_1$.

Lemma 5.1. *There are $(n+1)$ node-disjoint paths of length at most $(n+5)$ between any two nodes within a cluster of an HCN(n,n).*

Proof. Given two nodes A_1A_0 and A_1B_0, based on Theorem 5.1, there are n node-disjoint paths of length $\leq n+1$ between them through only the nodes of cluster A_1X.

(a) In the case of $A_1 \neq A_0$ and $A_1 \neq B_0$: The $(n+1)$-th path can be constructed as

$$P_s: A_1A_0 \rightarrow A_0A_1 \Rightarrow A_0B_0 \rightarrow B_0A_0 \Rightarrow B_0A_1 \rightarrow A_1B_0,$$

or

$$P_t: A_1A_0 \rightarrow A_0A_1 \rightarrow A_0A_1^i \rightarrow A_1^iA_0 \Rightarrow A_1^iB_0 \rightarrow B_0A_1^i \rightarrow B_0A_1 \rightarrow A_1B_0,$$

where

$$|P_s| \leq 2n - \mathrm{H}(A_0,B_0) + 3 \text{ and } |P_t| \text{ is } \mathrm{H}(A_0,B_0) + 6.$$

We choose the shorter between P_s and P_t as the $(n+1)$-th path; as a result, the length of the $(n+1)$-th path is at most $\lfloor(|P_s|+|P_t|)/2\rfloor \leq \lfloor(2n+9)/2\rfloor = n+4$.

(b) In the case of $A_1 = A_0$ and $A_1 \neq B_0$: The $(n+1)$-th path can be constructed as

$$P_u: A_1A_1 \rightarrow \overline{A}_1\overline{A}_1 \Rightarrow \overline{A}_1B_0 \rightarrow B_0\overline{A}_1 \Rightarrow B_0A_1 \rightarrow A_1B_0,$$

or

$$P_v: A_1A_1 \rightarrow \overline{A}_1\overline{A}_1 \rightarrow \overline{A}_1\overline{A}_1^i \rightarrow \overline{A}_1^i\overline{A}_1 \rightarrow \overline{A}_1^i\overline{A}_1^i \rightarrow A_1^iA_1^i \Rightarrow A_1^iB_0 \rightarrow B_0A_1^i \rightarrow B_0A_1 \rightarrow A_1B_0,$$

where $|P_u| = 2n+3-\mathrm{H}(A_1,B_0)$, and $|P_v| = \mathrm{H}(A_1,B_0)+7$ since there is some i such that $\mathrm{H}(A_1^i,B_0) = \mathrm{H}(A_1,B_0) - 1$. Similar to (a), the shorter path between P_u and P_v is of length at most $(n+5)$. □

Lemma 5.2. *There are $(n+1)$ node-disjoint paths of length $\leq (2n+6)$ between any two nodes located in distinct clusters of HCN(n,n).*

Proof. Given two nodes A_1A_2 and B_1B_2 with $A_1 \neq B_1$, we construct $(n+1)$ node-disjoint paths between them as follows.

(a) If $A_0^i \neq B_0$ for all i, $0 \leq i < n$, we show $(n+1)$ node-disjoint paths P_j, $1 \leq j \leq n+1$ as follows.

For $1 \leq j \leq n$:

$$P_j: A_1A_0 \rightarrow A_1A_0^{j-1} \rightarrow A_0^{j-1}A_1 \Rightarrow A_0^{j-1}B_0^{j-1} \rightarrow B_0^{j-1}A_0^{j-1} \Rightarrow B_0^{j-1}B_1 \rightarrow B_1B_0^{j-1} \rightarrow B_1B_0.$$

For $j = n+1$: If $A_1 \neq A_0$ and $B_1 \neq B_0$,

$$P_{n+1}^1: A_1A_0 \rightarrow A_0A_1 \Rightarrow A_0B_0 \rightarrow B_0A_0 \Rightarrow B_0B_1 \rightarrow B_1B_0.$$

If $A_1 = A_0$ and $B_1 \neq B_0$,

$$P_{n+1}^2: A_1A_1 \rightarrow \overline{A}_1\overline{A}_1 \Rightarrow \overline{A}_1B_0 \rightarrow B_0\overline{A}_1 \Rightarrow B_0B_1 \rightarrow B_1B_0.$$

If $A_1 \neq A_0$ and $B_0 = B_1$,

$$P_{n+1}^3: A_1A_0 \rightarrow A_0A_1 \Rightarrow A_0\overline{B}_1 \rightarrow \overline{B}_1A_0 \Rightarrow \overline{B}_1\overline{B}_1 \rightarrow B_1B_1.$$

If $A_1 = A_0$ and $B_1 = B_0$,

$$P_{n+1}^4 : A_1 A_1 \to \bar{A}_1 \bar{A}_1 \Rightarrow \bar{A}_1 \bar{B}_1 \to \bar{B}_1 \bar{A}_1 \Rightarrow \bar{B}_1 \bar{B}_1 \to B_1 B_1.$$

(b) If $A_0^i = B_0$ for some i, $0 \le i < n$, P_i and P_{n+1}^1 have the common nodes $A_0 B_0$ and $B_0 A_0$. We need another path P_i^1 to replace P_i:

$$P_i^1 : A_1 A_0 \to A_1 A_0^i \to A_0^i A_1 \Rightarrow A_0^i B_0^m \to B_0^m A_0^i \to B_0^m A_0 \to A_0 B_0^m \Rightarrow A_0 B_1 \to B_1 A_0 \to B_1 B_0.$$

Now, the length of each of these paths is listed below:

$$|P_j| = \mathrm{H}(A_1, B_0^{j-1}) + \mathrm{H}(A_0^{j-1}, B_1) + 5 \le 2n + 5.$$

$$|P_{n+1}^1|, |P_{n+1}^2|, |P_{n+1}^3|, |P_{n+1}^4| \le 2n + 3.$$

$$|P_i^1| = \mathrm{H}(A_1, B_0^m) + \mathrm{H}(B_0^m, B_1) + 7 \le 2n - \mathrm{H}(A_1, B_1) + 7 \le 2n + 6.$$

The longest path among the above paths is of length at most $(2n + 6)$. □

From the results of Lemmas 5.1 and 5.2, we have the following theorem.

Theorem 5.2. *There are $(n+1)$ node-disjoint paths of length $\le (2n+6)$ between two arbitrary nodes in an HCN(n,n).*

For any network, the number of node-disjoint paths between any two nodes is bounded from above by the minimum degree. The HCN(n,n) of degree $(n+1)$ has $(n+1)$ node-disjoint paths between two arbitrary nodes; as a result, the HCN(n,n) is said to be optimal fault tolerant.

The node-connectivity κ of a network G is defined as the minimum number of nodes whose removal results in a disconnected or trivial network. The fault-diameter of a network G with connectivity κ is defined as the maximum diameter of any network obtained from G by removing $(\kappa - 1)$ nodes. Due to Menger's theorem [5]: a network G has connectivity κ if and only if every pair of nodes in the network is connected by at least κ node-disjoint paths. We obtain the following corollaries.

Corollary 5.1. *The node connectivity of an HCN(n,n) is $(n+1)$.*

Corollary 5.2. *The fault-diameter of an HCN(n,n) is less than or equal to $(2n+6)$.*

6. Hamiltonian

In this section we prove the existence of a Hamiltonian cycle in an HCN(n,n) by illustrating how to construct it. Before proving that the HCN is Hamiltonian, we first note that each cluster of the HCN(n,n) is an n-cube. The n-cube is known to have a Hamiltonian cycle. One way to find a Hamiltonian cycle in an n-cube is to generate the n-bit Gray code G_n.

There are many different ways in which Gray codes can be generated, but the best known method to generate G_n, called binary reflected Gray code [13], is as follows.

G_n is an n-bit Gray code obtained by the recursion:

$$G_1 = \{0,1\}, \text{ and } G_i = \{0G_{i-1}, 1G_{i-1}^R\},$$

where G_i^R is the sequence obtained by reversing the order of the numbers of G_i, and $0G_i$ ($1G_i$) is the sequence obtained by prepending 0(1) to each element of the sequence of G_i. For example, $G_2 = \{00,01,11,10\}$ and $G_3 = \{000,001,011,010,110,111,101,100\}$.

Consider G_n as an ordered set, i.e. $G_n = \{g_i \mid i = 1,2,\ldots,2^n\}$ where g_i denotes the i-th element in G_n, $1 \le i \le 2^n$. Any element g_i in G_n can be selected as the starting number of a new sequence, and then the resultant sequence is still a Hamiltonian path of the n-cube whether the order of the sequence is clockwise or counterclockwise. Therefore, we use $S_n^+(i)$ and $S_n^-(i)$ to denote these two sequences independently. That is, $S_n^+(i) = (g_i, g_{i+1}, \ldots, g_{2^n}, g_1, \ldots, g_{i-1})$ and $S_n^-(i) = (g_i, g_{i-1}, \ldots, g_1, g_{2^n}, \ldots, g_{i+1})$.

For the purpose of clarity, we use (A_1, A_0) to denote node $A_1 A_0$, symbol $\leftrightarrow$ to denote a switch edge, and symbol – to denote a complement edge. For brevity, let $[g_i, S_n^+(j)]$ and $[g_i, S_n^-(j)]$ represent two Hamiltonian paths in cluster (g_i, X) separately, where $[g_i, S_n^+(j)] = [(g_i, g_j)(g_i, g_{j+1})\ldots(g_i, g_{j-1})]$ and $[g_i, S_n^-(j)] = [(g_i, g_j)(g_i, g_{j-1})\ldots(g_i, g_{j+1})]$.

Lemma 6.1. *An HCN(n,n) has a Hamiltonian path for any $n \ge 1$.*

Proof. We construct the sequence of a Hamiltonian path in an HCN(n,n) as follows:

$$\begin{aligned}&[g_1, S_n^+(1)] \leftrightarrow [g_{2^n}, S_n^-(1)] \leftrightarrow [g_2, S_n^+(2^n)] \leftrightarrow [g_{2^n-1}, S_n^-(2)] \leftrightarrow \ldots \leftrightarrow [g_i, S_n^+(2^n - i + 2)]\\ &\quad\leftrightarrow [g_{2^n-i+1}, S_n^-(i)] \leftrightarrow \ldots \leftrightarrow [g_{2^{n-1}}, S_n^+(2^{n-1}+1)] \leftrightarrow [g_{2^{n-1}+1}, S_n - (2^{n-1})].\end{aligned}$$ □

Example 6.1. A Hamiltonian path of an HCN(2,2) is constructed by a sequence of nodes as follows:

$$\begin{aligned}&[(00,00)(00,01)(00,11)(00,10)] \leftrightarrow [(10,00)(10,10)(10,11)(10,01)]\\ &\quad\leftrightarrow [(01,10)(01,00)(01,01)(01,11)]\\ &\quad\leftrightarrow [(11,01)(11,00)(11,10)(11,11)].\end{aligned}$$

Theorem 6.1. *An HCN(n,n) is Hamiltonian for any $n \ge 1$.*

Proof. Clearly, an HCN(1,1) is Hamiltonian, since $[(0,0)(0,1)] \leftrightarrow [(1,0)(1,1)]$ form a cycle. We can construct a Hamiltonian cycle of an HCN$(n+1, n+1)$ for $n \ge 1$ by using the sequence of the Hamiltonian path of an HCN(n,n) in Lemma 6.1. The construction of the Hamiltonian cycle is divided into three steps as shown below.

Step 1. Insert a zero in front of each element of the sequence in the proof of Lemma 6.1. That is,

$$[0g_1, 0S_n^+(1)] \leftrightarrow [0g_{2^n}, 0S_n^-(1)] \leftrightarrow \ldots \leftrightarrow [0g_{2^{n-1}}, 0S_n^+(2^{n-1}+1)] \leftrightarrow [0g_{2^{n-1}+1}, 0S_n^-(2^{n-1})].$$

The number of nodes in this sequence is 2^{2n}, the same as that in an HCN(n,n). Note that symbol $\leftrightarrow$ is used here to separate consecutive subsequences.

Step 2. Enlarge each subsequence of the sequence in Step 1. Consider a subsequence $[(0g_i, 0h_1)(0g_i, 0h_2)(0g_i, 0h_3)(0g_i, 0h_4)\ldots(0g_i, 0h_{2^n})]$, where g_i and h_j are n-bit binary sequences. For each odd j, the edge between $(0g_i, 0h_j)$ and $(0g_i, 0h_{j+1})$ is replaced with a path of length 3: $(0g_i, 0h_j)(0g_i, 1h_j)(0g_i, 1h_{j+1})(0g_i, 0h_{j+1})$. After enlarging all the subsequences, the resulting path, called

0-path, travels all the nodes in clusters $(0g_i, X)$, $i = 1,2,\ldots,2^n$ of an HCN(n,n) once and exactly once, and its length is 2^{2n+1}, two times the length of the original sequence in Step 1.

Step 3. Obtain the 1-path by complementing the label of each node in the 0-path. After complementing the labels of all the nodes in the 0-path, the new sequence, called 1-path, travels all the nodes in clusters $(1\bar{g}_i, X)$, $i = 1,2,\ldots,2^n$. Furthermore, the first node $(0g_1, 0g_1)$ and the last node $(0g_{2^{n-1}+1}, 0g_{2^{n-1}+1})$ of the 0-path are connected directly with the first node $(1\bar{g}_1, 1\bar{g}_1)$ and the last node $(1\bar{g}_{2^{n-1}+1}, 1\bar{g}_{2^{n-1}+1})$ of the 1-path through complement edges, respectively; as a result, these two paths with the two complement edges form a Hamiltonian cycle in an HCN$(n+1, n+1)$. □

Example 6.2. A Hamiltonian cycle of an HCN(3,3) can be constructed as observed from the proof of Theorem 6.1. First, by prepending 0 to each element of the Hamiltonian path in the HCN(2,2) from Example 6.1, we have the following sequence:

$$
\begin{aligned}
&[(000,000)(000,001)(000,011)(000,010)] \leftrightarrow [(010,000)(010,010)(010,011)(010,001)] \\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad \leftrightarrow [(001,010)(001,000)(001,001)(001,011)] \\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad \leftrightarrow [(011,001)(011,000)(011,010)(011,011)].
\end{aligned}
$$

Second, enlarge the sequence so that the resulting path (0-path) travels clusters $(000, X)$, $(010, X)$, $(001, X)$ and $(011, X)$. That is, the 0-path is

$$
\begin{aligned}
&[(000,000)(000,100)(000,101)(000,001)(000,011)(000,111)(000,110)(000,010)] \\
&\leftrightarrow [(010,000)(010,100)(010,110)(010,010)(010,011)(010,111)(010,101)(010,001)] \\
&\leftrightarrow [(001,010)(001,110)(001,100)(001,000)(001,001)(001,101)(001,111)(001,011)] \\
&\leftrightarrow [(011,001)(011,101)(011,100)(011,000)(011,010)(011,110)(011,111)(011,011)].
\end{aligned}
$$

Third, by complementing the sequence of the 0-path, we obtain a new path (1-path) which travels clusters $(111, X)$, $(101, X)$, $(110, X)$ and $(100, X)$ as listed below:

$$
\begin{aligned}
&[(111,111)(111,011)(111,010)(111,110)(111,100)(111,000)(111,001)(111,101)] \\
&\leftrightarrow [(101,111)(101,011)(101,001)(101,101)(101,100)(101,000)(101,010)(101,110)] \\
&\leftrightarrow [(110,101)(110,001)(110,011)(110,111)(110,110)(110,010)(110,000)(110,100)] \\
&\leftrightarrow [(100,110)(100,010)(100,011)(100,111)(100,101)(100,001)(100,000)(100,100)].
\end{aligned}
$$

With complement edges, (000,000) is connected to (111,111), and (011,011) is connected to (100,100). Therefore, these two paths (0-path and 1-path) with the two complement edges form a Hamiltonian cycle in an HCN(3,3).

7. Emulation of hypercubes on HCNs

A very important property for new networks would be the emulation of the hypercube with a small degradation in time performance so that all the algorithms designed for the hypercube can be executed on the new network with a small degradation in time performance. For the purpose of evaluating the degradation in performance, the dilation of edges associated with such a hypercube mapping must be found.

Given a mapping M of the guest graph G onto the host graph H, the dilation of the edge connecting the two nodes u and v in G is defined as the distance between the two nodes $M(u)$ and $M(v)$ in H.

Let a $2n$-cube be the guest graph and an HCN(n,n) be the host graph. They consist of the same number of nodes. Assume that nodes from the hypercube are mapped to nodes of the HCN with the same address; that is, node $A_1 A_0$ in the $2n$-cube is mapped to node $A_1 A_0$ in the HCN(n,n), where A_1 and A_0 denote two n-bit binary sequences. The dilation measures the increase of the communication overhead in the HCN when compared to one-hop data transfers in the hypercube. The following theorem presents the resultant dilation of edges.

Theorem 7.1. *For the emulation of a $2n$-cube on an HCN(n,n), the dilations of the edges incident to node $A_1 A_0$ of the hypercube are:*

1 for n of them, and 2 for n of them	*if* $A_1 = A_0$;
1 for n of them, 2 for 1 of them, and 3 for $(n-1)$ of them	*if* $A_1 = A_0^i$;
1 for n of them, and 3 for n of them	*if* $A_1 \neq (A_0 \text{ or } A_0^i)$,

where A_1 and A_0 represent two n-bit binary sequences.

Proof. There is an edge between two nodes in the hypercube if their addresses differ in exactly one bit position. An edge is of dimension j if it connects two nodes that differ in the j-th bit. Each node in the hypercube connects $2n$ adjacent nodes with edges of dimensions j, $j = 0,1,\ldots,(2n-1)$.

First, edges of dimensions j, $0 \le j < n$, in the $2n$-cube are retained in the HCN(n,n). As a result, the dilations of these edges are equal to 1. Then, we consider the dilations of the other edges of dimensions j, $n \le j < 2n$, in the hypercube. Since these edges are absent in the HCN(n,n), they are emulated in different cases as follows.

Case 1. Any edge of dimension $(n+j)$, $0 \le j < n$, incident to node $A_1 A_1$ in the $2n$-cube, connecting $A_1 A_1$ and $A_1^j A_1$, can be emulated in the HCN(n,n) as a path of length 2:

$$A_1 A_1 \to A_1 A_1^j \to A_1^j A_1.$$

Case 2. Any edge of dimension $(n+j)$, $0 \le j < n$, and $j \neq i$, incident to node $A_1 A_1^i$ in the $2n$-cube, connecting $A_1 A_1^i$ and $A_1^j A_1^i$, can be emulated in the HCN(n,n) as a path of length 3:

$$A_1 A_1^i \to A_1^i A_1 \to A_1^i A_1^j \to A_1^j A_1^i.$$

However, the edge of dimension $(n+i)$, connecting $A_1 A_1^i$ and $A_l i A_1^i$, can be emulated as a path of length 2:

$$A_1 A_1^i \to A_1^i A_1 \to A_1^i A_1^i.$$

Case 3. Any edge of dimension $(n+j)$, $0 \le j < n$, incident to node $A_1 A_0$ with $A_1 \neq (A_0$ or $A_0^i)$ in the $2n$-cube, connecting $A_1 A_0$ and $A_1^j A_0$, can be emulated in the HCN(n,n) as a path of length 3:

$$A_1 A_0 \to A_0 A_1 \to A_0 A_1^j \to A_1^j A_0.$$

The following two corollaries give the maximum and average dilations of edges for hypercube emulation. The average dilation of edges for hypercube emulation is defined as the ratio of the sum of the dilations of all the edges in the hypercube to the total number of edges in the hypercube.

Corollary 7.1. *The largest dilation of edges for hypercube emulation on an HCN(n,n) is* 3.

Corollary 7.2. *The average dilation of edges for hypercube emulation on an HCN(n,n) is*

$$2-\frac{1}{2^n}.$$

Proof. Let S_1 be the set of nodes labeled A_1A_0 with $A_1=A_0$, S_2 the set of nodes labeled A_1A_0 with $A_1=A_0^i$, and S_3 the set of nodes labeled A_1A_0 with $X\neq(A_0$ or $A_0^i)$. Let $|S_i|$ denote the number of nodes in S_i. The sum of the dilations of edges incident to each node in S_1 is $3n$; that incident to each node in S_2 is $(4n-1)$; that incident to each node in S_3 is $4n$.

Since the dilation of each edge is counted twice, the total sum of the dilations of all the edges is equal to the following:

$$\frac{1}{2}\{|S_1|\times(3n)+|S_2|\times(4n-1)+|S_3|\times(4n)\}$$

$$=\frac{1}{2}\{(2^n)\times(3n)+(2^n\times n)\times(4n-1)+(2^{2n}-2^n-2^n\times n)\times(4n)\}=2n\times2^{2n}-n\times2^n.$$

Therefore, the average dilation of edges for hypercube emulation on an HCN(n,n) is equal to

$$\frac{2n\times2^{2n}-n\times2^n}{n\times2^{2n}}=2-\frac{1}{2^n}.\quad\square$$

The observation that the average dilation of edges for hypercube emulation is bounded by 2 guarantees small performance degradation for the implementation of hypercube algorithms on HCNs.

8. Conclusion

We have investigated topological properties of the HCN and proposed a shortest-path routing algorithm and an optimal broadcasting algorithm for the HCN. The diameter of HCN is about two-third of that of the comparable hypercube with the same number of nodes. The connectivity of an HCN(n,n) is $(n+1)$, and its fault-diameter is at most $(2n+6)$. Moreover, the HCN is Hamiltonian like the hypercube. Although the HCN has fewer edges than the comparable hypercube, the degradation for data communication performance is better than one may expect. The HCN is shown to emulate the hypercube with dilation 3, a small degradation in time performance. Therefore, HCNs are appropriate candidates for the implementation of massively parallel systems. Our further research will focus on the investigation of embedding frequently used topologies into HCNs and the development of efficient application algorithms for HCNs.

References

[1] S. Abraham and K. Padmanabhan. Performance of the Direct Binary n-Cube Network for Multiprocessors. *IEEE Transactions on Computers* 38(7):1000–1011, July 1989.

[2] S.B. Akers, D. Horel and B. Krishnamurthy. The Star Graph: An Attractive Alternative to the n-Cube. In *Proceedings of the International Conference on Parallel Processing*, pages 393–400, 1987.

[3] S.G. Akl. *The Design and Analysis of Parallel Algorithms*. Prentice Hall, Englewood Cliffs, NJ, 1989.

[4] J.C. Bermond, P. Hell, A.L. Liestman and J.G. Peters. Broadcasting in Bounded Degree Graphs. *SIAM Journal of Discrete Mathematics* 5(1):10–24, Feb. 1992.

[5] J. A. Bondy and U.S.R. Murty. *Graph Theory with Applications*. North-Holland, New York, 1976.

[6] W.-K. Chiang and R.-J. Chen. The (n,k)-Star Graph: A Generalized Star Graph. *Information Processing Letters* 56:259–264, Dec. 1995.

[7] S.P. Dandamudi and D.L. Eager. On Hierarchical Hypercube Multicomputer Interconnection Networks Design. *Journal of Parallel and Distributed Computing* 12:283–289, July 1991.

[8] K. Ghose and K.R. Desai. The HCN: A Versatile Interconnection Network Based on Cubes. In *Proceedings of the International Conference on Supercomputers*, pages 426–435, 1989.

[9] K. Ghose and K.R. Desai. The Design and Evaluation of The Hierarchical Cubic Network. In *Proceedings of the International Conference on Parallel Processing, Vol. 1*, pages 355–362, 1990.

[10] K. Ghose and K.R. Desai. Hierarchical Cubic Networks. *IEEE Transactions on Parallel and Distributed Systems* 6:427–435, April 1995.

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

[12] Q.M. Malluhi and M.A. Bayoumi. The Hierarchical Hypercube: A New Interconnection Topology for Massively Parallel Systems. *IEEE Transactions on Parallel and Distributed Systems* 5:17–30, Jan. 1994.

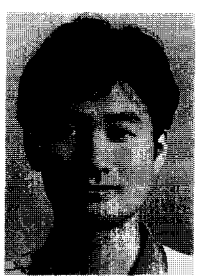
[13] E.M. Reingold, J. Nievergelf and N. Deo. *Combinatorial Algorithms*. Prentice Hall, Englewood Cliffs, NJ, 1977.

[14] Y. Saad and M.H. Schultz. Topological Properties of Hypercubes. *IEEE Transactions on Computers* 37:867–872, July 1988.

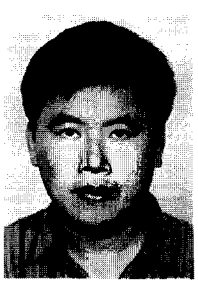
[15] C.L. Seitz. The Cosmic Cube. *Communications of the ACM* 28(1):22–33, Jan. 1985.

[16] H. Shen and R.-J. Back. Construction of Large-Size Interconnection Networks with High Performance. *Microprocessing and Microprogramming* 35:545–554, 1992.

[17] J. Wu and M.M. Larrondo-Petrie. Cube-Connected-Cubes Network. *Microprocessing and Microprogramming* 33:299–310, 1991/92.



Wei-Kuo Chiang was born in Taiwan in 1967. He received his B.S. and M.S. degrees in Computer Science and Information Engineering from National Chiao-Tung University, Hsinchu, Taiwan, in 1989 and 1991, respectively. He is currently a Ph.D. candidate in the Department of Computer Science and Information Engineering at National Chiao-Tung University, Taiwan. His research interests include interconnection networks, parallel processing, parallel architectures, graph theory.



Rong-Jaye Chen was born in Taiwan in 1952. He received his B.S. degree in Mathematics from National Tsing-Hua University, Taiwan, in 1977, and his Ph.D. degree in Computer Science from University of Wisconsin-Madison, in 1987. He is now a professor in the Department of Computer Science and Information Engineering at National Chiao-Tung University, Taiwan. Professor Chen is a member of IEEE. His research interests include network optimization, parallel computation, algorithm design.