Swapped Networks: Unifying the Architectures and Algorithms of a Wide Class of Hierarchical Parallel Processors

Chi-Hsiang Yeh and Behrooz Parhami
Department of Electrical and Computer Engineering
University of California
Santa Barbara, CA 93106-9560, USA

Abstract

In this paper, we propose a new class of interconnection networks, called swapped networks, for general-purpose parallel processing. Swapped networks not only generate a wide class of high-performance interconnection networks, but also generalize, and serve to unify, many proposed parallel architectures as well as their algorithms. We show that swapped networks can efficiently emulate hypercubes, high-dimensional meshes, or generalized hypercubes, while having node degrees significantly smaller than the emulated network in each case. We also show that some subclasses of swapped networks can achieve asymptotically optimal diameters. Swapped networks are highly modularized, make the use of fixed-degree building blocks possible for any practically realizable system, and lead to the construction of high-performance scalable networks with reasonable cost.

1 Introduction

Many interconnection schemes for parallel architectures have been proposed in the literatures [2, 3, 4, 5, 6, 10, 11, 13]. Among them, the hierarchical cubic network (HCN) [4], hierarchical folded-hypercube network (HFN) [3], three-level hierarchical cubic network (3-HCN) [12], hypernet [5], symmetric hypernet [6], and WK-recursive network [11] offer various desirable properties. HCNs (HFNs) use (folded) hypercube networks as basic modules, and 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 in $O(1)$ time. 3-HCNs use hypercube networks as basic modules, and are composed of nodes with degree $n/3+2$, perform matrix-multiplication faster than a hypercube of the same size, and also emulate a hypercube in $O(1)$ time. Hypernets use a cubelet, treelet, or buslet as the basic module and are communication-efficient for large-scale parallel systems. Symmetric hypernets (WK-recursive networks) use a hypercube (complete graph) as the basic module and are composed of nodes with constant degree. The first three networks offer better topological and algorithmic properties, while the last two networks are more scalable; hypernets fall between these classes.

Although the structures of these networks do not resemble each other at first glance, they actually belong to the same class of hierarchical parallel architectures we call *swapped networks* and share many properties and algorithms in common. Swapped networks not only generalize, and serve to unify, these parallel architectures as well as their algorithms, but also generate a much wider class of high-performance scalable interconnection networks. In particular, we study *recursive swapped networks* based on various nucleus graphs and show how they lead to high-performance and scalability, providing tradeoffs between them. We also specify the relevant parameters of swapped networks to establish each of the six networks listed above as a special case.

A major characteristic of swapped networks is that the address of each neighbor of a node is obtained by "swapping" two equal-length bit-strings in the node address. The use of bit-string swapping as the rule for connectivity establishes swapped networks as a subclass of multi-level fully-connected (MFC) networks [13], which have more general connectivities. Swapped networks, being a subclass of MFC networks, have most of their desirable properties. In this paper we show that swapped networks can emulate hypercubes, generalized hypercubes, or high-dimensional meshes efficiently. As a consequence, we obtain a variety of algorithms on swapped networks through emulation, thus proving their potential for use in high-performance general-purpose parallel architectures.

In Section 2, we define recursive swapped networks, derive some of their parameters, and establish HCNs and HFNs as special subclasses (2-level recursive swapped networks based on hypercubes and folded-hypercubes, respectively). In Section 3, we study

0-8186-7267-6/96 $05.00

recursive swapped networks based on the n-cube. We present ascend/descend algorithms on hypercube-based recursive swapped networks. We also show how to emulate a hypercube efficiently. In Section 4, we present recursive swapped networks based on a complete graph, generalized hypercube, or mesh. In Section 5, we present general, more flexibly structured, swapped networks that can fit various applications. We construct 3-HCNs, which have smaller step size than recursive swapped networks. We establish hypernets, symmetric hypernets, and WK-recursive networks as subclasses of partially-linked swapped networks. We conclude that swapped networks are cost-effective and have desirable topological and algorithmic properties. They appear to be attractive candidates for versatile high-performance interconnection networks.

2 Recursive Swapped Networks

A swapped network is characterized by its nucleus graph, number of hierarchical levels, number of clusters in each level, and the number of links connecting each pair of clusters. A swapped network that has l levels and uses the graph G as its nucleus is called a G-based l-level swapped network, and is denoted by $\mathrm{SN}(l, G)$. In this section, we define recursive swapped networks (RSN), a simple subclass of swapped networks, whose number of clusters in each level is equal to the number of nodes in a cluster and each cluster having a link that connects it to each of the other clusters at the same level. We will study RSNs based on different nucleus graphs in Sections 3 and 4 and then generalize their constructions and definitions to the entire family of swapped networks in Section 5.

2.1 Recursive Construction of RSNs

An l-level recursive swapped network, $\mathrm{RSN}(l, G)$, begins with a nucleus G, which forms an $\mathrm{RSN}(1, G)$ and can be any connected graph or hypergraph (of more than one node) such as a mesh, hypercube, complete graph, HCN, star graph, or buslet. For simplicity, we always refer to G as the nucleus "graph".

To build a 2-level recursive swapped network, $\mathrm{RSN}(2, G)$, we use N_1 identical copies of the nucleus G, each of which has N_1 nodes. Fig. 1 shows three types of 2-level recursive swapped networks based on 4-node nuclei (2-cube, complete graph, and 1-D mesh).

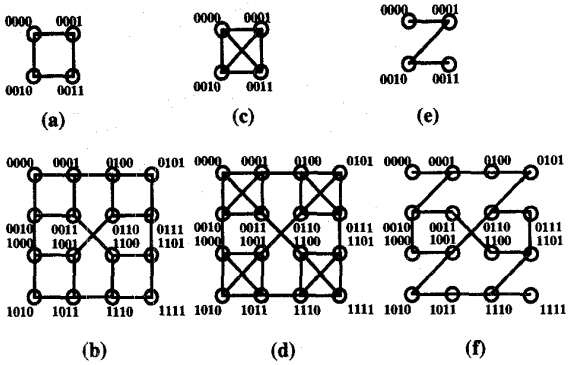

Fig. 1. Structure of 2-level recursive swapped networks. (a) nucleus 2-cube, Q_2. (b) $\mathrm{RSN}(2, Q_2)$. (c) nucleus complete graph K_4. (d) $\mathrm{RSN}(2, K_4)$. (e) nucleus 1-D mesh, M_4. (f) $\mathrm{RSN}(2, M_4)$.

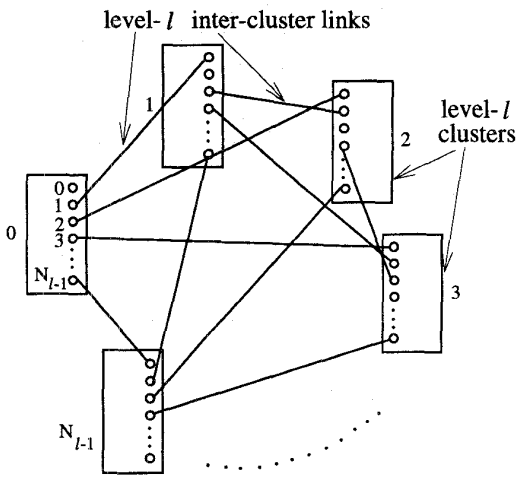

Fig. 2. Top-level connectivity of an l-level recursive swapped network.

We give each nucleus a k_1-bit string Y_1' as its address, where $k_1 = \lceil \log_2 N_1 \rceil$; we also give each node a k_1-bit string Y_1'' as its address within the nucleus to which it belongs. Node Y_1'' within the nucleus Y_1' has a k_2-bit string $Y_2'' = Y_1'Y_1''$ as its address within the $\mathrm{RSN}(2, G)$, where $k_2 = 2k_1$. Each of the N_1 nucleus copies has a link connecting it to each of the other $N_1 - 1$ nuclei, via which node $X_1'X_1''$ connects to node $X_1''X_1'$. These links are called *level-2 inter-cluster links*, the connected nodes are called *level-2 neighbors*, and nucleus copies are called *level-2 clusters*. The resultant G-based 2-level recursive swapped network is denoted by $\mathrm{RSN}(2, G)$.

To build an l-level recursive swapped network, $\mathrm{RSN}(l, G)$, we use $N_{l-1} = N_{l-2}^2$ identical copies of

RSN$(l-1, G)$, each of which has N_{l-1} nodes. The top view of an l-level RSN is shown in Fig. 2. Each copy of RSN$(l-1, G)$ is viewed as a level-l cluster, and is given a k_{l-1}-bit string Y'_{l-1} as its address, where $k_{l-1} = 2k_{l-2}$; we also give each node a k_{l-1}-bit string Y''_{l-1} as its address within the level-l cluster to which it belongs. Node Y''_{l-1} within the level-l cluster Y'_{l-1} has a k_l-bit string $Y''_l = Y'_{l-1}Y''_{l-1}$ as its address within the RSN(l, G), where $k_l = 2k_{l-1}$. Each of the N_{l-1} level-l clusters has a link connecting it to each of the other $N_{l-1}-1$ level-l clusters, via which node $X'_{l-1}X''_{l-1}$ connects to node $X''_{l-1}X'_{l-1}$. This connectivity, which is illustrated in Fig. 3, is the reason we call such networks "swapped networks." The connecting links are called *level-l inter-cluster links*, and the connected nodes are called *level-l neighbors*.

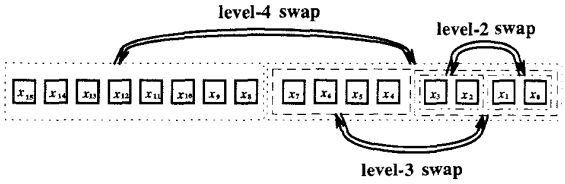

Fig. 3. The address of the level-i neighbor of node X is obtained from level-i swap, $i = 2, 3, 4$, where the address of node X is $(x_{15}, x_{14}, ..., x_0)_2$. The example shown corresponds to each node having a 2-bit address within the nucleus.

The node that does not have a level-l inter-cluster link is called the *leader* 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 swapped networks without increasing the node degree of the network. If leader $X'_{l-1}X'_{l-1}$ connects to leader $X''_{l-1}X''_{l-1}$, where $X''_{l-1} = N_{l-1} - X'_{l-1} - 1$, the average distance between nodes and, in most cases, the diameter of the network will be reduced. This type of swapped network is called *swapped network with diameter links*, and the links that connect level-l leaders are called *level-l diameter links*. The resultant G-based l-level recursive swapped network with diameter links is referred to as RSN(l, G) with diameter links. This recursive definition allows us to construct arbitrary-level recursive swapped networks based on any type of nucleus.

2.2 Subclasses of RSNs

It is worth noting that two recently proposed interconnection networks, hierarchical cubic network (HCN) [4] and hierarchical folded-hypercube networks (HFN) [3], are subclasses of 2-level recursive swapped networks: HCN(n, n) is a hypercube-based 2-level recursive swapped network, RSN$(2, Q_n)$ with diameter links, where the nucleus Q_n is an n-dimensional hypercube; HFN(n, n) is a folded-hypercube-based 2-level recursive swapped network, RSN$(2, FQ_n)$, where the nucleus FQ_n is an n-dimensional folded-hypercube. Our results show that by increasing the number of recursive levels, swapped networks can be constructed with node degrees as small as $O(\log \log N)$, while retaining good algorithmic properties and possessing even smaller diameters.

2.3 Network Size and Level

Let the nucleus G be a graph with N_1 nodes. The number of nodes in a recursive swapped network is squared when the level is increased by 1. Thus, the size (i.e., the number of nodes) N of an RSN(l, G) is

$$N = N_{l-1}^2 = N_1^{2^{l-1}}. \quad (1)$$

From Eq. (1), the number of levels of RSN(l, G) having N nodes is

$$l = \log_2 \log_2 N - \log_2 \log_2 N_1 + 1. \quad (2)$$

2.4 Node Degree

Let the nucleus G be a graph with node degree d_1. According to the definition of recursive swapped networks, the node degree is increased by 1 with each additional level. Thus, the node degree of RSN(l, G) is

$$d = d_1 + l - 1 = \log_2 \log_2 N - \log_2 \log_2 N_1 + d_1. \quad (3)$$

2.5 Packet Routing

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

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

Let the addresses of nodes X and Y within the RSN(l, G) be $X'X''$ and $Y'Y''$, respectively, with the bit-strings X' and Y' being the addresses of the level-l clusters to which nodes X and Y belong.

- Case 1: $X' = Y'$: Nodes X and Y belong to the same level-l cluster. We use the routing algorithm for RSN$(l-1, G)$ to route the packet, since any level-l cluster is also an RSN$(l-1, G)$.
- Case 2: $X' \neq Y'$: Nodes X and Y belong to different level-l clusters. The level-l inter-cluster link that connects clusters X' and Y' is $(X'Y', Y'X')$

in a recursive swapped network. To route a packet from node X to node Y, we use the routing algorithm for an $(l-1)$-level recursive swapped network, RSN$(l-1,G)$, to route the packet from node $X'X''$ to node $X'Y'$, send the packet from node $X'Y'$ to node $Y'X'$ via the level-l inter-cluster link in a step, and then use the routing algorithm for RSN$(l-1,G)$ again to route the packet from node $Y'X'$ to node $Y'Y''$.

If the routing algorithm in RSN(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 RSN(l,G) requires time at most

$$T_R(l) = 2T_R(l-1)+1 = 2^{l-1}T_R(1) + 2^{l-1} - 1. \quad (4)$$

If we use an optimal algorithm for routing in a nucleus (i.e., $T_R(1) = D_G$), we obtain an upper bound for the diameter of RSN(l,G)

$$D \leq 2^{l-1}(D_G+1) - 1,$$

where D_G is the diameter of the nucleus G. It can be proved that if the distance between nodes X and Y in a nucleus G is D_G, then the distance between nodes $\underbrace{XX\cdots X}_{2^{l-1}}$ and $\underbrace{YY\cdots Y}_{2^{l-1}}$ in an RSN(l,G) without diameter links is $2^{l-1}(D_G+1)-1$. Thus, the diameter of an RSN(l,G) without diameter links is

$$D = 2^{l-1}(D_G+1) - 1 = \frac{D_G+1}{\log_2 N_1}\log_2 N - 1. \quad (5)$$

3 Hypercube-Based RSNs

In this section, we show the desirable properties of hypercube-based recursive swapped networks. We then present algorithms for emulating a hypercube on such networks under different assumptions. We also outline the structure of elegant ascend/descend algorithms on such networks.

3.1 Basic Properties

Let the nucleus G be an n-dimensional hypercube Q_n. Then the number of nodes N_1 in the nucleus Q_n is 2^n, its node degree d_1 is n, and its diameter is n. As a consequence, a recursive swapped network based on the n-cube, RSN(l,Q_n), has $N = 2^{2^{l-1}n}$ nodes, node degree $n+l-1 = \log_2\log_2 N + n - \log_2 n$, and diameter $(1+1/n)\log_2 N - 1$ from Eqs. (1), (3), and (5). If the number of levels is a constant, say, $l=3$, then we have the number of nodes $N = 2^{4n}$, node degree $n+2 = \frac{1}{4}\log_2 N + 2$, and diameter $\log_2 N + 3$.

From Eq. (5), we have

$$D_l \leq 2D_{l-1} + 1 \leq 2^{l-2}D_2 + 2^{l-2} - 1.$$

If a recursive swapped network has level-1 diameter links, we can obtain a better upper bound on its diameter by using a result originally proved for HCN [4]: a packet can be routed in no more than $n + \lfloor n/2 \rfloor + 1$ time in an RSN$(2,Q_n)$ with diameter links (denoted by HCN(n,n) in [4]) using the OPT algorithm given in [4]. However, if all the packets are routed using the OPT algorithm congestion may occur around the diameter links when the load is not very light.

3.2 Ascend/Descend Algorithms

"Ascend/descend" algorithms [9] 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 recursive swapped networks.

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

- Phase 1: Perform the ascend algorithm on each level-l cluster, which is an RSN$(l-1,Q_n)$.
- Phase 2: All nodes, except level-l leaders, exchange data via their level-l inter-cluster links.
- Phase 3: Perform again the ascend algorithm on each cluster, which is an RSN$(l-1,Q_n)$.
- Phase 4: All nodes, except level-l leaders, exchange data via their level-l inter-cluster links again.

By performing the exchange step via level-l inter-cluster links in Phase 2, node $X'X''$ will hold the data item from node $X''X'$, where bit-strings X'' and X' are addresses of two connected nodes within the level-l clusters X' and X'', respectively, to which they belong. In essence, this moves data items separated by a distance of 2^j, $j = n2^{l-2}, n2^{l-2}+1, ..., n2^{l-1}-1$, into the same cluster, such that they are now separated by a distance of $2^{j-n2^{l-2}}$. Thus, we can then use the ascend algorithm on RSN$(l-1,Q_n)$ to emulate steps needed in the higher $n2^{l-2}$ dimensions.

Let $T_{asc}(l)$ denote the time required for the ascend algorithm on RSN(l, G). Then we have

$$T_{asc}(l) = 2T_{asc}(l-1) + 2 = 2^{l-1}T_{asc}(1) + 2^l.$$

If the time required for the ascend algorithm on the nucleus n-cube (RSN$(1, Q_n)$) is $T_{asc}(1) = n$, then

$$T_{asc}(l) = (1 + 2/n)\log_2 N.$$

It can be seen that performance of the ascend algorithms on RSN(l, Q_n) will be close to that of a hypercube of the same size for large n.

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

3.3 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 SIMD architectures and their algorithms in order to reduce the cost of implementation, is called *single-dimension communication* in this paper. We show that N-node RSN(l, Q_n) can emulate a hypercube of the same size in $O(l) = O(\log\log N - \log n)$ time, which is much better than the results achieved by CCC, butterfly network, and most other hypercubic variants under this assumption. We present this emulation algorithm in recursive form.

Assume that the emulated computation-routing step is along dimension j in a $2^{l-1}n$-dimensional binary hypercube, where $1 \le j \le 2^{l-1}n$.

Emul(j, l, n)

- Step 1: If $j > 2^{l-2}n$, each node exchanges data via its level-l inter-cluster link, and sets $j' := j - 2^{l-2}n$.
- Step 2: Perform the emulation algorithm Emul$(j', l-1, n)$ on each level-l cluster RSN$(l-1, Q_n)$.
- Step 3: If Step 1 was executed, each node exchanges data via its level-l inter-cluster link again.

The time required for the emulation algorithm Emul(j, l, n) on RSN(l, Q_n) is at most

$$T_q(l) = T_q(l-1) + 2 = T_q(1) + 2(l-1) = 2l - 1,$$

where $T_q(1) = 1$ is the time required for the emulation on a nucleus n-cube, RSN$(1, Q_n)$.

The worst cases of this algorithm occur when $(2^{l-1} - 1)n < j \le 2^{l-1}n$. In this case, exchanging data via inter-cluster links is required at all levels (i.e., levels $l, l-1, ..., 2$). Thus, any step of a $2^{l-1}n$-cube algorithm with single-port communication can be emulated in at most $2l - 1$ steps on an RSN(l, Q_n). To emulate a step of a $2^{l-1}n$-cube algorithm with all-port communication, we simply perform the Emu(j, l, n) algorithm for all j, $j = 1, 2, ..., 2^{l-1}n$ with proper scheduling. The time required for an N-node system is $O(\log N)$, which is the same as that required on an HCN$(2^{l-1}n, 2^{l-1}n)$. The time required for emulating a hypercube on an RSN is much better than what can be done on a CCC, shuffle-exchange, or other related hypercubic networks, assuming either the single-dimension or all-port communication model.

4 RSNs Based on Other Graphs

RSNs based on complete graphs or generalized hypercubes can achieve asymptotically optimal diameters; RSNs based on 2-D meshes can take advantage of the area-efficiency of 2-D meshes using current VLSI technology. Using algorithms similar to those for hypercube-based RSNs, these networks can emulate generalized hypercubes or high-dimensional meshes efficiently, assuming single-dimension communication.

4.1 Complete-Graph-Based RSNs

When recursive swapped networks use complete graphs (of at least $\Omega(\log\log N)$ nodes) as nucleus graphs, they gain the desirable topological property of having asymptotically optimal diameters with respect to their node degrees. They also possess desirable algorithmic properties, such as emulating efficiently a binary hypercube or a generalized hypercube of radix $N_1 > 2$.

An l-level recursive swapped network based on a complete graph is denoted by RSN(l, K_{N_1}), where K_{N_1} is a complete graph with N_1 nodes. The node degree of an RSN(l, K_{N_1}) is $N_1 + l - 2 = N_1 + \log_2\log_2 N - \log_2\log_2 N_1 - 2$ from Eq. (3). The diameter of RSN(l, K_{N_1}) is no more than $2^l - 1 = 2\log_2 N/\log_2 N_1 - 1$ from Eq. (5). It can be seen that the diameter of an RSN(l, K_{N_1}) (with/without diameter links) is always smaller than that of a hypercube of the same size for $N_1 \ge 4$.

It is well known that the diameter of any N-node network with maximum node degree d is $\Omega(\log N/\log d)$. Substituting the node degree $d = N_1 + l - 2$, the lower bound on the diameter of an RSN(l, K_{N_1}) becomes

$$D = \Omega\left(\frac{\log N}{\log(N_1 + l)}\right) = \Omega\left(\frac{\log N}{\log(N_1 + \log\log N)}\right).$$

For a nucleus of size $N_1 = \Omega(\log\log N)$, the diameter $D = O(\log N/\log N_1)$ matches the lower bound

$$D = \Omega\left(\frac{\log N}{\log(N_1 + \log\log N)}\right) = \Omega\left(\frac{\log N}{\log N_1}\right).$$

Thus, the diameter $\Theta(\log N/\log N_1)$ of an RSN(l, K_{N_1}) is always asymptotically optimal with respect to its node degree for $N_1 = \Omega(\log\log N)$. The diameter is comparable to that of the star graph for $N_1 = \Theta(\log N/\log\log N)$ and is better than hypercube. Moreover, recursive swapped networks based on a complete graph offer much wider range of optimal diameters. They can not only achieve optimal $\Theta(\log N/\log\log\log N)$ diameter for $N_1 = \Theta(\log\log N)$, but also as constant diameter $\Theta(1/\epsilon)$ for $N_1 = \Theta(N^\epsilon)$, where $\epsilon = 2^{-l+1}$, when the number l of hierarchical levels is a constant.

An RSN(l, K_{N_1}) can emulate a 2^{l-1}-dimensional hypercube of radix N_1 using an algorithm similar to the algorithm Emul given in Subsection 3.3. The only difference is to replace each occurrence of "n" with "1" and "Q_n" with "K_{N_1}" in the algorithm Emul. The time required is equal to $2l - 1$.

4.2 Generalized-Hypercube-Based RSNs

When recursive swapped networks use generalized hypercubes (GQ) as nucleus graphs, they can possess optimal diameter if the nucleus size and the dimension of the nucleus generalized hypercube are properly chosen. They can also emulate a corresponding generalized hypercube efficiently. For example, let GQ_{n_1,n_2} be a 2-dimensional GQ with mixed-radix (n_1, n_2). Then an RSN(l, GQ_{n_1,n_2}) can emulate a 2^l-dimensional GQ with mixed-radix $(n_1, n_2, n_1, n_2, ..., n_1, n_2)$ with $2l-1$ slowdown, assuming single-dimension communication.

4.3 Mesh-Based RSNs

The compact layout of a 2-D mesh on a VLSI chip makes it also an attractive candidate for the nucleus graph of an RSN-configured multicomputer. For a constant number of hierarchical levels l and a nucleus n-D mesh M, an RSN(l, M) has constant node degree $l + 2n - 1$.

Let the nucleus M be an n-D $m_1 \times m_2 \times \cdots \times m_n$ mesh. Then an RSN(l, M) can emulate a $2^{l-1}n$-D $m_1 \times m_2 \times \cdots \times m_n \times \cdots \times m_1 \times m_2 \times \cdots \times m_n$ mesh efficiently with a slowdown factor $2l - 1$, assuming single-dimension communication.

5 General Swapped Networks

In this section, we generalize the definition of recursive swapped networks to the entire family of swapped networks.

An l-level swapped network, SN(l, G), begins with a nucleus G, which forms an SN($1, G$) and can be any connected graph or hypergraph (of more than one node), such as mesh, hypercube, complete graph, HCN, star graph, or buslet. To build a 2-level swapped network, SN($2, G$), we use M_2 identical copies of the nucleus G, each of which has N_1 nodes. We give each nucleus an h_2-bit string Y_1' as its address, where $h_2 = \lceil \log_2 M_2 \rceil$; we also give each node a k_1-bit string Y_1'' as its address within the nucleus to which it belongs, where $k_1 = \lceil \log_2 N_1 \rceil$. Node Y_1'' within the nucleus Y_1' has a k_2-bit string $Y_2'' = Y_1'Y_1''$ as its address within the SN($2, G$), where $k_2 = h_2 + k_1$. Each of the M_2 nucleus copies has at least one link connecting it to each of the other $M_2 - 1$ nuclei. These links are called *level-2 inter-cluster links*, the connected nodes are called *level-2 neighbors*, and nucleus copies are called *level-2 clusters*. The resultant G-based 2-level swapped network is denoted by SN($2, G$).

To build an l-level swapped network, SN(l, G), we use M_l identical copies of SN($l-1, G$), each of which has N_{l-1} nodes. Each copy of SN($l-1, G$) is viewed as a level-l cluster, and is given a h_l-bit string Y_{l-1}' as its address, where $h_l = \lceil \log_2 M_l \rceil$; we also give each node a k_{l-1}-bit string Y_{l-1}'' as its address within the cluster to which it belongs. Node Y_{l-1}'' within level-l cluster Y_{l-1}' has a k_l-bit string $Y_l'' = Y_{l-1}'Y_{l-1}''$ as its address within the SN(l, G), where $k_l = h_l + k_{l-1}$. Each of the M_l level-l cluster X_{l-1}' has at least one link connecting it to each of the other $M_l - 1$ level-l cluster X_{l-1}''. The connecting links are called *level-l inter-cluster links*, and the connected nodes are called *level-l neighbors*. 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 swapped networks without increasing the node degree of the network.

This recursive definition allows us to construct arbitrary-level swapped networks based on any type of nucleus. The addresses of the neighbors of a node X are still obtained from "swapping" bit string in the address of node X. However, there are many different ways for swapping bits in the address of node X. We can give also some restrictions to "mask" the nodes that have inter-cluster links of a certain level. Moreover, the size of a general swapped network does not have to be squared when the level is increased by 1. By relaxing the composition rule, we obtain a wide class of interconnection networks, which share many topological and algorithmic properties in common.

5.1 Swapped Networks with Smaller Step Sizes

A possible drawback of recursive swapped networks is that their step sizes may be too large to be practical. Increasing the level of a recursive swapped network by one leads to squaring of the size of the resultant network. To remedy this problem, we allow the use of fewer copies of the clusters at the last level. For instance, we can connect 16 (rather than 256) copies of an RSN$(3, Q_2)$ to form a 4-level swapped network. Level-4 links connect nodes X, Y, where the address of Y is obtained from swapping the most significant 4 bits with the least significant 4 bits of X. Note that each small cluster (i.e., RSN$(2, Q_2)$) in this construction has links connecting it to all other level-4 clusters. We can, of course, obtain swapped networks using even smaller step sizes. For instance, we can connect two SN$(4, Q_2)$ to form a 5-level swapped network using similar construction rules.

As another example, 3-HCN uses $\sqrt[3]{N}$ identical copies of a $\sqrt[3]{N^2}$-node HCN as basic modules, each pair of which is connected through $\sqrt[3]{N}$ links, via which node $X'X''X'''$ is connected to node $X'''X''X'$. Since HCN is a recursive swapped network RSN$(2, Q_n)$ with diameter links, and the top-level inter-cluster links of a 3-HCN are obtained by "swapping" the most significant n bits X' of a node's address with the least significant ones X''', 3-HCN (SN$(3, Q_n)$) is a subclass of this family of swapped networks. Hierarchical swapped networks (HSN) [14], which include 3-HCN, HCN, and HFN as special cases, are also a subclass of SNs with smaller step sizes. Emulation of a hypercube on this family of swapped networks can be done in a manner similar to recursive swapped networks. The details can be found in [12, 14].

5.2 Partially-Linked Swapped Networks

In a partially-linked swapped network, only part (say, about 1/3 or 1/4) of its nodes have inter-cluster links of certain level (say, level 4), rather than each of the nodes except leaders having a link for each level as in recursive swapped networks. We present another way for swapping addresses, which is more suitable for this class of swapped networks.

We connect 16 copies of an RSN$(3, Q_2)$ to construct a 4-level swapped network as in the preceding example, while using different connection rules. Let Y be the level-4 neighbor of node X. The address of Y is obtained from swapping the most significant 4 bits of the address of node X, $(x_{11}, x_{10}, x_9, x_8)$, with bits (x_7, x_6, x_5, x_4), and the remaining bits of the address of node Y are the same as those of node X. One can arrange such that only some of the nodes have level-4 links; for example, only nodes whose least significant bit x_0 is zero may be given level-4 inter-cluster links. As a results, approximately half of the nodes do not have a level-4 link, and this is the reason we call the resultant networks "partially-linked." An example is illustrated in Fig. 4.

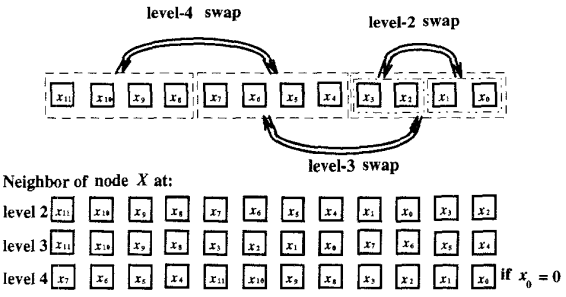

Fig. 4. The neighbors of node X in a 4-level partially-linked swapped network. Note that the level-4 neighbor of node X exists if and only if $x_0 = 0$, where the address of node X is $(x_{15}, x_{14}, ..., x_0)_2$.

If the nodes with least significant address bit $x_0 = 1$ have packets that need to be sent via level-4 links, these nodes can first send the packets to their dimension 1 intra-nucleus neighbors, where they gain access to level-4 links. It can be seen that only routing within a nucleus (or a lowest-level cluster) will be needed for nodes to share the higher-level links in this construction. This property makes it a better construction (swapping scheme) for partially-linked swapped networks than the previous one. Hypernet, symmetric hypernet, and WK-recursive networks are subclasses of partially-linked swapped networks. In what follows, we briefly establish these topologies as subclasses of partially-linked swapped networks. Detailed analysis and comparison of such networks will be reported in future.

Symmetric hypernets and WK-recursive networks use the N_1-node hypercube and complete graph, respectively, as 1-level basic modules. An l-level symmetric hypernet or WK-recursive network uses N_1 identical copies of an $(l-1)$-level network, each pair of which is connected through exactly one link, via which node $X'X''X''\cdots X''$ is connected to node $X''X'X'\cdots X'$. The top-level inter-cluster links are obtained by "swapping" the most significant k_1 bits X' of a node's address with the second most significant k_1 bits X'', where $k_1 = \lceil \log_2 N_1 \rceil$, and then arranging that only nodes whose second most significant k_1 bits of the address are the same as the third most,

fourth most,..., least significant k_1 bits are allowed to have level-l inter-cluster links. Thus, symmetric hypernets and WK-recursive networks are subclasses of partially-linked swapped networks based on different nucleus graphs. Of course, there exist other differences between them. For instance, symmetric hypernets use two different types of physical nodes (i.e., I/O nodes and processing nodes), and the use of gN_1 $(l-1)$-level symmetric hypernets to construct an l-level symmetric hypernet is allowed, where $g \geq 1$ is an integer.

Hypernets based on cubelets or buslets are also subclasses of partially-linked swapped networks. An N_1-node cubelet (buslets) is viewed as a 1-level hypernet. An l-level hypernet uses $2^{-l+1}N_{l-1}$ identical copies of an N_{l-1}-node $(l-1)$-level hypernet, each pair of which is connected through exactly one link, via which node $X'X''011\cdots1$ is connected to node $X''X'011\cdots1$. The top-level inter-cluster links are obtained by "swapping" the most significant h_l bits X' of a node's address with the second most significant h_l bits X'', where $h_l = \lceil \log_2 N_{l-1} \rceil - l + 1$. We arrange that only nodes with least significant $k - 2h_l$ bits of their addresses equal to $011\cdots1$ are allowed to have level-l inter-cluster links, where $k = \lceil \log_2 N \rceil$ and $N = 2^{-l+1}N_{l-1}^2$ is the number of nodes in the l-level hypernet. Thus, it can be seen that hypernets based on cublets or buslets are also a subclass of the partially-linked swapped networks.

These networks have the advantages of fixed node degrees and better scalability compared with recursive swapped networks. However, performance and the simplicity of algorithms on partially-linked swapped networks are inevitably traded off for the lower cost.

6 Conclusion

In this paper, we have proposed a new class of interconnection networks for modular construction of massively parallel computers. Swapped networks not only have desirable algorithmic and topological properties, but also use nodes of low degree, requiring only a small number of links per node, and are highly modularized, making them considerably less expensive to implement. Several emulation algorithms have been developed. It was shown that swapped networks can emulate several high-degree interconnection networks efficiently. By using a few data permutation steps, internode communications can be largely restricted to nodes within a much smaller cluster. As a consequence, the communication patterns of swapped networks tend to be localized. These results demonstrate that swapped networks are attractive candidates for the realization of high-performance scalable networks with reasonable cost.

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

[3] 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, pp. 714-723, July 1995.

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

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

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

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

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

[9] 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.

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

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

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

[13] Yeh, C.-H. and B. Parhami, "Unified formulation of a wide class of scalable interconnection networks based on recursive graphs," *Proc. Int'l Conf. Sys. Engr.,* 1996, to appear.

[14] Yeh, C.-H. and B. Parhami, "Hierarchical swapped networks: efficient low-degree alternatives to hypercubes and generalized hypercubes," *Proc. Int'l Symp. Parallel Architectures, Algorithms, and Networks,* 1996, to appear.