Shortest-Path Routing Algorithm and Topological Properties for Two-Level Hypernet Networks

Hui-Ling Huang and Gen-Huey Chen
Department of Computer Science and Information Engineering,
National Taiwan University, Taipei, Taiwan

Abstract

Although many networks have been proposed as the topology of a large-scale parallel and distributed system. Most of them are neither expansible nor of equal degree. The inexpansibility and inequality of node degrees will make their VLSI implementation more difficult and more expensive. The hypernet, which was proposed by Hwang and Ghosh, represents a family of recursively scalable networks that are both expansible and of equal degree. In addition to the two merits, the hypernet has proven efficient for communication and computation. But, unfortunately, most topological properties and the problem of shortest-path routing for the hypernet are still unsolved. In this paper, we are concerned with the hypernet of two levels, we obtain the following results: (1) a shortest-path routing algorithm, (2) the diameter, (3) the connectivity, and (4) embedding of a ring, a torus, and a hypercube.

1 Introduction

One crucial step on designing a large-scale multiprocessor system is to determine the topology of the interconnection network (network for short). In the recent decade, a number of networks have been proposed in the literature: for example, hypercubes [25], cube-connected cycles [23], star networks [1], arrangement graphs [11], rotator graphs [9], hypernets [22], WK-recursive networks [26], and some hypercube-related networks [3, 10, 13-16, 20]. The readers are referred to two special issues [2, 19] for extensive references. Among them, only the hypernets [22] and the WK-recursive networks [26] own the two merits of expansibility and equal degree. A network is *expansible* if no changes to node configuration and link connections are required when it is expanded, and of *equal degree* if its nodes have the same degree which is maintained a constant. A network with these two properties will gain the advantages of easy implementation and low cost when it is manufactured.

The hypernet, which was proposed by Hwang and Ghosh [22], represents a family of recursively scalable networks. Since the hypernet has a recursive structure, it can be constructed incrementally by methodically putting together a number of basic modules. Although cubelet, treelet, and buslet can all serve as the basic module, the discussion in [22] assumed the cubelet to be the basic module. Throughout this paper we make the same assumption as [22] about the basic module.

The structure of the hypernet is characterized by two parameters: dimension of the cubelet (d) and expansion level (l). The hypernet with parameters d and l is composed of d-dimensional cubelets that are organized in a hirarchy of l levels. As computed in [22], there are $N=2^{2^{l-1}(d-2)+l+1}$ nodes contained in such a hypernet. Notice that N increases as a doubly exponential function of l as $d>2$. In Table I, we show the values of N for $2 \le d \le 7$ and $2 \le l \le 7$. As can be seen, N grows drastically with l for $d>2$. The entries in the shaded area contain more than one million nodes. Under current hardware technology, it is more feasible to build a massively parallel system containing thousands or tens of thousands of nodes. Therefore, it is meaningful to concentrate our effort on the hypernet with small l.

Now we briefly review earlier work about the hypernet. In [22], the performance of the hypernet was analyzed in terms of average path delay, hardware requirements, and support for communication traffic in typical application domains. Moreover, the mapping of a wide range of algorithms on the hypernet was demonstrated. A simple routing algorihm based on the recursiveness of the hypernet was also suggested in [22]. This algorithm, although very easy to be understood, cannot guarantee the shortest routing path. An easy upper bound of $2^{l-1}(d+1)-1$ on the diameter was further derived from the routing algorithm and the recursive structure. A broadcasting algorithm for the hypernet can be found in [7].

Before few results were obtained about the topological properties such as diameter, hamiltonicity, and connectivity of the hypernet. The diameter of a network is usually adopted as a measure of the maximum transmission delay among the network, and threrefore it can influence the efficiencies of many applications such as broadcasting. A hamiltonian network has the capability of embedding a ring network of the same size with dilation 1. In other words, a hamiltonian network can simulate a ring network of the same size without loss of efficiency. The

1087-4089/96 $5.00 © 1996 IEEE

connectivity of a network is commonly used as a measure of the robustness of the network. By Menger's theorem [5], within a network the number of node-disjoint paths (except the two end nodes) between any two nodes is bounded above by the connectivity.

Besides, no shortest-path routing algorithm is available for the hypernet. In order for the hypernet to be considered a general-purpose parallel and distributed system, efficient communication must be addressed because communication delay has been recognized as a major cause of performance degradation in such a computing environment. Shortest-path problems are the most fundamental problems in the study of communication within a network. Therefore it is not surprising that a large number of researches (see [12]) have been devoted to the subject of shortest-path problems, and efficient shortest-path routing algorithms have been designed for many existing networks [1], [6], [9], [11], [13], [15], [17].

It is not easy to solve the problems above for the whole family of the hypernet networks because the structure of the hypernet is rather complex. For example, it is hard to imagine the structure of a hypernet even if its size is small e.g., d=3 and l=3. Also note that the structure of the hypernet is not symmetric. In this paper, we try to attack these problems starting with a special subclass, i.e., l=2, of the hypernet networks. Our result is helpful for interested readers to work on larger subclasses. Undoubtedly, the first step to investigate the topology of a network is to provide a mathematical (graph) description of the network. Hence, a concise mathematical definition of the hypernet is first introduced in the next section. Some basic properties of the hypernet are also proved. A shortest-path routing algorithm is then proposed in Section 3. The diameter, and connectivity are computed in Section 4. To compute the connectivity, maximum number of node-disjoint paths between any two nodes are constructed. The hamiltonicity is shown in Section 5, where the embedding of rings, tori, and hypercubes are also discussed. Finally, this paper is concluded in Section 6.

2 Hypernet networks

For convenience we use HN(d, l) to denote a hypernet of level l whose basic modules are each a d-dimensional cubelet (d-cubelet for short). A link of the hypernet is referred to as an *internal link* if it connects two nodes within the same basic module, and an *external link* otherwise. A d-cubelet is indeed a d-dimensional hypercube augmented with an external link at each node. So a d-cubelet contains 2^d nodes and 2^d external links. The basic module is denoted by HN(d, 1). HN(d, 2) can be constructed by grouping 2^{d-1} HN(d, 1)s as follows. Each HN(d, 1) assigns one of its external links as an I/O channel, and reserves half (i.e., 2^{d-1}) of them for future use. The remaining 2^{d-1}-1 external links are used to connect other 2^{d-1}-1 HN(d, 1)s. Hence, if each HN(d, 1) is regarded as a vertex, then HN(d, 2) forms a 2^{d-1}-vertex complete graph. HN(d, 3) can be constructed from HN(d, 2)s all in a similar way. In general, HN(d, l) for l≥2 can be constructed recursively by grouping $M_{d,l}=2^{(d-2)2^{l-2}+1}$ HN(d, l-1)s. There are $2M_{d,l}$ external links available for each HN(d, l-1); one is dedicated as an I/O channel, $M_{d,l}$ are reserved for future expansion, and $M_{d,l}-1$ are used to connect other HN(d, l-1)s. There are $N=2^{2^{l-1}(d-2)+l+1}$ nodes contained in HN(d, l) (N and $M_{d,l}$ were computed in [22]). Letting $n=\log_2 N$ and $m_{d,l}=\log_2 M_{d,l}$ (note that $2m_{d,l}+l-1=n$), HN(d, l) can be defined mathematically as follows.

Definition 2.1. The node set of HN(d, l) is denoted by $\{b_{n-1}b_{n-2}...b_0 \mid b_i=0 \text{ or } 1 \text{ for } 0 \le i \le n-1\}$. Node adjacency is defined as follows: $b_{n-1}b_{n-2}...b_0$ is adjacent to (1) $b_{n-1}...b_{d-1}...b_i'...b_0$, where b_i' is the complement of b_i and $0 \le i \le d-1$, and (2) $b_{n-1}...b_{j+1+2m_{d,j+2}}b_{j+m_{d,j+2}}...b_{j+1}b_{j+2m_{d,j+2}}...b_{j+1+m_{d,j+2}}b_j...b_0$ (i.e., swapping $b_{j+2m_{d,j+2}}...b_{j+1+m_{d,j+2}}$ with $b_{j+m_{d,j+2}}...b_{j+1}$) if $b_j=0$ and $b_{j-1}=b_{j-2}=...=b_0=1$, where $0 \le j \le l-2$.

Each node of HN(d, l) is assigned with an n-bit address $b_{n-1}b_{n-2}...b_0$. The leftmost $m_{d,l}$ bits, i.e., $b_{n-1}b_{n-2}...b_{n-m_{d,l}}$, identify the HN($d$, l-1) where the node resides. The next $n-m_{d,l}-d$ bits further identify the cubelet where the node resides. The remaining d bits distinguish the node from the others within the same cubelet. The links defined by (1) are internal links. The links defined by (2) are external links each connecting two HN(d, j+1)s (within the same HN(d, j+2)). These links are referred to as *level* j+1 links. The external links incident on nodes with $b_{j+2m_{d,j+2}}...b_{j+1+m_{d,j+2}}=b_{j+m_{d,j+2}}...b_{j+1}$ are assigned as I/O channels. The other external links are reserved for future expansion.

Figure 1 shows the structure of HN(3, 3). Let us consider the node 11010101 that is located in the HN(3, 2) with identifier 110 and in the cubelet with identifier 10. Since b_1=0 and b_0=1, there is an external link stemming from it and reaching the node 10111001. Since this link connects two HN(3, 2)'s within the same HN(3, 3), it is a level two link. Another example is the link between 11010000 and 11000100. This link is established because b_0=0 and both end nodes can be obtained from each other by swapping b_4b_3 with b_2b_1. Since it connectes two HN(3, 1)'s within the same HN(3, 2), it is a level one link. Also note that the external link incident on 11010100 is used as an I/O channel because $b_4b_3=b_2b_1$.

For easy of the following discussion, we represent each node $b_{n-1}b_{n-2}...b_0$ by a three-tuple (I, J, K), where $I=b_{n-1}...b_{n-m_{d,l}}$, $J=b_{n-m_{d,l}-1}...b_{l-1}$, and $K=b_{l-2}...b_0$. Note that node (I, J, K) is connected to node (J, I, K) if $K=01^{l-2}$, where 1^{l-2} represents l-2 consecutive 1s. In the rest of this section, we introduce three lemmas that are helpful to shortest-path routing.

Lemma 2.1. [21] Suppose P is an arbitrary path from the source node to the destination node in HN(d, l), where $l\geq 2$. If P contains three or more external links of level l-1, then P is not the shortest.

Lemma 2.2. [21] Suppose (I_s, J_s, K_s) and (I_d, J_d, K_d) are two nodes in HN(d, l) and they reside in the same HN(d, i) for some $1\leq i\leq l$-1, where $l\geq 2$. Then their shortest path does not contain external link of level equal to or greater than i.

Lemma 2.3. [21] Suppose (I_s, J_s, K_s) and (I_d, J_d, K_d) are two nodes in HN(d, l) and they reside in different HN(d, l-1)s (i.e., $I_s\neq I_d$), where $l\geq 2$. If their shortest path contains exactly one external link of level l-1, then it has the form: $(I_s, J_s, K_s) \rightarrow\rightarrow_* (I_s, I_d, 01^{l-2}) \rightarrow_{l-1} (I_d, I_s, 01^{l-2}) \rightarrow\rightarrow_* (I_d, J_d, K_d)$, where $\rightarrow\rightarrow_*$ indicates a shortest path.

Suppose (I, J_s, K_s) and (I, J_d, K_d) are two nodes in HN(d, l). Since they are located in the same HN(d, l-1) (with identifier I), we can use J_sK_s and J_dK_d to represent their addresses in the HN(d, l-1). Moreover, we use $d(J_sK_s, J_dK_d)$ to denote their distance. The *distance* between two nodes of a network is defined as the length of their shortest path. We have the following lemma.

Lemma 2.4. [21] Suppose (I_s, J_s, K_s) and (I_d, J_d, K_d) are two nodes in HN(d, l) and they reside in different HN(d, l-1)s, where $l\geq 2$. If their shortest path contains two external links of level l-1, then it has the form: $(I_s, J_s, K_s) \rightarrow\rightarrow_* (I_s, J^*, 01^{l-2}) \rightarrow_{l-1} (J^*, I_s, 01^{l-2}) \rightarrow\rightarrow_* (J^*, I_d, 01^{l-2}) \rightarrow_{l-1} (I_d, J^*, 01^{l-2}) \rightarrow\rightarrow_* (I_d, J_d, K_d)$, where J^* minimizes the value of $d(J_sK_s, J^*01^{l-2})+d(J^*01^{l-2}, J_dK_d)$.

3 Shortest-path routing algorithm

Given arbitrary two nodes in a network, the *routing problem* is to determine a path such that messages can be transmitted from one to the other. In this section, an algorithm is proposed to determine a shortest path between arbitrary two nodes of HN(d, 2). Extension to higher-level hypernet networks is also discussed.

Suppose (I_1, J_1, K_1) and (I_2, J_2, K_2) are two nodes in HN(d, 2), where I_1, J_1, I_2, and J_2 have length d-1, and K_1 and K_2 have length 1. The combination of J_1 and K_1 (J_2 and K_2) uniquely identifies a node in the HN(d, 1) with identifier I_1 (I_2). If (I_1, J_1, K_1) and (I_2, J_2, K_2) are in the same HN(d, 1) (i.e., $I_1=I_2$), then their distance can be computed as the number of 1s in $J_1K_1\oplus J_2K_2$, where $\oplus$ represents the exclusive-OR operation. Besides, their shortest path can be determined easily. Hence, in the rest of this section, we concentrate our effort on the case of $I_1\neq I_2$. Letting $B(J_1K_1\oplus J_2K_2)$ represent the number of 1s in $J_1K_1\oplus J_2K_2$, we have the following lemma by the aid of Lemma 2.3.

Lemma 3.1. Suppose (I_s, J_s, K_s) and (I_d, J_d, K_d) are two nodes in HN(d, 2), where $I_s\neq I_d$. Then, their shortest path that contains exactly one external link of level one has length equal to $B(J_sK_s\oplus I_d0)+B(I_s0\oplus J_dK_d)+1=B(J_s\oplus I_d)+B(I_s\oplus J_d)+B(K_s\oplus 0)+B(0\oplus K_d)+1$.

Similarly, by the aid of Lemma 2.4, the length of the shortest path that contains two external links of level one can be computed as follows.

Lemma 3.2. [21] Suppose (I_s, J_s, K_s) and (I_d, J_d, K_d) are two nodes in HN(d, 2), where $I_s\neq I_d$. Then, their shortest path that contains two external links of level one has length equal to $B(I_s\oplus I_d)+B(J_s\oplus J_d)+B(K_s\oplus 0)+B(0\oplus K_d)+2$.

Combining Lemmas 2.1, 3.1, and 3.2, the length of the shortest path from (I_s, J_s, K_s) to (I_d, J_d, K_d) can be determined as follows.

Theorem 3.1. Suppose (I_s, J_s, K_s) and (I_d, J_d, K_d) are two nodes in HN(d, 2), where $I_s\neq I_d$. Then, their shortest path has length equal to $\min\{B(J_s\oplus I_d)+B(I_s\oplus J_d)+B(K_s\oplus 0)+B(0\oplus K_d)+1, B(I_s\oplus I_d)+B(J_s\oplus J_d)+B(K_s\oplus 0)+B(0\oplus K_d)+2\}$.

Since determining the shortest path between two nodes of HN(d, 1) is very easy, the shortest path between (I_s, J_s, K_s) and (I_d, J_d, K_d) can be constructed according to either Lemma 2.3 or Lemma 2.4.

Before ending this section, we briefly discuss how to obtain the shortest path in HN(d, l) for l>2. First, we discuss the case of l=3. Suppose (I_s, J_s, K_s) and (I_d, J_d, K_d) are two nodes in HN(d, 3). If $I_s=I_d$, they are located in the same HN(d, 2), and the discussion is the same as the case of l=2. So, we assume $I_s\neq I_d$. Lemma 2.1 assures that their shortest path contains one or two external links of level 2. If the shortest path contains one external link of level 2, its length can be obtained by the aid of Lemma 2.3, because the lengths of the subpaths $(I_s, J_s, K_s) \rightarrow\rightarrow_* (I_s, I_d, 01^{l-2})$ and $(I_d, I_s, 01^{l-2}) \rightarrow\rightarrow_* (I_d, J_d, K_d)$ can be computed as stated in Theorem 3.1. Otherwise, computing the length of the shortest path is reduced to determining the J^* in Lemma 2.4. Unfortunately, no efficient method for computing J^* is available as yet except a brute-force method.

For the case of l>3, the level i, where $1\leq i\leq l$, is first determined such that (I_s, J_s, K_s) and (I_d, J_d, K_d) are in the same HN(d, i), but in different HN(d, i-1)s. Lemma 2.2 assures that their shortest path will not travel outside the HN(d, i) where they are resident. Again, by Lemma 2.1, the shortest path contains one or two external links of level i-1. For the former case, the shortest path can be determined, provided the shortest-path routing problem is solvable for HN(d, i-1). For the latter case, the J^* in

Lemma 2.4 should be determined before the shortest path can be computed.

To sum up, the main difficulty of the shortest-path routing problem arises in computing the J^*.

4 Diameter and connectivity

In this section, the diameter and the connectivity of HN(d, 2) are computed.

A. Diameter

The *diameter* of a network is defined as the maximum distance between any two of its nodes. By $D_{d,l}$ we denote the diameter of HN(d, l). Hwang and Ghosh [22] have proposed an upper bound on $D_{d,l}$ as stated below.

Lemma 4.1. [22] $D_{d,l} \leq 2^{l-1}(d+1)-1$.

Letting l=2, we obtain $D_{d,2} \leq 2d+1$ from Lemma 4.1. A lower bound can be established by computing the distance of two nodes. Let us consider node $0^{n-1}1$ and node $1^{n-1}1$ in HN(d, 2), where $n=(d-1)+(d-1)+1=2d-1$ is the length of the node identifier. By Theorem 3.1, $d(0^{n-1}1, 1^{n-1}1)$ is computed as $\min\{2d+1, 2d+2\}=2d+1$, which is identical to Hwang and Ghosh's bound. Hence, the following theorem holds.

Theorem 4.1. $D_{d,2}=2d+1$.

B. Connectivity

An important issue when we design a parallel and distributed system is how fault-tolerant it is. A commonly used measure for the fault-tolerance of a network is the minimum number of nodes whose removal can result in disconnecting the network. This measure has been referred to as *connectivity*. Thus, a network with connectivity κ is guaranteed to remain connected even if κ-1 nodes are removed.

Obviously the connectivity of a network is bounded above by the network degree which is defined as the minimum of its node degrees. If the connectivity is equal to the network degree, then the network is said to be *maximally fault-tolerant*. By Menger's theorem [5] a lower bound on the connectivity can be obtained by counting the maximum number of node-disjoint paths between any two nodes. It is important for a network to have node-disjoint paths between any two of its nodes, in order to speed up the transfer of a large amount of data and provide alternative routes in case of node failures.

There are d node-disjoint paths between arbitrary two nodes of HN(d, 2), and they can be constructed as follows. Suppose (I_A, J_A, K_A) and (I_B, J_B, K_B) are two distinct nodes of HN(d, 2). If they belong to the same HN(d, 1), then the d node-disjoint paths can be constructed according to Saad and Schultz's work [25]. Otherwise, the d node-disjoint paths are constructed as $(I_A, J_A, K_A) \rightarrow \ldots \rightarrow (I_A, X_i, 0) \rightarrow (X_i, I_A, 0) \rightarrow \ldots \rightarrow (X_i, I_B, 0) \rightarrow (I_B, X_i, 0) \rightarrow \ldots \rightarrow (I_B, J_B, K_B)$, where i=1, 2, ..., d, and X_is are all distinct. Since the node-disjoint property of the set of subpaths $(I_A, J_A, K_A) \rightarrow \ldots \rightarrow (I_A, X_i, 0)$ and the set of subpaths $(I_B, X_i, 0) \rightarrow \ldots \rightarrow (I_B, J_B, K_B)$ can be guaranteed by Rabin's work [24], these d paths are node-disjoint. Since HN(d, 2) has degree d, the following theorem is thus concluded.

Theorem 4.2. HN(d, 2) has connectivity d, and is maximally fault-tolerant.

5 Embedding

In this section the capability of HN(d, 2) to embed rings, tori, and hypercubes are shown. An *embedding* of a network (*guest network*) onto another network (*host network*) is a mapping ϕ from the node set of the guest network to the node set of the host network. Thus, a link in the guest network may correspond to a path in the host network. Four measures, i.e., dilation, congestion, expansion, and load, are commonly used to evaluate a mapping. The *dilation* of ϕ is defined as the maximum distance between $\phi(u)$ and $\phi(v)$ for all links (u, v) in the guest network. The *congestion* of ϕ is defined as the maximum number of links in the guest network whose corresponding paths in the host network contain the same link in the host network. The *expansion* of ϕ is defined as the ratio of the number of nodes in the host network to the number of nodes in the guest network. The *load* of ϕ is defined as the maximum number of nodes in the guest network that are mapped to the same node in the host network. If all four measures are constant, the host network can efficiently simulate the guest network with constant slowdown.

A. Embedding of rings

A cycle in a network is called a *hamiltonian cycle* if it contains every node of the network exactly once [4]. Similarly, a *hamiltonian path* is a path that contains every node of the network exactly once. A network is *hamiltonian* if it contains a hamiltonian cycle. A direct consequence of the hamiltonicity of a network is its capability of embedding a ring. If a network is hamiltonian, then it can embed a ring with all four measures equal to one.

It is not difficult to construct a hamiltonian cycle for the hypercube. For example, the three-dimensional hypercube contains a hamiltonian cycle as follows: (000, 001, 011, 010, 110, 111, 101, 100), which is really a Gray code of three bits. A *Gray code* [8] of n bits contains 2^n distinct codewords $(G(0), G(1), G(2), \ldots, G(2^n-1))$, where each $G(i)$, $0 \leq i \leq 2^n-1$, is an n-bit sequence, such that $G(i)$ and $G((i \pm 1) \text{ MOD } 2^n)$ differ in exactly one bit position. For illustration, (000, 001, 101, 100, 110, 111, 011, 010) is another Gray code of three bits.

Gray codes can be constructed conveniently by a recursive method. Suppose $(G(0), G(1), G(2), ..., G(2^n-1))$ is a Gray code of n bits. Then, $(0G(0), 0G(1), 0G(2), ..., 0G(2^n-1), 1G(2^n-1), ..., 1G(2), 1G(1), 1G(0))$ forms a Gray code of $n+1$ bits. Also note that Gray codes thus constructed have a property that the first codeword and the $(2^{i-1}+1)$th codeword differ from the 2^ith codeword in exactly one bit position. In the last instance, $0G(0)$, $1G(2^n-1)$, and $1G(0)$ are the first, the (2^n+1)th, and the 2^{n+1}th codewords, respectively, and $0G(0)$ and $1G(2^n-1)$ differ from $1G(0)$ in exactly one bit position. A more detailed description about Gray codes can be found in [8].

Since the hypercube is hamiltonian, there is a hamiltonian path between its two adjacent nodes. So, in HN(d, 2) there is a hamiltonian path between node $(I, G(i), 0)$ and node $(I, G(i\pm1) \text{ MOD } 2^{d-1}, 0)$, where $0\leq i\leq 2^{d-1}-1$, and it is conveniently expressed as $(I, G(i), 0) \rightarrow_H (I, G(i\pm1) \text{ MOD } 2^{d-1}, 0)$. By the aid of Gray codes, a hamiltonian cycle of HN(3, 2) can be constructed as follows. Letting $(G(0), G(1), G(2), G(3))$ denote a Gray code of two bits, a hamiltonian cycle of HN(3, 2) is expressed as $(G(0), G(2), 0) \rightarrow_H (G(0), G(3), 0) \rightarrow (G(3), G(0), 0) \rightarrow_H (G(3), G(1), 0) \rightarrow (G(1), G(3), 0) \rightarrow_H (G(1), G(2), 0) \rightarrow (G(2), G(1), 0) \rightarrow_H (G(2), G(0), 0) \rightarrow (G(0), G(2), 0)$. In this construction, the nodes are traveled in the sequence of $(G(0), *, *)$, $(G(3), *, *)$, $(G(1), *, *)$, and $(G(2), *, *)$, where $(G(i), *, *)$, $0\leq i\leq 3$, represents the set of nodes contained in the HN(3, 1) with identifier $G(i)$.

The construction above can be extended to HN(d, 2) for $d>3$. Since HN(d, 2) is hamiltonian, the following result is concluded.

Theorem 5.1. [21] HN(d, 2) can embed a ring with dilation 1, congestion 1, expansion 1, and load 1.

B. Embedding of tori

Next, we consider an embedding of a torus T of size $2^{d-1}\times2^d$ onto HN(d, 2). A torus is simply a mesh with wrap-around links in the rows and the columns. The node set of T can be conveniently expressed by $\{T(i, j) \mid i=0, 1, ..., 2^{d-1}-1 \text{ and } j=0, 1, ..., 2^d-1\}$. Suppose $G_1(0), G_1(1), ..., G_1(2^{d-1}-1)$ constitute a Gray code of d-1 bits, and $G_2(0), G_2(1), ..., G_2(2^d-1)$ constitute a Gray code of d bits. Moreover, let $X(G_2(j))$ represent the subsequence of $G_2(j)$ that contains the leftmost d-1 bits and $Y(G_2(j))$ represent the rightmost bit of $G_2(j)$. For example, if $G_2(j)$=110011, then $X(G_2(j))$=11001 and $Y(G_2(j))$=1. A bijective mapping ϕ from the node set of T to the node set of HN(d, 2) is defined as follows: $\phi(T(i, j))=(G_1(i), X(G_2(j)), Y(G_2(j)))$. Clearly ϕ has load 1 and expansion 1.

Theorem 5.2. [21] HN(d, 2) can embed a torus of size $2^{d-1}\times2^d$ with dilation 5, average dilation $\frac{5}{2}-\frac{1}{2^{d-1}}$, congestion 8, average congestion $\frac{10\times2^d-8}{d\times2^d+2^{d-1}-1}$, load 1, and expansion 1.

The average dilation is bounded above by 2.5, and the average congestion is decreasing as d grows. For nontrivial HN(d, 2)s, i.e., $d\geq2$, the average congestion is bounded by 32/9. The following corollary is an immediate consequence of Theorems 5.1 and 5.2.

Corollary 5.1. Suppose A is an algorithm executable on a ring of length 2^{2d-1} or a torus of size $2^{d-1}\times2^d$. Then, A can be executed on HN(d, 2) as well with constant slowdown.

C. Embedding of hypercubes

We consider an embedding of a hypercube H of dimension $2d$-1 onto HN(d, 2). The node set of H is conveniently expressed by $\{H(K) \mid K \in \{0, 1\}^{2d-1}\}$. Let $K=I\|J$, where I is a (d-1)-bit sequence, J is a d-bit sequence, and $\|$ represents the concatenation of I and J. A bijective mapping ϕ from the node set of H to the node set of HN(d, 2) is defined as follows: ϕ: $H(I\|J) \rightarrow (I, X(J), Y(J))$, where $X(J)$ represents the subsequence of J that contains the leftmost d-1 bits and $Y(J)$ represents the rightmost bit of J. Clearly, the load and the expansion of ϕ are all 1.

Theorem 5.3. [21] HN(d, 2) can embed a hypercube of dimension $2d$-1 with dilation 5, average dilation $\frac{(5d-4)\times2^{2d-2}-(d-1)\times2^d}{(2d-1)\times2^{2d-2}}$ $(<\frac{5}{2})$, load 1, and expansion 1. Although the congestion is not constant, the average congestion is smaller than 5.

Now that the congestion is not constant, HN(d, 2) cannot simulate all algorithms of H with constant slowdown. However, if we forcus our attention on the data parallel algorithms [18], efficient simulation is still possible. The *data parallel algorithms* when they are executed on the hypercube have their operands transmitted over the links of the same dimension at every computation step. Most existing algorithms designed for the hypercube fall into this category. To simulate the data parallel algorithms, we only need to simulate the data transmission along each dimension. Assume data transmission is requested for the ith dimension from the right, where $1\leq i\leq 2d$-1. If $1\leq i\leq d$, the links used for data transmission are of type $(H(I\|J), H(I\|J'))$. According to the link-to-path correspondence, the data transmission can be simulated by the links $(I, X(J), Y(J)) \rightarrow (I, X(J'), Y(J'))$ of HN(d, 2). Note that different links of type $(H(I\|J), H(I\|J'))$ are simulated by different links of HN(d, 2).

On the other hand, if $d+1\leq i\leq 2d$-1, the links used for data transmission are of type $(H(I\|J), H(I'\|J))$. According to the link-to-path correspondence, the data transmission can be simulated by paths of HN(d, 2) whose lengths range from 2 to 5. Moreover, each link in these paths is responsible for simulating at most two links of H. For example, the link $(X(J), I, 0) \rightarrow (X(J), I', 0)$ is responsible for simulating two links, $(H(I||X(J)0), H(I'||X(J)0))$ and $(H(I||X(J)1), H(I'||X(J)1))$, of H, whereas the link $(X(J), I, 1) \rightarrow (X(J), I, 0)$ is responsible for simulating only one link, $(H(I||X(J)1), H(I'||X(J)1))$, of H. According to the discussion above, we have the following theorem.

Theorem 5.4. Suppose A is a data parallel algorithm executable on a hypercube of dimension $2d$-1. Then, A can be executed on HN(d, 2) as well with constant slowdown.

6 Conclusion

The hypernet, which integrates positive features of both hypercubes and tree-based topologies, is suitable to be the topology of a massively parallel computer system, especially for distributed supercomputing and AI applications. In addition to the two structural advantages of easy expansion and equal degree, the hypernet has proven efficient for communication and computation. However, there is a drawback for the hypernet: the number of nodes increases very rapidly. For example, the hypernet contains more than one million nodes if l=3 and d>5 or l=4 and d>3 or l>4 and d>2. Therefore, taking a practical consideration, we emphasize the hypernet of feasible size in this paper.

A mathematical definition of a network is very necessary, in order to clarify its structure. In [22], Hwang and Ghosh have described the structure of the hypernet, to a certain degree, informally. In this paper, we first introduce a concise mathematical definition for the hypernet. This definition is very crucial for us to work out subsequent results.

Although a heuristic routing algorithm for the hypernet has been proposed by Hwang and Ghosh [22], it is not a shortest-path one. In this paper, we designed a shortest-path routing algorithm for the hypernet of two levels. The design method can be generalized to the hypernet of higher levels.

Many important topological properties of the hypernet were not investigated before. Hwang and Ghosh [22] have suggested an upper bound of the diameter. In this paper, we computed the exact value of the diameter. The connectivity was obtained by constructing maximum number of node-disjoint paths between any two nodes. Moreover, we showed the embedding of rings, tori, and hypercubes into the hypernet.

When d=2, the number of nodes contained in the hypernet grows more slowly with respect to l (see Table I). So, it is the next subclass of the hypernet networks which we will pay more attention to in the near future.

References

[1] S. B. Akers, D. Harel, and B. Krishnamurthy, "The star graph: an attractive alternative to the n-cube," *Proceedings of the International Conference on Parallel Processing*, 1987, pp. 393-400.

[2] J. C. Bermond, Ed., *Interconnection Networks*, a special issue of *Discrete Applied Mathematics*, vol. 37+38, 1992.

[3] L. N. Bhuyan and D. P. Agrawal, "Generalized hypercube and hyperbus structures for a computer network," *IEEE Transactions on Computers*, vol. 33, no. 4, pp. 323-333, 1984.

[4] J. Bondy and U. Murthy, *Graph Theory with Applications*, Macmillan Press, 1976.

[5] F. Buckley and F. Harary, *Distance in Graphs*, Addison-Wesley, 1990.

[6] G. H. Chen and D. R. Duh, "Topological properties, communication, and computation on WK-recursive networks," *Networks*, vol. 24, no. 6, pp. 303-317, 1994.

[7] G. H. Chen, S. C. Hwang, and D. R. Duh, "A general broadcasting scheme for recursive networks with complete connection," submitted.

[8] M. S. Chen and K. G. Shin, "Processor allocation in an N-cube multiprocessor using Gray codes," *IEEE Transactions on Computers*, vol. C-36, no. 12, pp. 1396-1407, 1987.

[9] P. Corbett, "Rotator graphs: an efficient topology for point-to-point multiprocessor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 3, no. 5, pp. 622-626, 1992.

[10] S. P. Dandamudi and D. L. Eager, "Hierarchical interconnection networks for multicomputer systems," *IEEE Transactions on Computers*, vol. 39, no. 6, pp. 786-797, 1990.

[11] K. Day and A. Tripathi, "Arrangement graphs: a class of generalized star graphs," *Information Processing Letters*, vol. 42, no. 5, pp. 235-241, 1992.

[12] N. Deo and C. Pang, "Shortest-path algorithms: taxonomy and annotation," *Networks*, vol. 14, pp. 275-323, 1984.

[13] K. Efe, "The crossed cube architecture for parallel computation," *IEEE Transactions on Parallel and Distributed Systems*, vol. 3, no. 5, pp. 513-524, 1992.

[14] A. El-Amawy and S. Latifi, "Bridged hypercube networks," *Journal of Parallel and Distributed Computing*, vol. 10, no. 1, pp. 90-95, 1990.

[15] A. El-Amawy and S. Latifi, "Properties and performance of folded hypercubes," *IEEE Transactions on Parallel and Distributed Systems*, vol. 2, no. 1, pp. 31-42, 1991.

[16] A. H. Esfahanian, L. M. Ni, and B. E. Sagan, "The twisted N-cube with application to multiprocessing," *IEEE Transactions on Computers*, vol. 40, no. 1, pp. 88-93, 1991.

[17] V. Faber, J. W. Moore, and W. Y. C. Chen, "Cycle prefix digraphs for symmetric interconnection networks," *Networks*, vol. 23, pp. 641-649, 1993.

[18] K. Ghose and K. R. Desai, "Hierarchical cubic networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 6, no. 4, pp. 427-435, 1995.
[19] D. F. Hsu, *Interconnection Networks and Algorithms*, Special Issue in *Networks*, vol. 23, no. 4, 1993.
[20] W. J. Hsu, "Fibonacci cubes - a new interconnection topology," *IEEE Transactions on Parallel and Distributed Systems*, vol. 4, no. 1, pp. 3-12, 1993.
[21] H. L. Huang and G. H. Chen, "Shortest-Path Routing Algorithm and Topological Properties for Two-Level Hypernet Networks", *Technical Report NTUCSIE 95-08*.
[22] K. Hwang and J. Ghosh, "Hypernet: A Communication-Efficient Architecture for Constructing Massively Parallel Computers," *IEEE Transaction on Computers*, vol. C-36, no. 12, pp. 1450-1466, 1987.
[23] F. P. Preparata and J. Vuillemin, "The cube-connected cycles: a versatile network for parallel computation," *Communications of the ACM*, vol. 24, no. 5, pp. 300-309, 1981.
[24] M. O. Rabin, "Efficient dispersal of information for security, load balancing, and fault tolerance," *Journal of the ACM*, vol. 36, no. 2, pp. 335-348, 1989.
[25] Y. Saad and M. H. Schultz, "Topological properties of hypercubes," *IEEE Transactions on Computers*, vol. C-37, no. 7, pp. 867-872, 1988.
[26] G. D. Vecchia and C. Sanges, "A recursively scalable network VLSI implementation," *Future Generation Computer Systems*, vol. 4, no. 3, 1988, pp. 235-243.

Table I. The number of processors contained in the hypernet

l \ d	2	3	4	5	6	7
2	2^3	2^5	2^7	2^9	2^{11}	2^{13}
3	2^4	2^8	2^{12}	2^{16}	2^{20}	2^{24}
4	2^5	2^{13}	2^{21}	2^{29}	2^{37}	2^{45}
5	2^6	2^{22}	2^{38}	2^{54}	2^{70}	2^{86}
6	2^7	2^{39}	2^{71}	2^{103}	2^{135}	2^{167}
7	2^8	2^{72}	2^{136}	2^{200}	2^{264}	2^{328}

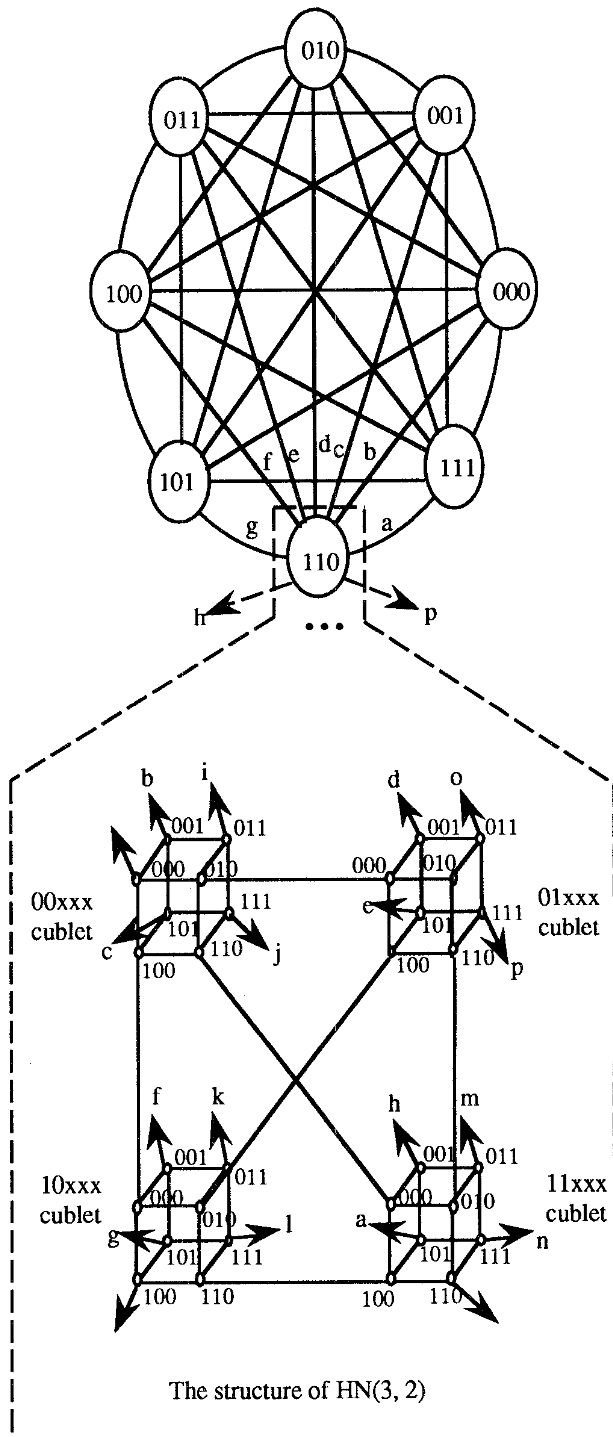

Figure 1. The structure of HN(3, 3).