JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 16, 291-300 (2000)

Short Paper

Incrementally Extensible Folded Hypercube Graphs

HUNG-YI CHANG AND RONG-JAYE CHEN
Department of Computer Science and Information Engineering
National Chiao Tung University
Hsinchu, Taiwan 300, R.O.C.

In this paper, we propose the incrementally extensible folded hypercube (IEFH) graph as a new class of interconnection networks for an arbitrary number of nodes. We show that this system is optimal fault tolerant and almost regular (i.e., the difference between the maximum and the minimum degree of nodes is at most one). The diameter of this topology is half of that of the incomplete hypercube (IH), the supercube, or the IEH graph. We also devise a simple routing algorithm for the IEFH graph. Finally, we embed cycles and complete binary trees into this graph optimally.

Keywords: hypercubes, folded hypercubes, incrementally extensible hypercubes, fault tolerance, graph embedding, interconnection networks

1. INTRODUCTION

In the research on interconnection networks, systems are modeled as graphs. In these graphs, nodes represent processors, and edges represent communication channels. Many topologies have been proposed for massively parallel machines-which include hypercubes, meshes, rings, star graphs, etc. Among these topologies, the hypercube is popular due to its properties, such as regularity, symmetry, low diameter, a simple routing algorithm, and optimal fault tolerance [7]. In recent years, several modifications have been proposed to enhance the performance of hypercubes. The folded hypercube (FH) [1], the crossed cube [4], and the Hierarchical Cubic Network (HCN) [5] are some examples of hypercube variants. They all are intended to reduce the diameter of the hypercube but reserve its advantages.

However, the hypercube and its variants have a common drawback in that the number of nodes must be a power of two. A few papers have so far been written to improve this drawback of hypercubes, but these approaches still have problems described briefly as follows. Katseff [6] proposed *incomplete hypercubes* (IHs), which suffer from the problem of fault tolerance–failure of a single node will cause the entire network to be disconnected. Sen [8] proposed *supercubes*, which become more irregular as the size of the network grows. Recently, Sur and Srimani [9] proposed a new generalization class of hypercube graphs, the

Received January 13, 1999; revised August 20, 1999; accepted November 26, 1999.
Communicated by Shang-Rong Tsai.

incrementally extensible hypercube (IEH) graph. This topology can be defined for an arbitrary number of nodes and still has several advantages, such as optimal fault tolerance, low diameter, a simple routing algorithm, and near regularity.

We observe that the diameters of these three incrementally extensible systems are about $n + 1$ when the sizes of these networks range from 2^n to 2^{n+1}. This motivated us to design a new incrementally extensible topology to satisfy the demand for further reduction of communication delay. In this paper, we introduce the *incrementally extensible folded hypercube* (IEFH) graph. Its diameter is half of that of IH, the supercube, or the IEH graph. The proposed network combines the ideas behind by IEH graphs and folded hypercubes. The basic concept is that we replace hypercubes with folded hypercubes in IEH graphs. We will show that the IEFH graph is incrementally extensible in steps of one, are almost regular, and are optimal fault tolerant. We have also devised a simple routing algorithm. We will show that the IEFH graph is Hamiltonian. Further, we can embed complete binary trees into this topology with adjacent edges reserved.

The rest of this paper is organized as follows. In section 2, we introduce basic terminology for the hypercube and the folded hypercube. We also review the structure of IEH graphs. In section 3, we present an algorithm used to construct an IEFH graph and derive some of its properties. In section 4, we present a simple routing algorithm for the IEFH graph. Thus, we obtain an upper bound of its diameter. In section 5, we embed cycles and complete binary trees into it. Finally, we give some conclusions in section 6.

2. PRELIMINARIES

A hypercube H_n (n-cube) is a graph $G(V, E)$, where V is the set of 2^n nodes which are labeled as binary numbers of length n; E is the set of edges that connects two nodes if and only if they differ in exactly one bit of their labels. A bit is said to be bit i if it is the ith bit with the least significant bit being bit 0. A link is said to be link i if it connects two nodes whose labels differ in bit i. $\mathbf{H}(x, y)$ is the number of bits in which node x and node y differ and denotes the *Hamming Distance* between them. The *node-connectivity* κ of a graph G is the minimum number of nodes whose removal results in a disconnected or trivial graph. This graph is said to be a *κ-connected* graph. An n-dimensional Folded Hypercube (FH_n) [1] is an n-cube with 2^{n-1} extra edges: each node $b_{n-1}...b_0$ has an extra edge to its complement node, i. e., $\overline{b_{n-1}}\cdots\overline{b_0}$.. Thus, these edges are referred as *complementary* edges. Fig. 1(a) shows an H_3 and (b) shows an FH_3 hypercube.

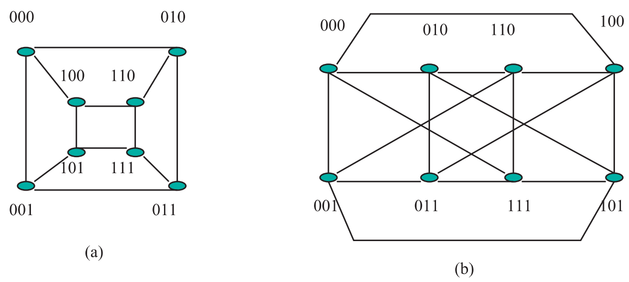

Fig. 1. An H_3 and an FH_3 hypercube.

An IEH graph, a generalized hypercube graph, is composed of several hypercubes of different sizes. These hypercubes are connected by *Inter-Cube* (IC) edges. Let an IEH(N) graph denote an IEH graph of N nodes. This graph can be constructed by the following algorithm [7].

Algorithm 1. (Construction of an IEH(N) graph)
Input: a positive integer N
Output: an IEH(N) graph

1. Express N as a binary number $(c_n, ..., c_1, c_0)_2$, where $c_n = 1$. For each c_i, with $c_i \neq 0$, construct a hypercube H_i. The edges constructed in this step are called *regular* edges.
2. For all H_is', label each node with a dedicated binary number $11...10b_{i-1}...b_0$, where the length of the leading 1s is $n - i$ and $b_{i-1}...b_0$ is the label of this node in the regular hypercube of dimension i.
3. Find the minimum i, where $c_i = 1$, set $G_j = H_i$, and set $j = i$:

$i = i + 1$.

While $i \leq n$
if $c_i \neq 0$ **then**
Connect the node $11...1b_jb_{j-1}...b_0$ in G_j to the following $i - j$ nodes in H_i:

$\overbrace{11\cdots1}^{n-i}0\overbrace{11\cdots1}^{i-j-1}\ b_jb_{j-1}...b_0$,

$\overbrace{11\cdots1}^{n-i}0\overbrace{01\cdots1}^{i-j-1}\ b_jb_{j-1}...b_0$,

...,

$\overbrace{11\cdots1}^{n-i}0\overbrace{11\cdots0}^{i-j-1}\ b_jb_{j-1}...b_0$.

Set $j = i$ and G_j to be the composed graph obtained in this step. /* G_j is the graph which is composed of H_ks' for $k \leq j$.*/
endif
$i = i + 1$.
endwhile
Thus, obtain G_n as the output.#

For the purpose of illustration, Fig. 2 shows an IEH(11) graph. Note that solid lines represent regular edges, and that dot lines represent IC edges.

3. THE IEFH GRAPH AND ITS PROPERTIES

In this section, we will describe IEFH graphs and derive some of their properties. An IEFH graph is composed of several folded hypercubes of different sizes. The basic philosophy in designing IEFH graphs is similar to that for IEH graphs. However, if we directly use Algorithm 1 to connect these folded hypercubes, the obtained graph will not be almost regular. For example, Fig. 3 shows a graph of size 11, whose FHs were connected by Algorithm 1. Observe that the degree of node 1100 is three, and that that of node 0000 is five. Hence, some extra edges are needed in order to maintain the property of near regularity. In the following algorithm, we add at most two edges to overcome this drawback. Further, the node connectivity can be improved from n to $n + 1$.

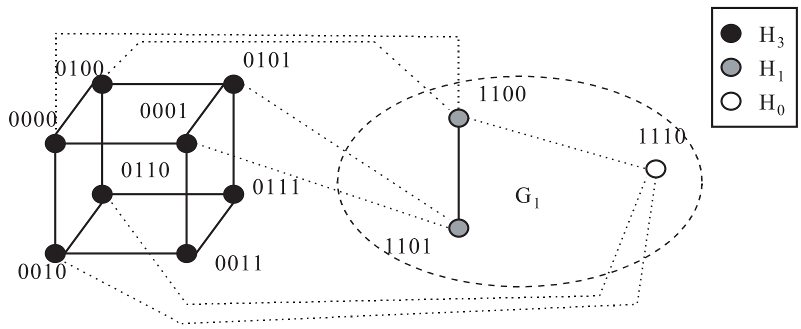

Fig. 2. An IEH(11) graph.

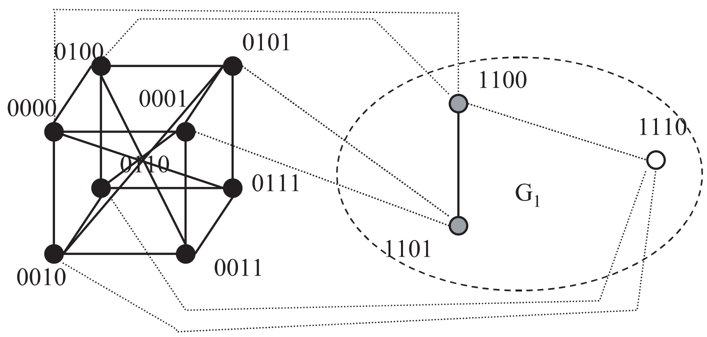

Fig. 3. The obtained graph of size 11.

Algorithm 2. (Construction of an IEFH(N) graph)
Input: a positive integer N
Output: an IEFH(N) graph
1. Express N as a binary number $(c_n, ..., c_1, c_0)_2$, where $c_n = 1$. For each c_i, with $c_i \neq 0$, construct a folded hypercube FH_i.
2. For all FH_is', label each node with a dedicated binary number $11...10b_{i-1}...b_0$, where the length of the leading 1s is $n - i$ and $b_{i-1}...b_0$ is the label of this node in the regular folded hypercube of dimension i.
3. If ($c_1c_0 = 00$), then goto Step 4.
   else if ($c_1c_0 = 01$)
      find the minimum i, where $c_i = 1$ and $i > 1$,
      connect (11...10) to $(\overbrace{11\cdots10}^{n-i}\cdots01)$, and goto Step 4.
   else if ($c_1c_0 = 10$)
      find the minimum i, where $c_i = 1$ and $i > 1$,

connect (11...100) to $(\overbrace{11\cdots10}^{n-i}\cdots011)$,
connect (11...101) to $(\overbrace{11\cdots10}^{n-i}\cdots010)$, and goto Step 4.
else connect (11...10) and (11...101), and goto Step 4.

4. Find the minimum i, where $c_i = 1$, set $G_j = FH_i$, and set $j = i$:

$i = i + 1$.

While $i \le n$
if $c_i \ne 0$ **then**
Connect the node $11...1b_jb_{j-1}...b_0$ in G_j to the following $i - j$ nodes in FH_i:

$\overbrace{11\cdots10}^{n-i}\overbrace{11\cdots1}^{i-j-1}\, b_jb_{j-1}...b_0$,
$\overbrace{11\cdots10}^{n-i}\overbrace{01\cdots1}^{i-j-1}\, b_jb_{j-1}...b_0$,

...,
$\overbrace{11\cdots10}^{n-i}\overbrace{11\cdots0}^{i-j-1}\, b_jb_{j-1}...b_0$.

Set $j = i$ and G_j be the composed graph obtained in this step. /* G_j is the graph which is composed of FH_ks' for $k \le j$.*/
endif
$i = i + 1$.
endwhile
Thus, obtain G_n as the output.#

In Algorithm 2, we observe two useful properties. First, G_i is an IEFH($\sum_{k=0}^{i} c_k 2^k$) graph. Second, whenever we connect an FH_i and some G_j for $i > j$, the degree of each node of FH_i increases by at most one. Fig. 4 shows an IEFH(11) graph.

Theorem 1. The IEFH(N) graph is almost regular for $2^n < N \le 2^{n+1}$. Further, the degree of each node is either $n + 1$ or $n + 2$.

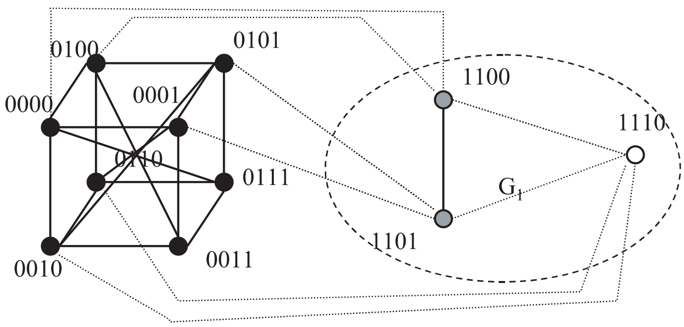

Fig. 4. An IEFH(11) graph.

Proof: Let $N = (c_n, ..., c_1, c_0)_2$, where $c_n = 1$. Without loss of generality, let v be a node in an FH_i, where $2 \leq i \leq n$. By Algorithm 2, v has $i + 1$ edges in FH_i and $n - i$ edges to FH_ks', where $c_k = 1$ and $n \geq k > i$. Furthermore, it has at most one edge from G_j. Here, j is the maximal integer where $c_j = 1$ and $j < i$. Hence, the proof. #

Next, we will derive the connectivity of an IEFH(N) graph and show it is optimal fault tolerant. The following lemmas are necessary.

Lemma 1. [9] In an m-connected graph, any node has m node-disjoint paths to an arbitrary set of m different nodes in the graph. #

Lemma 2. [9] Given an m-connected graph G and an n-connected graph H, $m < n$, if each node of G is connected to $(n - m)$ different nodes of H (assume $|H| \geq (n - m)|G|$), then the combined graph is also n-connected. #

Theorem 2. The node connectivity of an IEFH(N) graph is $n + 1$ for $2^n < N \leq 2^{n+1}$.

Proof: At the beginning of Steps 3 and 4 in Algorithm 2, we connect the two smallest FH_i and FH_j, where $i > j$ into G_i. By the previous lemmas, G_i is an $(i + 1)$-connected graph. Then, whenever we join FH_k with a G_l for $k > l$ by means of IC edges in Step 4 of Algorithm 2, G_k is a $(k + 1)$-connected graph. Hence, when our algorithm terminates, the IEFH(N) graph has a node connectivity of $n + 1$. #

Corollary 1. The IEFH graph is optimal fault tolerant.

Proof: Since the node connectivity is equal to the minimum degree of the graph, the IEFH graph is optimal fault tolerant. #

The degree of a topology is usually measured as the cost of hardware. Now, we will present the average degree of the IEFH graph. In the following results, we will derive the total number of edges and the average degree of an IEFH(N) graph.

Theorem 3. The total number of edges of an IEFH(N) graph, where $(c_n, ..., c_1, c_0)_2$ is the binary expression of N, is $\delta + \sum_{i=1}^{n} c_i (i+1)2^{i-1} + \sum_{i=0}^{n-1} c_i 2^i (n-i)$ for .

$$\delta = \begin{cases} 0 & if \quad c_1 c_0 = 00 \\ 2 & if \quad c_1 c_0 = 10. \\ 1 & otherwise \end{cases}$$

Proof: The first summation counts the total number of edges in all FH_s; each FH_i has $(i + 1)2^{i-1}$ edges. The second summation counts the number of IC edges; each FH_i has 2^i nodes, and each node has $(n - i)$ IC edges connecting to higher order FH_s by means of Step 4. Last, we consider (for different cases in Step 3 of Algorithm 2. Hence, the proof. #

Corollary 2. The average degree of an IEFH(N) graph is $2(\delta + \sum_{i=1}^{n} c_i (i+1)2^{i-1} + \sum_{i=0}^{n-1} c_i 2^i (n-i)) / N$, where $(c_n, ..., c_1, c_0)_2$ is the binary expression of N and

$$\delta = \begin{cases} 0 & if \quad c_1 c_0 = 00 \\ 2 & if \quad c_1 c_0 = 10 \\ 1 & otherwise \end{cases} \#.$$

Proof: In Theorem 3, we derive the total number of edges of an IEFH(N) graph. Since each edge joins two nodes, the total number of edges is half the number of total degrees in a graph. Hence the proof. #

4. A SIMPLE ROUTING ALGORITHM

Message routing is a central problem in an interconnection network. A good interconnection network should route messages in a simple and efficient way so as to increase the overall performance of the multiprocessor system built on this network. In this section, we will devise a simple and efficient routing algorithm for IEFH graphs.

Algorithm 3.(Routing in an IEFH(N) graph)
Input: a source node x in FH_i and a destination node y in FH_j (Without loss of generality, let $j \geq i$.)
Output: the next node in the routing path
If (x == y), then send the message to the local processor.
Else if (i == j), then route the message as in a folded hypercube of dimension i.
    Else route the message by means of an IC edge to a node z in FH_j such that
    $H(x, z) = H(x, y) - h$, where h is as large as possible. #

Theorem 4. The diameter of an IEFH(N) graph is at most $\left\lceil \frac{n}{2} \right\rceil + 1$ for $N = (c_n, ..., c_1, c_0)_2$.

Proof: Algorithm 3 can route messages from node x to node y in at most $\left\lceil \frac{j}{2} \right\rceil + 1$ steps. Hence the proof. #

5. EMBEDDING GRAPHS INTO IEFH GRAPHS

Graph embedding has been used to model the problem of simulating a parallel algorithm in a parallel machine. It is a mapping M of a guest graph G onto a host graph H. The cost of an embedding is measured in terms of *dilation*, *congestion*, and *expansion* [10]. The dilation of an embedding is the maximum distance of all edges of G in H. The congestion of an embedding is the maximum number of edges of G that share an edge of H. The expansion of an embedding is the ratio of the size of H to the size of G. Intuitively, dilation measures communication performance, congestion measures queuing delay, and expansion measures processor utilization. If G can be embedded into H with dilation 1 and expansion 1, then we say the embedding is optimal [11].

A. Cycles

In this subsection, we will try to embed cycles into the IEFH graph. In our previous work, we obtained the following results for IEH graphs [2].

Result 1. IEH graphs are *Hamiltonian* if the sizes of these graphs are not $2^n - 1$ for all $n \geq 2$.
Result 2. For an IEH(N) graph, an arbitrary cycle of even length N_e, where $3 < N_e < N$, is found.
Result 3. We find an arbitrary cycle of odd length N_o in an IEH(N) graph, where $2 < N_o < N$, if and only if a node of this graph has at least one forward 2-IC edge.

Obviously, an IEH graph is a spanning subgraph of an IEFH graph by Algorithm 2. Thus, the above results for embedding cycles into IEH graphs can also be applied to IEFH graphs. Moreover, in the next theorem, we will show that the IEFH graph is Hamiltonian.

Theorem 5. The IEFH graph is Hamiltonian.

Proof: Consider an IEFH($2^n - 1$) graph for $n \geq 2$. In this graph, node FH_0 exists by Step 3 of Algorithm 2. Further, FH_0 has two neighbor nodes (11...100) and (11...101). From Result 1, we know that an IEFH($2^n - 2$) graph is Hamiltonian. Without loss of generality, let P(11...100,11...101) be a Hamiltonian path in the IEFH($2^n - 2$) graph. By adding two edges (11...10, 11...100) and (11...10,11...101), the IEFH($2^n - 1$) graph becomes Hamiltonian. Since an IEH graph, which is Hamiltonian if its size is not $2^n - 1$, is a spanning subgraph of an IEFH graph and an IEFH($2^n - 1$) graph is Hamiltonian, the IEFH graph is Hamiltonian. #

B. Complete Binary Trees

In this subsection, we will embed a complete tree into an IEFH graph. Let a CBT_h denote a complete binary tree of height h. Our previous work showed that the minimum size of IEH graphs that contains a CBT_h is $2^h + 1$ [3]. However, we can optimally embed a CBT_h into an IEFH graph. The following definition and lemma are necessary.

Definition 1. [10] A *double-rooted binary tree* $DRBT_d$, where d is the height of the tree, is a complete binary tree with the root replaced by a path of length two.#

Lemma 3. [10] A double-rooted binary tree of height h can be embedded into an H_{h+1} with edge adjacency reserved. #

Theorem 6. A CBT_h can be embedded into an IEFH graph with dilation 1, congestion 1, and expansion 1.

Proof: First, we can embed a $DRBT_h$ into an H_{h+1} with edge adjacency reserved by Lemma 3. Since an FH_{h+1} contains an H_{h+1} as a spanning subgraph and is symmetric, we can embed a $DRBT_h$ into an FH_{h+1} with edge adjacency reserved. Then, we can let the roots of this DRBT be 11...10 and 11...1. Further, the son of 11...1 can be assigned as 11...101 since (11...10, 11...101) is an edge of an IEFH($2^{h+1} - 1$) graph by Algorithm 2. Hence the proof. #

6. CONCLUSIONS

In this paper, we have designed the IEFH graph as a new class of interconnection networks for an arbitrary number of nodes. We have shown this system is optimal fault tolerant and almost regular. We have also devised a simple routing algorithm for it. Thus, the diameter of this topology is only half of that of the incomplete hypercube (IH), the supercube, or the IEH graph. Finally, we have embedded cycles and complete binary trees into this graph with dilation 1, expansion 1, and congestion 1.

It is worth noting that we can construct new topologies by replacing folded hypercubes with crossed cubes or HCN graphs in IEFH graphs. These graphs also have low diameters. Further studies are being conducted to derive their properties and design mappings of parallel algorithms on them.

REFERENCES

1. A. E. Amawy and S. Latifi, "Properties and performance of folded hypercubes," *IEEE Transactions on Parallel and Distributed Systems*, Vol. 2, No. 1, 1991, pp. 31-42.
2. H. Y. Chang and R. J. Chen, "Embedding cycles in IEH graphs," *Information Processing Letters*, Vol. 64, 1997, pp. 23-27.
3. H. Y. Chang and R. J. Chen, "Graph embedding aspect of IEH graphs," *Journal of Information Science and Engineering*, 1998, to appear.
4. K. Efe, "The crossed cube architecture for parallel computation," *IEEE Transactions on Parallel and Distributed Systems*, Vol. 3, No. 5, 1992, pp. 513-524.
5. K. Ghose and K.R. Desai, "Hierarchical cubic network," *IEEE Transactions on Parallel and Distributed Systems*, Vol. 6, No. 4, 1995, pp. 427-435.
6. H. P. Katseff, "Incomplete hypercubes," *IEEE Transactions on Computers*, Vol. 37, No. 5, 1988, pp. 604-607.
7. Y. Saad and M. H. Schultz, "Topological properties of hypercubes," *IEEE Transactions on Computers*, Vol. 37, No. 7, 1988, pp. 867-872.
8. A. Sen, "Supercube: an optimal fault tolerant network architecture," *ACTA Inform*, Vol. 26, 1989, pp. 741-748.
9. S. Sur and P. K. Srimani, "IEH graphs: a novel generalization of hypercube graphs," *ACTA Inform*, Vol. 32, 1995, pp. 597-609.
10. T. Leighton, *Introduction to Parallel Algorithms and Architectures: Arrays, Trees, Hypercubes*, Morgan Kaufmann, reading, MA, 1992.
11. Y. M. Yeh and Y. C. Shyu, "Efficient distributed schemes for embedding binary trees into imcomplete hypercubes," in *Proceedings of 1994 IEEE Region Ninth Annual International Conference Theme: Frontiers of Computer Technology*, Vol. 1, 1994, pp. 182-186.

Hung-Yi Chang (張弘毅) was born in Taiwan in 1970. He received his B.S. and M.S. degrees in Computer Science and Information Engineering from National Chiao-Tung University, Hsinchu, Taiwan, in 1992 and 1994, respectively. He is currently a Ph.D. candidate in the Department of Computer Science and Information Engineering in National Chiao-Tung University, Hsinchu, Taiwan. His research interests include interconnection networks, parallel architecture, graph theory, and reliability.

Rong-Jaye Chen (陳榮傑) 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 the 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, Hsinchu, Taiwan. Professor Chen is a member of IEEE. His research interests include algorithm design, theory of computation, DNA computing, interconnection networks, mobile computing, network optimization, and combinatorial optimization.