Hierarchical Swapped Networks: Efficient Low-Degree Alternatives to Hypercubes and Generalized Hypercubes

Chi-Hsiang Yeh and Behrooz Parhami
Department of Electrical and Computer Engineering
University of California
Santa Barbara, CA 93106, USA

Abstract

In this paper, we propose a new class of interconnection networks called hierarchical swapped networks (HSNs). We show that some subclasses of HSNs can efficiently emulate hypercubes, or generalized hypercubes, while having node degrees significantly smaller than the emulated networks. In particular, a suitably constructed HSN can emulate a hypercube or generalized hypercube with constant slowdown under the single-dimension communication model and asymptotically optimal slowdown with respect to its node degree under the all-port communication model. As a consequence, we obtain a variety of efficient algorithms on HSNs through emulation, thus proving the versatility of HSN. Some subclasses of HSNs are also shown to have asymptotically optimal diameters with respect to their node degrees. HSNs appear to be attractive low-degree alternatives to hypercubes and generalized hypercubes for general-purpose parallel computers.

1 Introduction

Many interconnection schemes for parallel architectures have been proposed in recent years [3, 5, 6, 7, 8, 13, 14, 15]. Among them, the hierarchical cubic network (HCN) [6], hierarchical folded-hypercube network (HFN) [5], and three-level hierarchical cubic network (3-HCN) [15] offer various desirable properties. HCNs (HFNs) use (folded) hypercube networks as basic modules, are composed of nodes with degree $n/2+1$ $(n/2+2)$, as opposed to n for a hypercube of the same size, and can emulate a hypercube with single-port communication in $O(1)$ time. 3-HCNs use hypercube networks as basic modules, are composed of nodes with degree $n/3+2$, and also emulate a hypercube in $O(1)$ time. These networks have diameters smaller than that of a hypercube of the same size. Hierarchical swapped networks (HSNs), a subclass of swapped networks [16], not only generalize, and serve to unify, these parallel architectures as well as their algorithms, but also generate a much wider class of cost-effective high-performance interconnection networks.

In this paper, we show that HSNs can emulate hypercubes, generalized hypercubes, or high-dimensional meshes efficiently. As a consequence, we obtain a variety of algorithms on HSNs through emulation. We also develop efficient and elegant algorithms for packet routing and ascend/descend algorithms. An HSN has node degree considerably smaller than that of hypercube when the network size grows very large, and can achieve optimal diameter with respect to its node degree. Thus, HSNs have potential for use in high-performance general-purpose parallel architectures. By comparing the properties and performance of HSNs with other networks, we conclude that HSNs have many advantages and appear to be an attractive alternative to the hypercube, generalized hypercube, as well as other high node-degree networks.

In Section 2, we define HSNs, derive some of their parameters, and establish HCNs and HFNs as subclasses of HSNs. In Section 3, we present HSNs based on an n-cube. We present ascend/descend algorithms on hypercube-based HSNs. We show how to emulate a hypercube efficiently under different communication models. In Section 4, we present HSNs based on a complete graph. We show that these networks can emulate generalized hypercubes efficiently. We also show that N-node HSNs based on M-node complete-graph nuclei always achieve optimal diameter for $M = \Omega(\log N/\log\log N)$. In Section 5, we construct HSNs based on other nucleus graphs. We conclude that HSNs are cost-effective, have desirable topological and algorithmic properties, and appear to be suitable abstractions for implementing versatile high-performance interconnection networks with reasonable cost.

2 Hierarchical Swapped Networks

A hierarchical swapped network that has l levels and uses the graph G as its nucleus is called an l-level G-based hierarchical swapped network, and is denoted

1087-4089/96 $5.00

by HSN(l, G). We define HSNs and derive their key parameters in this section. We will study HSNs based on various nucleus graphs in Sections 3, 4 and 5.

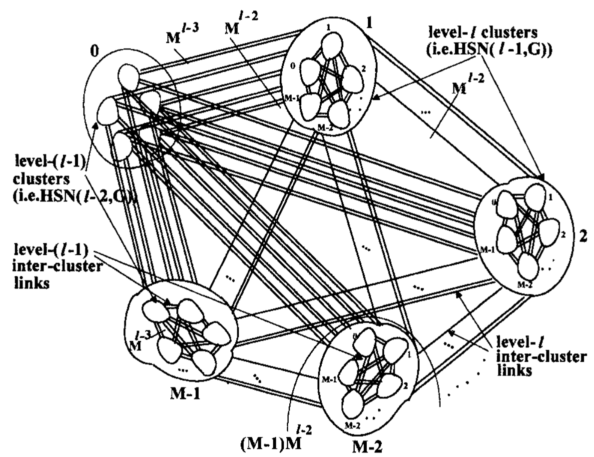

Fig. 1. Top view of an l-level hierarchical swapped network, HSN(l, G), with an M-node nucleus G.

2.1 Hierarchical Construction of Basic HSNs

An HSN(l, G) begins with a nucleus G, which forms an HSN$(1, G)$ and can be any connected graph or hypergraph (of more than one node), such as a mesh, hypercube, complete graph, star graph, or buslet. (For simplicity, we always refer to G as the nucleus "graph".)

To build a 2-level hierarchical swapped network, HSN$(2, G)$, we use M identical copies of the nucleus G, each of which has M nodes. Each nucleus is viewed as a level-2 cluster, and is given a k-bit string X_2 as its address, where $k = \lceil \log_2 M \rceil$; we also give each node a k-bit string X_1 as its address within the nucleus to which it belongs. Node X_1 within nucleus X_2 has a $2k$-bit string $X_2' = X_2X_1$ as its address within the HSN$(2, G)$. Each of the M nucleus copies has a link connecting it to each of the other $M - 1$ nuclei, via which node X_2X_1 connects to node X_1X_2. These links are called *level-2 inter-cluster links*, or simply *level-2 links*, and the connected nodes are called *level-2 neighbors*. The links connecting nodes within the same nucleus are called *nucleus links*, or *level-1 links*, and the connected nodes within the same nucleus are called *nucleus neighbors*, or *level-1 neighbors*.

To build an l-level hierarchical swapped network, HSN(l, G), we use M identical copies of HSN$(l-1, G)$. The top view of an HSN(l, G) is shown in Fig. 1. Each copy of HSN$(l - 1, G)$ is viewed as a level-l cluster, and is given a k-bit string X_l as its address; each node is already given a $k(l - 1)$-bit string $X_{l-1}' = X_{l-1:1}$ as its address within the level-l cluster to which it belongs, where $X_{i:j} = X_iX_{i-1} \cdots X_{j+1}X_j$. Node X_{l-1}' within the level-l cluster X_l has a kl-bit string $X_l' = X_lX_{l-1}' = X_{l:1}$ as its address within the HSN(l, G). Each of the M level-l clusters has M^{l-2} links connecting it to each of the other $M - 1$ level-l clusters, via which node $X_lX_{l-1:2}X_1$ connects to node $X_1X_{l-1:2}X_l$. This connectivity and the hierarchical construction are the reasons we call such networks "hierarchical swapped networks." The connecting links are called *level-l inter-cluster links*, and the connected nodes are called *level-l neighbors*. The resultant G-based l-level HSN is denoted by HSN(l, G). The recursive definition allows us to construct arbitrary-level HSNs based on any type of nucleus.

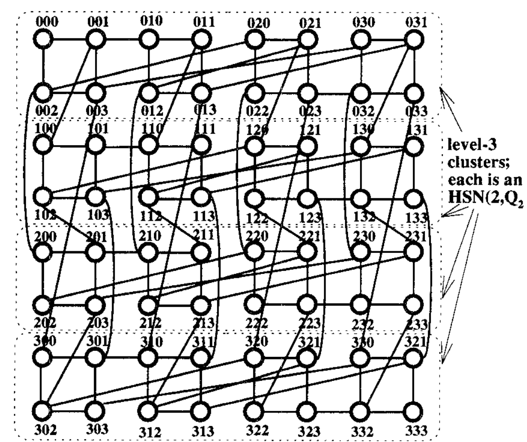

Fig. 2. The complete structure of HSN$(3, Q_2)$. Node addresses are expressed as radix-4 numbers.

The nodes that do not have a level-l inter-cluster link are called the *leaders* of that level-l cluster. Leaders can be used as I/O ports or be connected to other leaders via their unused ports to provide better fault tolerance or to improve the performance and reduce the diameter of HSNs without increasing the node degree of the network. If leader $X_l'X_{l-1:2}X_l'$ connects to leader $X_l''X_{l-1:2}X_l''$, where $X_l' = M - X_l'' - 1$, the average distance between nodes and, in most cases, the diameter of the network will be reduced. This type of HSN is called *HSN with diameter links*, and the links that connect level-l leaders are called *level-l diameter links*. The resultant G-based l-level HSN with diameter links is referred to as HSN(l, G) with diameter links. Varying the connectivity between leaders results in other classes of HSNs.

Fig. 2 shows a 3-level HSN based on the 2-cube.

2.2 Related Topologies

To obtain smaller step sizes for HSNs, we allow the number M_l of top-level clusters to be a divisor of M. Node X_1 within a nucleus connects to a node in the top-level cluster $X_1 \bmod M_l$ via its top-level inter-cluster link. For example, node $X_l X_{l-1:2} X_1$ connects to node $X_R X_{l-1:2}(X_Q M_l + X_l)$, where $X_1 = X_Q M_l + X_R$, and X_Q, X_R are positive integers with $X_R < M_l$. Most results derived in this paper can be applied to such HSN variants either directly or with minor modifications. However, to simplify the presentation, we will discuss only the case of $M_l = M$ and will use HSN to refer to such complete networks unless explicitly stated otherwise.

It is worth noting that three recently proposed interconnection networks, the hierarchical cubic network (HCN) [6], hierarchical folded-hypercube network (HFN) [5], and three-level hierarchical cubic network (3-HCN) [15], are subclasses of HSNs: HCN(k,k) is a 2-level hypercube-based HSN, HSN$(2, Q_k)$ with diameter links, where the nucleus Q_k is a k-cube; HFN(k,k) is a 2-level folded-hypercube-based HSN, HSN$(2, FQ_k)$, where the nucleus FQ_k is a k-dimensional folded-hypercube; 3-HCN is a hypercube-based 3-level HSN, HSN$(3, Q_k)$ with/without diameter links. HCN, HFN, and 3-HCN can emulate a hypercube in $O(1)$ time (assuming single-port communication), have diameters smaller than that of hypercube, and have node degrees about 1/2, 1/2, and 1/3, respectively, of that of a similar-sized hypercube. However, they require $\Theta(\log N)$ time to emulate a step of a hypercube algorithm under the all-port communication model, even though their node degrees are in the same order as that of a similar-sized hypercube.

2.3 Packet Routing

In this subsection, we present a recursive routing algorithm to route a packet from node X to node Y in an HSN(l, G).

Suppose that a routing algorithm for the nucleus G is known and that the routing algorithm for an HSN$(l-1, G)$ network is also known. Then, here is how routing is done in level l.

Let the addresses of nodes X and Y within the HSN(l, G) be $X_{l:1}$ and $Y_{l:1}$, respectively, with the bit-strings X_l and Y_l being the addresses of the level-l clusters to which nodes X and Y belong.

- Case 1: $X_l = Y_l$: Nodes X and Y belong to the same level-l cluster. We use the routing algorithm for HSN$(l-1, G)$ to route the packet, since any level-l cluster is an HSN$(l-1, G)$.
- Case 2: $X_l \neq Y_l$: Nodes X and Y belong to different level-l clusters. To route a packet from node X to node Y, we use the routing algorithm for the nucleus G to route the packet from node $X_l X_{l-1:2} X_1$ to node $X_l X_{l-1:2} Y_l$. We then send the packet to node $Y_l X_{l-1:2} X_l$ via its level-l inter-cluster link in one step, and then use the routing algorithm for HSN$(l-1, G)$ to route the packet to node $Y_l Y_{l-1:1}$. That is, the path through which the packet travels can be expressed as follows:

$$X_l X_{l-1:2} X_1 \xrightarrow{\text{nucleus}} X_l X_{l-1:2} Y_l$$

$$\xrightarrow[\text{link}]{\text{level-}l} Y_l X_{l-1:2} X_l \xrightarrow[\text{cluster}]{\text{level-}l} Y_l Y_{l-1:1}.$$

If the routing algorithm on HSN(i, G) takes at most $T_R(i)$ time steps, and the routing algorithm on nucleus G takes $T_R(1)$ time steps, the recursive routing algorithm on HSN(l, G) requires time at most

$$T_R(l) = T_R(l-1) + T_R(1) + 1 = l T_R(1) + l - 1. \quad (1)$$

2.4 Topological Properties

Let the nucleus G be a graph with M nodes of degree d_1. The number of nodes in an HSN is increased by a factor of M when the level is increased by 1. Thus, the number of nodes N of an HSN(l, G) is

$$N = N_{l-1} \cdot M = M^l, \quad (2)$$

where N_i is the number of nodes in an HSN(i, G). From Eq. 2, the level of HSN(l, G) of N nodes is

$$l = \frac{\log_2 N}{\log_2 M}. \quad (3)$$

Since the node degree is increased by 1 with each additional level, the node degree of HSN(l, G) is

$$d_l = d_1 + l - 1 = d_1 + \frac{\log_2 N}{\log_2 M} - 1. \quad (4)$$

The diameter of HSN(l, G) is obtainable from the routing algorithm given in Subsection 2.3.

Theorem 2.1 *The diameter of an HSN(l, G) (without diameter links) is $D_1 l + l - 1$, where D_1 is the diameter of the nucleus G.*

Proof: Let X' and Y' be the addresses of two nodes that have distance D_1 within the same nucleus G. It is straightforward to prove that the routing algorithm presented in Subsection 2.3 is optimal for routing from node $X = \underbrace{X'X' \cdots X'}_{l}$ to node $Y = \underbrace{Y'Y' \cdots Y'}_{l}$. Thus, the time complexity of the algorithm, assuming optimal D_1-step routing within the nucleus, provide both an upper bound and a lower bound for the diameter of HSN(l, G). □

3 Hypercube-Based HSNs

When HSNs use hypercubes as nucleus graphs, they have node degree considerably smaller than that of a similar-sized hypercube, and acquire desirable algorithmic properties. These include emulating the hypercube efficiently, performing ascend/descend operations at high speed, and generally running many algorithms with performance comparable to or even better than the hypercube.

3.1 Ascend/Descend Algorithms

"Ascend/descend" algorithms [12] require successive operations on data items that are separated by a distance equal to a power of 2. Many applications, such as Fast Fourier Transform, bitonic sort, matrix multiplication, and convolution, can be formulated using algorithms in this general category. Ascend/descend algorithms can be performed efficiently on hypercube-based HSNs.

We first present an elegant ascend algorithm on HSNs based on a hypercube, and then modify the algorithm for performing descend algorithms and normal hypercube algorithms. It is obvious that ascend algorithms can be performed on the hypercube nucleus, HSN$(1, Q_k)$. The following algorithm uses inter-cluster links to bring data which belong to nodes separated by a distance 2^i, $i > k$ into the same nucleus k-cube, and then makes use of the nucleus neighbors to perform ascend operations. This recursive ascend algorithm for HSN(l, Q_k) has 4 phases:

- Phase 1: Perform the ascend algorithm on each of the level-l clusters, which are HSN$(l-1, Q_k)$.
- Phase 2: Each node, except level-l leaders, exchanges data via its level-l inter-cluster link.
- Phase 3: Perform again the ascend algorithm on each of the nuclei, which are k-cubes (HSN$(1, Q_k)$).
- Phase 4: Each node, except level-l leaders, exchanges data via its level-l inter-cluster link again.

By performing the exchange step via level-l inter-cluster links in Phase 2, node $X_l X_{l-1:2} X_1$ will hold the data item from node $X_1 X_{l-1:2} X_l$. In essence, this moves data items separated by a distance of 2^j, $j = kl-k, kl-k+1, ..., kl-1$, into the same nucleus, such that they are now separated by a distance of 2^{j-kl+k}. In effect, we use the ascend algorithm on hypercube nucleus (HSN$(1, Q_k)$) to emulate the steps needed in the highest k dimensions.

Theorem 3.1 *Ascend/descend algorithms on an HSN(l, Q_k) can be performed in the time required on a hypercube of the same size plus $2(l-1)$ exchange steps.*

Proof: Let $T_{asc}(l, Q_k)$ denote the time required for the ascend algorithm on HSN(l, Q_k). Then we have

$$T_{asc}(l, Q_k) = T_{asc}(l-1, Q_k) + T_{asc}(1, Q_k) + 2$$
$$= lT_{asc}(1, Q_k) + 2l - 2.$$

If the time required for the ascend algorithm on the nucleus k-cube (HSN$(1, Q_k)$) is $T_{asc}(1, Q_k) = k$, then

$$T_{asc}(l, Q_k) = \log_2 N + 2l - 2.$$

To perform descend algorithms, we simply reverse the order of the phases in the ascend algorithm and replace each occurrence of "ascend" with "descend." □

Normal hypercube algorithms can be emulated in similar time complexity using a construction similar to ascend/descend algorithms. The details are omitted.

3.2 Emulating a Hypercube with Single-Dimension Communication

In this subsection, we assume single-port communication, with all the nodes only capable of using links of the same dimension at the same time. This assumption, used in some SIMD architectures and their algorithms in order to reduce the cost of implementation, is called *single-dimension communication* in this paper. We show that an HSN(l, Q_k) can emulate a hypercube of the same size with a slowdown factor 3, which is much better than the results achieved by hypernet, CCC, butterfly network, and most other hypercubic variants under this assumption.

The algorithm and its performance are given below.

- Step 1: If the computation-routing step is along dimension j, with $k(i-1) < j \leq ki$, $i = 2, 3, ..., l$, each node exchanges data via its level-i inter-cluster link.
- Step 2: Each node exchanges data via its dimension $j - k(i-1)$ nucleus link within the k-cube nucleus to which it belongs.
- Step 3: If Step 1 was executed, each node exchanges data via its level-i inter-cluster link again and then performs computation.

Theorem 3.2 *Any step of an lk-dimensional hypercube algorithm with single-dimension communication can be emulated on an HSN(l, Q_k) with single-dimension communication in 3 steps.*

3.3 Emulating a Hypercube with All-Port Communication

In this subsection, we assume all-port communication, with all the nodes capable of using links of all dimensions at the same time.

Theorem 3.3 *Any step of an lk-dimensional hypercube algorithm with all-port communication can be emulated on an HSN(l, Q_k) in $max(2k+1, l+1)$ steps.*

Proof: In Subsection 3.2, we showed that an HSN(l, Q_k) can emulate any step of an lk-cube algorithm in 3 steps with single-dimension communication. To emulate a hypercube algorithm with all-port communication, we simply perform the single-dimension emulation for all dimensions at the same time with proper scheduling. The details are omitted. □

By properly choosing the nucleus size and, as a result, the number of hierarchical levels, we can optimally emulate a hypercube on a hypercube-based HSN.

Corollary 3.4 *Any step of a hypercube algorithm with all-port communication can be optimally emulated on an HSN that uses N nodes of degree $\Theta(\sqrt{\log N})$ in $\Theta(\sqrt{\log N})$ steps.*

Proof: To optimally emulate an N-node hypercube with all-port communication, we select $k = \Theta(\sqrt{\log N})$ for the HSN(l, Q_k). Then we have

$$l = \frac{\log_2 N}{k} = \Theta(\sqrt{\log N}).$$

We know from Theorem 3.3 that the time required for emulation is $O(\sqrt{\log N})$. Since the node degree of the HSN(l, Q_k) is $\Theta(\sqrt{\log N})$, it requires $\Omega(\sqrt{\log N})$ time for a node to receive $\log_2 N$ packets. Thus, the algorithm is optimal asymptotically. □

We thus obtain several optimal communication algorithms through emulating the hypercube.

Corollary 3.5 *The total exchange task can be optimally executed on an HSN that uses N nodes of degree $\Theta(\sqrt{\log N})$ in $\Theta(N\sqrt{\log N})$ time.*

Proof: Total exchange can be performed on an N-node hypercube in $\Theta(N)$ time [2], so it can be performed in $\Theta(N\sqrt{\log N})$ time on an HSN that uses N nodes of degree $\Theta(\sqrt{\log N})$ through emulation (Corollary 3.4). Since the average inter-node distance in an N-node HSN is $\Theta(\log N)$, and the total number of packets generated when performing total exchange is $N(N-1)$, the total number of hops in the paths that these packets travel through is $\Theta(N^2 \log N)$. On the other hand, the total number of hops that can be provided in one time step by an HSN with N nodes of degree $\Theta(\sqrt{\log N})$ is at most $\Theta(N\sqrt{\log N})$. So the minimum time required for total exchange algorithm is $\Omega\left(\frac{N^2 \log N}{N\sqrt{\log N}}\right) = \Omega(N\sqrt{\log N})$. This lower bound is achieved by emulating the hypercube total exchange algorithm. □

Corollary 3.6 *The multiple-node broadcast task can be optimally executed on an HSN with N nodes of degree $\Theta(\sqrt{\log N})$ in $\Theta\left(N/\sqrt{\log N}\right)$ time.*

Proof: Multiple-node broadcasting can be performed on an N-node hypercube in $\Theta\left(N/\log N\right)$ time [2], so it can be performed in $\Theta(N/\sqrt{\log N})$ time on an HSN with N nodes of degree $\Theta(\sqrt{\log N})$ through emulation (Corollary 3.4). Since the degree of a node in the HSN is $\Theta(\sqrt{\log N})$, and each node has to receive $N-1$ packets, the minimum time required is $\Omega\left(N/\sqrt{\log N}\right)$. □

4 Complete-Graph-Based HSNs

When HSNs use complete graphs as nuclei, they gain a desirable topological property – asymptotically optimal diameter with respect to the node degree. These networks also have desirable algorithmic properties, such as efficiently emulating a hypercube or a generalized hypercube of radix M [3, 9].

4.1 Optimal Diameter

An l-level hierarchical swapped network based on a complete graph is denoted by HSN(l, K_M), where K_M is a complete graph of M nodes. In this subsection, we will show that the diameter of an HSN(l, K_M) is always optimal (asymptotically within a small constant factor) with respect to its node degree, for $M = \Omega\left(\log N/\log\log N\right)$.

From Eq. 4, the node degree of an HSN(l, K_M) is

$$M + l - 2 = M + \frac{\log_2 N}{\log_2 M} - 2.$$

The diameter of an HSN(l, K_M) is given by

$$2l - 1 = \frac{2\log_2 N}{\log_2 M} - 1$$

from Theorem 2.1. It can be seen that the diameter of an HSN(l, K_M) (with/without diameter links) is always smaller than that of a hypercube of the same size for $M \geq 4$.

Theorem 4.1 *The diameter of an HSN(l, K_M) is always optimal (asymptotically within a constant factor) with respect to its node degree for $M = \Omega(\log N / \log\log N)$, where K_M is a complete graph with M nodes.*

Proof: It is well known that the diameter of any N-node network with maximum node degree d is $\Omega(\log N / \log d)$. Substituting node degree $d = M + l - 2$, the lower bound on the diameter of HSN(l, K_M) becomes

$$D = \Omega\left(\frac{\log N}{\log(M+l)}\right). \tag{5}$$

For a nucleus of size $M = \Omega(\log N / \log\log N)$, the diameter $D = O(\log N / \log M)$ matches the lower bound

$$D = \Omega\left(\frac{\log N}{\log M + \frac{\log N}{\log M}}\right) = \Omega\left(\frac{\log N}{\log M}\right).$$

□

This property is the same as star graph for $M = \Theta(\log N / \log\log N)$ and is better than hypercube. Moreover, HSNs based on complete graphs offer wider range of optimal diameters. They can even achieve constant diameter $\Theta(1/\epsilon)$ for $M = \Theta(N^\epsilon)$, where $\epsilon = 1/l$, when the number l of hierarchical levels is a constant.

4.2 Emulating Generalized Hypercubes

Using emulation algorithms similar to those on hypercube-based HSNs, an HSN(l, K_M) can emulates a hypercube of radix M [3, 9] efficiently under various communication models. We summarize the results in the following theorems.

Theorem 4.2 *Any step of an l-dimensional radix-M hypercube algorithm with single-dimension communication can be emulated on an HSN(l, K_M) with single-dimension communication in 3 steps.*

Theorem 4.3 *Any step of an l-dimensional radix-M hypercube algorithm with all-port communication can be emulated on an HSN(l, K_M) in max$(2M + 1, l + 1)$ steps.*

Corollary 4.4 *An HSN(l, K_M) that uses N nodes of degree $\Theta(\log N / \log\log N)$ can emulate any step of an l-dimensional hypercube of radix M with degree $\Theta\left(\log^2 N / (\log\log N)^2\right)$ under the all-port communication model with optimal slowdown $\Theta(\log N / \log\log N)$.*

5 HSNs Based on Other Graphs

HSNs based on low-dimensional meshes have small node degrees, while their performance is similar to high-dimensional meshes with single-dimension communication. HSNs based on folded-hypercubes have diameters smaller than that of a hypercube of the same size. In this section, we briefly present the properties of HSN based on these nucleus graphs. We also discuss other subclasses of HSNs that have asymptotically optimal diameters.

5.1 Mesh-Based HSNs

For a constant number l of hierarchical levels and an n-D mesh nucleus M_n, an HSN(l, M_n) has constant node degree $l + 2n - 1$.

Let M_n be an n-dimensional $m_1 \times m_2 \cdots \times m_n$ mesh, and M_{nl} be an nl-dimensional $\underbrace{m_1 \times \cdots \times m_1}_{l} \times \underbrace{m_2 \times \cdots \times m_2}_{l} \times \cdots \times \underbrace{m_n \times \cdots \times m_n}_{l}$ mesh. Using emulation algorithms similar to those on hypercube-based HSNs, an HSN(l, M_n) can emulates the high-dimensional mesh M_{nl} efficiently under various communication models. We summarize the results in the following theorems.

Theorem 5.1 *Any step of an algorithm on the nl-dimensional mesh M_{nl} with single-dimension communication can be emulated on an HSN(l, M_n) with single-dimension communication in 3 steps.*

Theorem 5.2 *Any step of an algorithm on an nl-dimensional mesh M_{nl} with all-port communication can be emulated on an HSN(l, M_n) in max$(4n+1, l+1)$ steps.*

5.2 Folded-Hypercube-Based HSNs and Folded HSNs

If the nucleus of an HSN is a k-dimensional folded hypercube FQ_k, the nucleus has node degree $k+1$ and diameter $\lceil \frac{k}{2} \rceil$. As a consequence, HSN$(l, FQ_k)$ has node degree $k + l = \frac{\log_2 N}{k} + k$ and diameter $\frac{1}{2}\log_2 N + \Theta(l)$ from Eq. 4 and Theorem 2.1.

Since HSN(l, Q_k) is a subgraph of HSN(l, FQ_k), HSN(l, FQ_k) can emulate hypercube algorithms efficiently. However, such networks cannot emulate a folded hypercube efficiently since emulation of the additional complementary link would require l steps.

A variation of HSN is to add the complementary links to an HSN(l, Q_k), which we call folded

HSN(l, Q_k). Such networks not only emulate folded hypercube efficiently, but also have diameters smaller than hypercube and the corresponding HSN(l, FQ_k).

5.3 Generalized Hypercubes, Star Graphs, and Other Nucleus Graphs

A generalized-hypercube-based HSN can emulate a corresponding generalized hypercube (GQ) efficiently. For example, let GQ_{n_1,n_2} be a 2-dimensional GQ with mixed-radix (n_1, n_2), then an HSN(l, GQ_{n_1,n_2}) can emulate a $2l$-dimensional GQ with mixed-radix $(n_1, n_2, n_1, n_2, ..., n_1, n_2)$ with constant slowdown, assuming single-dimension communication. By properly choosing the nucleus size and the dimension of the nucleus GQ, such networks can have asymptotically optimal diameters with respect to the node degrees.

When HSNs use star graphs (of at least $\Omega(\sqrt{\log N / \log\log N})$ nodes) as nucleus graphs, they also have the desirable topological property of asymptotically optimal diameter.

Petersen graphs, buslets, and some other graphs or hypergraphs may also be desirable candidates for nucleus graphs. Detailed analysis of these and other subclasses of HSNs will be reported in future.

6 Conclusion

We have proposed HSNs as a new class of interconnection networks for the construction of massively parallel computers. HSNs have desirable algorithmic and topological properties, use nodes of low degree, and are modularized.

Several emulation algorithms were developed. It was shown that HSNs can emulate corresponding high-degree interconnection networks efficiently. HSNs based on various nucleus graphs were discussed. In particular, HSNs based on complete graphs, generalized hypercubes, or star graphs can have asymptotically optimal diameters. These results demonstrate that HSNs are attractive candidates for high-performance networks with reasonable cost. With various nucleus graphs and arbitrary number of hierarchical levels, HSNs can fit the needs of a wide range of applications, and satisfy the requirements of general-purpose as well as special-purpose parallel architectures.

References

[1] Akers, S.B., D. Harel, and B. Krishnamurthy, "The star graph: an attractive alternative to the n-cube," *Proc. Int'l Conf. Parallel Processing*, pp. 393-400, 1987.

[2] Bertsekas D., and J. Tsitsiklis, *Parallel and Distributed Computation – Numerical Methods*, Prentice Hall, Englewood Cliffs, NJ, 1989.

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

[4] Dekel, E., D. Nassimi, and S. Sahni, "Parallel matrix and graph algorithms," *SIAM J. Computing*, vol. 10, no. 4, Nov. 1981, pp. 657-675.

[5] Duh D., G. Chen, and J. Fang, "Algorithms and properties of a new two-level network with folded hypercubes as basic modules," *IEEE Trans. Parallel Distrib. Sys.*, vol. 6, no. 7, Jul. 1995, pp. 714-723.

[6] Ghose, K. and R. Desai, "Hierarchical cubic networks," *IEEE Trans. Parallel Distrib. Sys.*, vol. 6, No. 4, Apr., 1995, pp. 427-435.

[7] Hwang, K. and J. Ghosh, "Hypernet: a communication efficient architecture for constructing massively parallel computers," *IEEE Trans. Comput.*, vol. 36, no. 12, Dec. 1987, pp. 1450-1466.

[8] Kaushal, R.P., J.S. Bedi, "Comparison of hypercube, hypernet, and symmetric hypernet architectures," *Computer Architecture News*, vol.20, no. 5, pp. 13-25, Dec. 1992.

[9] Lakshmivarahan, S. and S.K. Dhall, "A new hierarchy of hypercube interconnection schemes for parallel computers," *J. Supercomputing*, vol. 2, 1988, pp. 81-108.

[10] Latifi, S., M. Azevedo, and N. Bagherzadeh, "The star connected cycles: a fixed-degree network for parallel processing," *Proc. Int'l Conf. Parallel Processing*, vol. I, pp. 91-95, 1993.

[11] Leighton, F.T., *Introduction to Parallel Algorithms and Architectures: Arrays, Trees, Hypercubes*, Morgan-Kaufman, San Mateo, CA, 1994.

[12] Preparata, F.P. and J.E. Vuillemin, "The cube-connected cycles: a versatile network for parallel computation," *Commun. ACM*, vol. 24, No. 5, pp. 300-309, May 1981.

[13] Scherson, I.D., and A.S. Youssef, *Interconnection Networks for High-Performance Parallel Computers*, IEEE Computer Society Press, 1994.

[14] Vecchia, G.D. and C. Sanges, "Recursively scalable networks for message passing architectures," *Proc. Conf. Parallel Processing and Applications*, pp. 33-40, 1987.

[15] Yeh, C.-H. and B. Parhami, "Parallel algorithms on three-level hierarchical cubic networks," *Proc. High Performance Computing Symp.*, 1996, to appear.

[16] Yeh, C.-H. and B. Parhami, "Swapped networks: unifying the architectures and algorithms of a wide class of hierarchical parallel processors," *Proc. Int'l Conf. Parallel Distrib. Sys.* 1996, to appear.