Macro-Star Networks: Efficient Low-Degree Alternatives to Star Graphs for Large-Scale Parallel Architectures

Chi-Hsiang Yeh and Emmanouel (Manos) Varvarigos
Department of Electrical and Computer Engineering,
University of California,
Santa Barbara, CA 93106-9560.

Abstract

We propose a new class of interconnection networks called macro-star networks, which belong to the class of Cayley graphs and use the star graph as a basic building module. A macro-star network can have node degree that is considerably smaller than that of a star graph of the same size, and diameter that is asymptotically within a factor of 1.25 from a universal lower bound (given its node degree). We show that algorithms developed for star graphs can be emulated on suitably constructed macro-stars with asymptotically optimal slowdown. In particular, we obtain asymptotically optimal algorithms to execute the multinode broadcast and total exchange communication tasks in a macro-star network, under both the single-port and the all-port communication models.

1 Introduction

A large variety of topologies have been proposed and analyzed in the literature [1, 11, 12, 14, 15, 17, 18, 19, 20, 23, 26, 30, 31] for the interconnection of processors in parallel computer systems. Among them, the star graph [1, 2] has received a lot of attention as an attractive alternative to the hypercube for parallel computers. The star graph belongs to the class of Cayley graphs [3], is symmetric and strongly hierarchical, and has diameter and node degree that are superior to those of a similar-sized hypercube. Also, it has been shown that a number of important algorithms can be performed efficiently on the star graph [4, 5, 6, 7, 8, 13, 21, 22, 24, 25, 27].

Even though the hypercube and the star graph have desirable topological, algorithmic, and fault-tolerance properties, their node degrees are large when the network sizes are large. To overcome this problem, constant-degree variants of these topologies, such as the cube connected cycles (CCC) [23], the de Bruijn graph [20], and the star connected cycles (SCC) [18], have been proposed and shown to have several desirable properties. Other graphs proposed as alternatives to the hypercube include the hierarchical cubic network (HCN) [14], the hierarchical folded-hypercube network (HFN) [12], and the recursive hierarchical swapped networks (RHSN) [30, 31], all of which have small degrees and diameters and can efficiently emulate hypercube algorithms.

The purpose of this paper is to develop a new family of parallel architectures that meet the following requirements: 1) small node degree, 2) small diameter, 3) symmetry properties, 4) efficient emulation of popular topologies, 5) balanced traffic, and 6) suitability for VLSI implementation. We consider the fourth requirement important since emulation of popular topologies seems to be the fastest and most cost-effective way to obtain a variety of algorithms for a new topology. Since congestion is the limiting factor on the performance when the network load is large, balanced utilization of the network links (the fifth requirement) is also important.

The macro-star (MS) networks introduced in this paper form a subclass of Cayley graphs and use the star graph as a basic building module. MS networks are vertex-symmetric, hierarchical, and modularized, and their node degree can be considerably smaller than that of similar-sized star graphs. MS networks come at various sizes and degrees, which are determined by two parameters l and n. An $\mathrm{MS}(l,n)$ network has $N=(nl+1)!$ nodes, degree $n+l-1$, and diameter $\Theta(\log N/\log\log N)$. The diameter of an $\mathrm{MS}(l,n)$ network with $l=\Theta(n)$ is asymptotically within a factor of 1.25 from a universal lower bound. We show that an MS network can emulate a star graph of the same size with asymptotically optimal slowdown. As a consequence, we obtain through emulation many efficient algorithms for the MS network, which indicates its versatility. In particular, we obtain asymptotically optimal algorithms to execute basic communication tasks, such as the multinode broadcast and the total exchange [9, 13, 22, 28, 29], assuming either single-port or all-port communication. We also show that the multinode broadcast and the total exchange task cannot be performed in any interconnection networks with

[†]Research supported partially by NSF under grant NSF-RIA-08930554.

1088-4955/96 $5.00 © 1996 IEEE

similar node degree in time that is asymptotically better by more than a constant factor than the time required in a suitably constructed MS network, assuming either single-port or all-port communication. The traffic on all the links of such MS networks is shown to be uniform within a constant factor for all algorithms considered in this paper.

The MS networks compare favorably to many other popular topologies in terms of diameter, node degree, symmetry, and algorithmic properties, and they appear to be efficient low-degree alternatives to the star graph for the construction of large-scale massively parallel systems.

2 Macro-Star Networks

In this section, we define the macro-star network and introduce some related notation.

A permutation of k distinct symbols in the set $\{1,2,...,k\}$ is represented by $U = u_{1:k} = u_1 u_2 \cdots u_k$, where $u_i \in \{1,2,...,k\}$ and $u_i \neq u_j$ for $i \neq j$, $1 \leq i,j \leq k$. On the set of all possible permutations of k symbols, we introduce the following two types of operators, which are themselves permutations and will be useful in defining the macro-star topology and describing its algorithms.

Definition 2.1 (Transposition Generator T_i) :
Given a permutation $U = u_{1:k}$, we define the *dimension-i transposition generator* T_i, $i = 2,3,...,k$, as the operator that interchanges symbol u_i with symbol u_1 in $u_{1:k}$.

In other words, for $i = 2,3,...,k$,

$$T_i(U) = u_i u_{2:i-1} u_1 u_{i+1:k},$$

where the notation $u_{j_1:j_2}$, $j_1 \leq j_2$, denotes the sequence $u_{j_1} u_{j_1+1} \cdots u_{j_2}$.

Definition 2.2 (Swap Generator $S_{n,i}$) :
Given a permutation $U = u_{1:k}$, we define the *level-i swap generator* $S_{n,i}$ as the operator that interchanges the sequence of symbols $u_{(i-1)n+2:in+1}$ with the sequence of symbols $u_{2:n+1}$ in $u_{1:k}$ where $2 \leq i \leq l$ and $k = nl + 1$.

Therefore, for $i = 2,3,...,l$, we have

$$S_{n,i}(u_{1:k}) = u_1 u_{(i-1)n+2:in+1} u_{n+2:(i-1)n+1} u_{2:n+1} u_{in+2:k}.$$

For example, for the permutation $I = 1\ 23\ 45\ 67\ 89$, we have

$$S_{2,2}(I) = 1\ 45\ 23\ 67\ 89, \quad S_{2,3}(I) = 1\ 67\ 45\ 23\ 89.$$

It can be seen that the sequence of generators $S_{n,i} T_j S_{n,i}$, which stands for the chain function

$$S_{n,i} T_j S_{n,i}(U) = S_{n,i}(T_j(S_{n,i}(U))),$$

is equivalent to the transposition generator $T_{(i-1)n+j}$ for $j = 2,3,...,n+1$.

A *macro-star network* $\mathrm{MS}(l,n)$ is a Cayley graph that has l levels of hierarchy and uses the $(n+1)$-star as a nucleus. In this paper, the integer "n" is exclusively used to mean the n in the nucleus "$(n+1)$-star;" the integer "l" is exclusively used to mean the number of hierarchical levels in the $\mathrm{MS}(l,n)$ network. For convenience, we always let $k = nl + 1$, which is the number of symbols in the permutation labeling a node of the $\mathrm{MS}(l,n)$ network. We also use S_i to signify $S_{n,i}$, suppressing the dependence on n, unless explicitly stated otherwise.

Definition 2.3 (Macro-Star $\mathrm{MS}(l,n)$ Networks) :
An l-level macro-star network based on an $(n+1)$-star is defined as the graph $\mathrm{MS}(l,n) = (\mathcal{V},\mathcal{E})$, where $\mathcal{V} = \{U = u_{1:k} | u_i, u_j \in \{1,2,...,k\},\ u_i \neq u_j \text{ for } i \neq j,\ 1 \leq i,j \leq k\}$ is the set of vertices, and $\mathcal{E} = \{(U,V) | U,V \in \mathcal{V} \text{ satisfying } U = T_j(V) \text{ or } U = S_i(V) \text{ for } 2 \leq j \leq n+1,\ 2 \leq i \leq l\}$ is the set of edges.

Clearly, the $\mathrm{MS}(l,n)$ network is a degree-$(l+n-1)$ regular graph that has $k! = (nl+1)!$ nodes, each corresponding to a permutation of $\{1,2,...,k\}$. Since MS networks form a subclass of Cayley graphs, they are vertex-symmetric.

We define the i^{th} *block* of node U as the sequence of symbols at positions $(i-1)n+2, (i-1)n+3, ..., in+1$ in the permutation of node U. According to Definition 2.3, two nodes U and V of an $\mathrm{MS}(l,n)$ network are connected by an undirected link if and only if the permutation of node V can be obtained from that of node U either by interchanging the first with the j^{th} symbol of U for some $j \in \{2,3,...,n+1\}$, or by swapping the first and the i^{th} block of U for some $i \in \{2,3,...,l\}$. The former corresponds to the actions of transposition generators, while the latter corresponds to the actions of swap generators. A link connecting node U to node $T_j(U)$, $2 \leq j \leq n+1$, will be referred to as the *dimension-j nucleus link* (or T_j link) of nodes U and $T_j(U)$. Similarly, a link connecting node U to node $S_i(U)$, $2 \leq i \leq l$, will be referred to as the *level-i inter-cluster link* (or S_i link) of nodes U and $S_i(U)$. Therefore, each node in an $\mathrm{MS}(l,n)$ network is connected to $l+n-1$ neighboring nodes through n nucleus links and $l-1$ inter-cluster links.

In order to better understand the structural and recursive properties of macro-star networks, the following definitions will be useful.

Definition 2.4 (Subgraph $\mathrm{MS}(l,n,u_{j:k})$) : Let $u_{j:k}$ be a permutation of $k-j+1$ distinct symbols in $\{1,2,...,k\}$, where $j \in \{1,2,...,k\}$. Then the graph $MS(l,n,u_{j:k})$ is defined as the subgraph $(\mathcal{V}_{u_{j:k}}, \mathcal{E}_{u_{j:k}})$ of the $\mathrm{MS}(l,n)$ network, where $\mathcal{V}_{u_{j:k}}$ is the set of nodes of $\mathrm{MS}(l,n)$ whose last $k-j+1$ symbols are equal to the sequence $u_{j:k}$, and $\mathcal{E}_{u_{j:k}}$ is the set of links of $\mathrm{MS}(l,n)$ that connect nodes in $\mathcal{V}_{u_{j:k}}$.

The sequence of symbols $u_{j:k}$ will be referred to as the *permutation* or *label* of the subgraph $\mathrm{MS}(l,n,u_{j:k})$ within the $\mathrm{MS}(l,n)$ network.

Definition 2.5 (Level-*i* Clusters): A *level-i cluster* of the $\mathrm{MS}(l,n)$ network, $i = 2,3,...,l$, is defined as the subgraph $\mathrm{MS}(l,n,u_{j:k})$, where $j=(i-1)n+2$, and $u_{j:k}$ is a permutation of $k-j+1$ distinct symbols in $\{1,2,...,k\}$.

By the definition of the MS network, a level-i cluster $\mathrm{MS}(l,n,u_{(i-1)n+2:k})$ is itself an $\mathrm{MS}(i-1,n)$ network. A level-2 cluster of the $\mathrm{MS}(l,n)$ network is an $\mathrm{MS}(1,n)$ network and is identical to an $(n+1)$-star; we will refer to a level-2 cluster as the *nucleus* of the $\mathrm{MS}(l,n)$ network. The nucleus links of an $\mathrm{MS}(l,n)$ network correspond to the links within its nucleus $(n+1)$-stars. Also a level-i inter-cluster link of an $\mathrm{MS}(l,n)$ network, $i \geq 2$, corresponds to a link connecting two level-i clusters within the same level-$(i+1)$ cluster in the MS network, where the level-$(l+1)$ cluster refers to the $\mathrm{MS}(l,n)$ network itself.

An $\mathrm{MS}(l,n)$ network has $k!/(k-n)!$ $\mathrm{MS}(l-1,n)$ subgraphs as its level-l clusters, each of which has $(k-n)!/(k-2n)!$ $\mathrm{MS}(l-2,n)$ subgraphs as its level-$(l-1)$ clusters, and so on. Thus, an MS network can be constructed recursively from identical copies of smaller MS networks. Fig. 1 shows the structure of an $\mathrm{MS}(2,2)$ network.

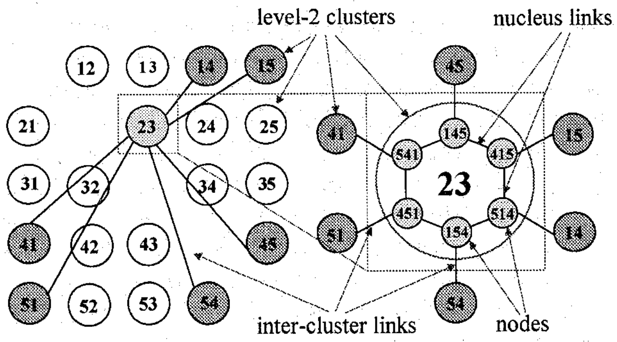

Fig. 1: The structure of an $\mathrm{MS}(2,2)$ network. All clusters that do not contain symbols 2 and 3 in their permutations have a node that is connected to some node in cluster $\mathrm{MS}(2,2,23)$.

3 Packet Routing and Topological Properties

In this section, we present an algorithm for packet routing and derive some basic properties of the MS network.

3.1 Packet Routing

As shown later (see Subsection 4.1), the $\mathrm{MS}(l,n)$ network can emulate an $(nl+1)$-star with a slowdown factor not exceeding 3, assuming single-dimension communication. Through the emulation of routing algorithms developed for the star graph [1], we can obtain simple algorithms to route a packet between any pair of nodes in an $\mathrm{MS}(l,n)$ network in at most $4.5nl+O(1)$ steps. In this section, we present a slightly more complicated algorithm that reduces the routing time to at most $2.5(nl+l-1)$ steps.

In our presentation, we assume that messages are transmitted as packets, each of which requires one unit of time (or slot) for transmission over a link. Since the MS network is vertex symmetric, we can assume, without loss of generality, that the destination is node $I = 123\cdots(k-1)k$.

The routing algorithm for the MS networks consists of two phases, both of which have similarities with routing algorithms developed for star graphs. Recall that the routing algorithm in a star graph, when the destination node is $I = 123\cdots(k-1)k$, can be viewed as "sorting" the symbols in the permutation of the source node so that symbol j appears at position j. Similarly, the goal of Phase 1 in the routing algorithm for an MS network is to place symbol j at the *proxy of position j*, defined as follows. Given a permutation t of the l distinct integers in the set $\{1,2,...,l\}$, called the *routing tag t*, the proxy of position j, $j \neq 1$, is defined as position $(t(j_1+1)-1)\cdot n+j_0+2$, where $j_0 = j-2 \bmod n$, $j_1 = \lfloor (j-2)/n \rfloor$, and $t(j_1+1)$ is the (j_1+1)-th element of tag t. (Note that $(j_1,j_0)_{(l,n)}$ is the (l,n) mixed radix representation [11] of $j-2$, and the proxy of position j is position $(t(j_1+1)-1,j_0)_{(l,n)}+2$.) Position 1 is the proxy of itself. When Phase 1 is completed, each symbol j will have fallen in its *proxy position*, the proxy of position j. Then, during Phase 2 of the routing algorithm, we use swap generators to rearrange the blocks so that symbols are placed in the order in which they appear in destination node I.

Before giving a formal description of the routing algorithm, we illustrate through the example of Fig. 2 how to route a packet from source node $X^{(0)} = 6\ 57\ 23\ 41$ to destination node $I = 1234567$ in an $\mathrm{MS}(3,2)$ network. We assume that $t = (2,3,1)$ is the initial value of the tag t (any choice of the initial routing tag would do). Since the proxy of position 6 is position 2, we first interchange symbol 6 with symbol 5. The node holding the packet after the first hop is $X^{(1)} = 5\ 67\ 23\ 41$. To put symbol 5 at its proxy position, we then have to interchange it with symbol 1. Since there is no generator in the $\mathrm{MS}(3,2)$ network to transpose the two symbols directly, we first swap blocks 1 and 3 of node $X^{(1)}$ using generator S_3, so that the node holding the packet after the 2nd hop is node $X^{(2)} = 5\ 41\ 23\ 67$. Note that the 2nd position of the grey block (Fig. 2) is the proxy of position 5 and should remain the proxy of position 5 after being swapped. This can be easily accomplished by updating the routing tag so that $t = (2,1,3)$. To complete Phase 1, symbols 5 and 1 are interchanged using the T_3 link, and the new location of the packet is node $X^{(3)} = 1\ 45\ 23\ 67$. In Phase 2, the blocks have to be rearranged so that they appear in the correct order. This can be done simply by swapping blocks 1 and 2 in the example of Fig. 2. The receiving node has permutation $I = 1\ 23\ 45\ 67$, which is the intended destination.

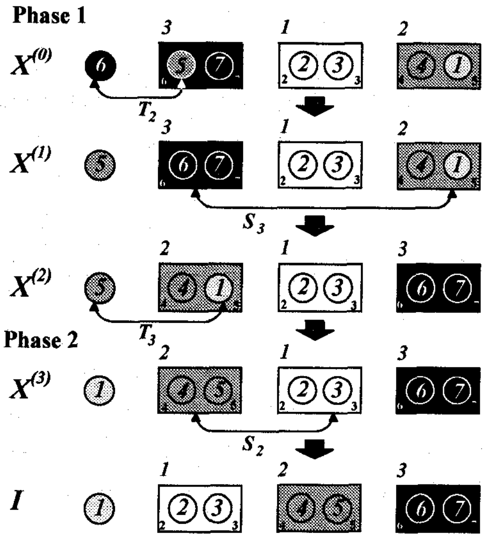

Fig. 2: Packet routing from source node $X^{(0)}$ to destination node I. The initial routing tag chosen is $t = (2,3,1)$, and the inverse tag is, therefore, $t^{-1} = (3,1,2)$. The black, white, and grey blocks are proxies of blocks 3,1, and 2, respectively. The small integers p that appear at the corners of the blocks indicate the proxy positions for symbols p, $p \in \{2,3,...,7\}$.

To formally present the routing algorithm, we let $X = x_{1:k}$ always be the permutation of the node currently holding the packet. We also define $x_{1,0} = x_1 - 2 \bmod n$ and $i = t(\lceil (x_1-2)/n \rceil + 1)$, both of which are updated immediately after each transmission. We also let t^{-1} be the inverse function of t. A block will be referred to as *dirty* if it contains at least one symbol that is not at its proxy position; otherwise, it is called *clean*.

Packet Routing Algorithm $\mathcal{A}(X, I, t, t^{-1}, \text{MS}(l,n))$

```
• Phase 1: While dirty blocks exist, do
    - 1.1: If $x_1 = 1$:
        * 1.1.1: If block 1 is clean, the packet is transmitted over the $S_q$ link, where $q$ is chosen so that block $q$ is dirty. We also set $t(t^{-1}(q)) := 1$ and $t(t^{-1}(1)) := q$, and exchange the values of $t^{-1}(q)$ and $t^{-1}(1)$.
        * 1.1.2: The packet is transmitted over the $T_p$ link, where $p$ is chosen so that $p$ is not the proxy of position $x_p$.
    - 1.2:
        * 1.2.1: If $i \neq 1$, the packet is transmitted over the $S_i$ link. We also set $t(t^{-1}(i)) := 1$, $t(t^{-1}(1)) := i$, and exchange the values of $t^{-1}(i)$ and $t^{-1}(1)$.
        * 1.2.2: The packet is transmitted over the $T_{x_{1,0}+2}$ link.
• Phase 2: While $t^{-1} \neq (1,2,...,l)$, do
    - 2.1: If $t^{-1}(1) = 1$, the packet is transmitted over the $S_q$ link, where $q$ is chosen so that $t^{-1}(q) \neq q$. We also exchange the values of $t^{-1}(q)$ and $t^{-1}(1)$.
    - 2.2: The packet is transmitted over the $S_{t^{-1}(1)}$ link. We also exchange the values of $t^{-1}(t^{-1}(1))$ and $t^{-1}(1)$.
```

Note that the goal of Phase 1 is to make all blocks clean. Also note that Phase 2 is equivalent to packet routing from source node t^{-1} to destination node $123...l$ in an l-star [1], and the blocks remain clean during the process.

3.2 Basic Properties

Recall that any permutation can be viewed as a set of *cycles* [1, 13, 16]. In the *cycle notation* each symbol's position is that occupied by the next symbol in the same cycle (cyclically). For example, the cycle notations of 341526 and 6572341 are (31),(245),6 and (6425371), respectively. In order to analyze the time required for the routing algorithm $\mathcal{A}$ above, it is useful to introduce a *proxy-cycle notation* for the nodes of the MS network. The proxy-cycle notation is similar to the cycle notation, with the difference that now it depends on the values of n and t. In a proxy-cycle notation each symbol's proxy position is that occupied by the next symbol in the cycle (cyclically). For example, the proxy cycles of 3415267 when $n = 2$ and $t = (2,3,1)$ are (32571),(46). Note that symbols transposed in consecutive iterations that execute Step 1.2 in the algorithm $\mathcal{A}$ (without executing Step 1.1) form a cycle in the proxy-cycle notation. For example, the proxy-cycle notation of the source node $X^{(0)} = 6\ 57\ 23\ 41$ with $t = (2,3,1)$ is given by

$$(651),2,3,4,7.$$

Any symbol that is located at its proxy position appears as a 1-cycle. It can be seen that, if two consecutive symbols within a proxy cycle are located in the same block of the source node (e.g., symbols 6 and 5 of node $X^{(0)}$ in Fig. 2), one step suffices to put the former symbol to its proxy position (e.g., routing from $X^{(0)}$ to $X^{(1)}$); otherwise (e.g., symbols 5 and 1 of node $X^{(0)}$), two steps are required (e.g., routing from $X^{(1)}$ to $X^{(3)}$ in the example of Fig. 2). An upper bound on the diameter of an MS(l,n) network can be obtained by counting the worst case number of communication steps in algorithm $\mathcal{A}$.

Theorem 3.1 *The diameter of the MS(l,n) network is no more than* $2.5(nl+l-1)$.

Proof: If k is odd, the worst case for Phase 1 of the routing algorithm $\mathcal{A}$ occurs when the first symbol of the source is 1, and all the other symbols form proxy cycles consisting of two symbols from different blocks. If k is even, one of the worst cases occurs when all the symbols form proxy cycles consisting of two symbols from different blocks. The time required for Phase 2 is at most $\lfloor 1.5(l-1) \rfloor$, which is equal to the diameter of an l-star [1]. Thus, the execution time of algorithm $\mathcal{A}$ is no more than $2nl$ (total count for executing Step 1.2) $+\lfloor nl/2 \rfloor$ (for Step 1.1.2) $+l-1$ (for Step 1.1.1) $+\lfloor 1.5(l-1) \rfloor$. □

Since the number of nodes in an MS(l,n) network is $N = k! = (nl+1)!$, we have that

$$k = nl+1 = \Theta\left(\frac{\log N}{\log\log N}\right), \qquad (1)$$

which implies the following corollary to Theorem 3.1.

Corollary 3.2 *The diameter of an N-node MS network is* $\Theta(\log N/\log\log N)$.

It can be shown (the proof being similar to that given in [30]) that the diameter of any MS network is asymptotically within a constant factor from a universal lower bound, given its node degree. Moreover, when $l = \Theta(n)$, the ratio of the diameter over its lower-bound for the MS(l,n) network is asymptotically no more than 1.25.

Even though the diameter of an interconnection network determines its delay under light load conditions, congestion becomes the limiting factor on the performance when the network load is large. It is therefore important that the utilization of the links is uniform, at least when the sources and destinations of the packets are uniformly distributed over all network nodes. The following theorem shows that this is indeed the case for the MS(l,n) network when $l = \Theta(n)$.

Corollary 3.3 *When the sources and destinations of the packets are uniformly distributed over all nodes of an MS(l,n) network with $l = \Theta(n)$ and the routing algorithm $\mathcal{A}$ is used, the expected traffic on the network links is equally balanced within a constant factor.*

Proof: This can be shown by computing the probability with which each link is used. The expected traffic on all inter-cluster links (also, on all nucleus links) in an MS network is the same due to symmetry. The expected traffic on an inter-cluster link is inversely proportional to $l-1$, while the expected traffic on a nucleus link is inversely proportional to n. Thus, the expected traffic is equally balanced on all the network links when $l = \Theta(n)$. □

The number l of hierarchical levels in an N-node MS(l,n) network is given by

$$l = \Theta\left(\frac{\log N}{n\log\log N}\right). \qquad (2)$$

Eq. (1) together with Eq. (2) imply that the node degree $d = n+l-1$ of an MS(l,n) network is minimized when $l = \Theta(n) = \Theta\left(\sqrt{\log N/\log\log N}\right)$, leading to the following lemma.

Lemma 3.4 *The node degree of an N-node MS(l,n) network is minimized and is equal to $\Theta\left(\sqrt{\log N/\log\log N}\right)$ if and only if $l = \Theta(n)$.*

Since $n+l-1 \leq nl = O(\log N/\log\log N)$ for any positive integers n and l, the node degree of an N-node MS(l,n) network can take value in the range from $\Omega\left(\sqrt{\frac{\log N}{\log\log N}}\right)$ to $O(\log N/\log\log N)$, depending on the particular choice of the parameters n and l.

As we will see later, the condition $l = \Theta(n)$, which leads to balanced traffic on all network links (Corollary 3.3) and guarantees minimal node degree (Lemma 3.4) and best diameter over lower-bound ratio, also results in the optimal emulation of the star graph under both the single dimension and the all-port communication models (Subsection 4.2). Moreover, the expected traffic is also uniform for algorithms emulating the star graph, assuming uniformly generated packets for neighboring nodes (Section 4). As a consequence, all these desirable properties can be achieved at the same time on an MS(l,n) network with $l = \Theta(n)$.

4 Emulation of Star Graphs

In this section, we show how to emulate algorithms developed for a k-dimensional star graph on an MS(l,n) network, where $k = nl+1$. In our emulation algorithm, a node in the k-star is one-to-one mapped on the node that has the same permutation in the MS(l,n) network. We show that algorithms developed for the star graph can be emulated on the MS(l,n) network with asymptotically optimal slowdown, assuming either single-dimension or all-port communication. In other words, the dilation and congestion of the embedding are asymptotically optimal, and the expansion and load are equal to 1.

4.1 Single-Dimension Emulation

In this subsection, we assume the *single-dimension communication* (SDC) model, where the nodes are allowed to use only links of the same dimension at any given time. This communication model is used in some SIMD architectures to reduce the cost of implementation, and is also suitable for parallel systems that use wormhole routing [10, 22]. Many algorithms developed for the star graph fall into this category. The following theorem shows that under the SDC model, the MS network can emulate a k-star with a slowdown factor not exceeding 3, which is considerably better than the slowdown factor required by an SCC graph to emulate a similar-sized star graph.

Theorem 4.1 *Any algorithm in a k-star under the SDC model can be emulated on the MS(l,n) network with a slowdown factor of 3.*

Proof: The dimension-j links T_j in a k-star can be emulated by links

$$S_{j_1+1}T_{j_0+2}S_{j_1+1},$$

where $j_0 = j-2 \bmod n$, $j_1 = \lfloor (j-2)/n \rfloor$, and $j_1 \neq 0$. That is, each node sends the packet for its dimension-j neighbor via its S_{j_1+1} link in step 1, then each node forwards the packet received in step 1 via its T_{j_0+2} link in step 2, and finally each node forwards the packet received in step 2 via its S_{j_1+1} link in step 3. It can be seen that each node receives the packet from its dimension-j neighbor (in the emulated star graph) in step 3. □

Note that the slowdown factor of 3 is also the dilation for embedding a k-star onto the MS(l,n) network. The emulation result of Theorem 4.1 permits us to find simple algorithms to execute certain prototype communication tasks in an MS network. Two basic communication tasks that arise often in applications are the multinode broadcast (MNB) and the total exchange (TE) [9, 28, 29]. In the MNB, each node has to broadcast a packet to all the other nodes of the network, while in the TE, each node has to send a different (personalized) packet to every other node of the network. Mišić and Jovanović [22], have proposed strictly optimal algorithms to execute both tasks in time $k!-1$ and $(k+1)!+o((k+1)!)$[‡], respectively, in a k-star with single-dimension communication. Using Theorem 4.1, the algorithms proposed in [22] give rise to corresponding asymptotically optimal algorithms for the MS(l,n) network.

Corollary 4.2 *The total exchange task can be performed in time $\Theta(N\log N/\log\log N)$ in an N-node MS network under the SDC model. This completion time is asymptotically optimal for the total exchange task over all interconnection networks that have N nodes and degree $O(\log^c N)$, where $c = O(1)$, assuming single-port communication.*

Proof: The completion time follows from Theorem 4.1 through emulating the TE algorithm given in [22]. It is well known that any interconnection network with N nodes of degree $O(\log^c N)$ has mean internodal distance of at least $\Omega(\log N/\log\log N)$. Therefore, the total number of packet transmissions required to execute the TE task is $\Omega(N^2\log N/\log\log N)$. Since at most N transmissions (one per node) can take place simultaneously under the single-port communication model, the corollary follows. □

When the dimensions of the links used by an algorithm in the star graph are consecutive, the algorithm can be emulated even more efficiently. For example, we can obtain an optimal algorithm (within a factor of 1, asymptotically) to execute the multinode broadcast task in the MS network, by emulating the corresponding algorithm given in [22] for star graphs. In other word, the multinode broadcast task can be performed in time $N+o(N)$ in an N-node MS network under the SDC model.

4.2 Emulation Using All-Port Communication

We now consider the all-port communication model, where a node is allowed to use all its incident links for packet transmission and reception at the same time. The packets transmitted on different outgoing links of a node can be different. In what follows, we show that the MS(l,n) network can emulate a star graph of the same size with asymptotically optimal slowdown.

Theorem 4.3 *Any algorithm in a k-star with all-port communication can be emulated on the MS(l,n) network with a slowdown factor of* $\max(2n,l)+1$.

Proof: In Subsection 4.1, we have shown that an MS(l,n) network can emulate an $(nl+1)$-star with a slowdown of 3 under the SDC model. The emulation algorithm with all-port communication simply performs single-dimension emulation for all dimensions at the same time with proper scheduling to minimize the congestion. There exist several schedules that guarantee the desired slowdown factor. The details are omitted. □

By properly choosing the parameters l and n, we can emulate a star graph with all-port communication on an MS(l,n) network with asymptotically optimal slowdown with respect to the node degrees.

Corollary 4.4 *Any algorithm in a k-star with all-port communication can be emulated on the MS(l,n) network with asymptotically optimal slowdown if $l = \Theta(n)$ [or equivalently, if the node degree is $\Theta(\sqrt{\log N/\log\log N})$].*

Proof: It follows from Lemma 3.4, Theorem 4.3, and the fact that a graph of degree $\Theta\left(\sqrt{\log N/\log\log N}\right)$ cannot emulate a graph of degree $\Theta(\log N/\log\log N)$ with a slowdown smaller than $\Theta\left(\sqrt{\log N/\log\log N}\right)$, under the all-port communication model. □

[‡] If $f(N) = o(g(N))$, $\lim_{N\to\infty} f(N)/g(N) = 0$.

Note that the slowdown factor of $\Theta\left(\sqrt{\log N/\log\log N}\right)$ is also the congestion for embedding a k-star on an MS(l,n) network with $l = \Theta(n)$. Therefore, no graph that has N nodes and degree $\Theta\left(\sqrt{\log N/\log\log N}\right)$ can embed an N-node star graph with asymptotically better congestion (by more than a constant factor) than that achieved by an MS(l,n) network with $l = \Theta(n)$.

Fragopoulou and Akl [13] have given optimal algorithms to execute the multinode broadcast and the total exchange communication tasks in a k-star with all-port communication in time $\Theta((k-1)!)$ and $\Theta(k!)$, respectively. Emulating their algorithms leads to the following asymptotically optimal algorithms.

Corollary 4.5 *The multinode broadcast task can be performed in asymptotically optimal time $\Theta\left(N\sqrt{\frac{\log\log N}{\log N}}\right)$ in an MS(l,n) network with $l = \Theta(n)$. This completion time is asymptotically optimal for the multinode broadcast task over all interconnection networks that have N nodes and degree $\Theta\left(\sqrt{\log N/\log\log N}\right)$, under the all-port communication model.*

Corollary 4.6 *The total exchange task can be performed in time $\Theta\left(N\sqrt{\frac{\log N}{\log\log N}}\right)$ in an MS(l,n) network with $l = \Theta(n)$. This completion time is asymptotically optimal for the total exchange task over all interconnection networks that have N nodes and degree $\Theta\left(\sqrt{\log N/\log\log N}\right)$, under the all-port communication model.*

Proof: Since the total exchange can be performed in an N-node star graph in $\Theta(N)$ time [13], it can be completed in time $O\left(N\sqrt{\log N/\log\log N}\right)$ in an MS(l,n) network with N nodes of degree $\Theta\left(\sqrt{\log N/\log\log N}\right)$ through emulation (Theorem 4.3), assuming all-port communication and $l = \Theta(n)$. It is well known that the mean internodal distance of an N-node graph with degree $\Theta(\log^c N)$, where $c = O(1)$, is at least $\Omega(\log N/\log\log N)$. Since the total number of packets that have to be exchanged to perform a total exchange is $N^2 - N$, the total number of packet transmissions is at least equal to $\Omega\left(N^2\sqrt{\log N/\log\log N}\right)$. Since the total number of transmissions that can take place simultaneously in an N-node interconnection network of degree $\Theta\left(\sqrt{\log N/\log\log N}\right)$ is at most equal to $O\left(N\sqrt{\log N/\log\log N}\right)$ with the all-port communication model, the time required to perform the total exchange must be at least

$$\Omega\left(\frac{\frac{N^2\log N}{\log\log N}}{N\sqrt{\frac{\log N}{\log\log N}}}\right) = \Omega\left(N\sqrt{\frac{\log N}{\log\log N}}\right).$$

This lower bound is of the same order of magnitude with the time required to emulate the total exchange algorithm [13] developed for the star graph on the MS(l,n) network with $l = \Theta(n)$. □

5 Scaling up the MS Networks

In this section, we briefly present several ways to scale up an MS network with a smaller step size, while preserving many of its desirable properties.

The first method is to provide more flexible connectivity at the top level. Consider a network that consists of $(k+n')!/k!$ identical copies of the MS(l,n) network. A node in this network is represented by $k+n'$ symbols, where $1 \le n' \le n$. Each node in this variant network is connected through an additional link to a new neighbor whose permutation is obtained by swapping the first n' symbols with the last (additional) n' symbols of the node label. Most of the results derived in this paper (for example, the emulation algorithms presented in Subsections 4.1 and 4.2) can be applied to this variant topology either directly or with minor modifications. The algorithm described in Section 3 to perform routing can also be used with minor modifications, by viewing the last n' (additional) symbols of a node as block $l+1$, and the variant topology as an $(l+1)$-level MS network. An upper bound on the diameter of the variant topology can then be found to be $2.5nl + O(n+l)$.

We can also scale up an MS network by using strategies similar to those used in the construction of the *clustered star* and *incomplete star* [19], or using strategies based on *Cayley coset graphs* [15].

6 Conclusion

In this paper, we have proposed a new class of interconnection networks for modular construction of massively parallel computers. MS networks have desirable algorithmic and topological properties, while using nodes of small degree, making them considerably less expensive to implement. We showed that MS networks have asymptotically optimal diameter, and we presented an efficient algorithm to perform routing in them. We also developed efficient algorithms to emulate the star graph, and asymptotically optimal algorithms to execute the MNB and TE communication tasks in an MS network, under both the single-port and the all-port communication models. In all routing algorithms, the expected traffic was shown to be balanced on all links of the MS(l,n) network for $l = O(n)$. We have also generalized the construction of MS networks by replacing each nucleus star graph with an MS network of the same size. We believe that the MS network and its variant topologies can fit the needs of high-performance interconnection networks, and appear to be efficient low-degree alternatives to the star graph for building large-scale 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,* 1987, pp. 393-400.

[2] Akers, S.B. and B. Krishnamurthy, "The fault tolerance of star graphs," *Proc. Int'l Conf. Supercomputing,* Vol. III, 1987, pp. 270-276.

[3] Akers, S.B. and B. Krishnamurthy, "A group-theoretic model for symmetric interconnection networks," *IEEE Trans. Comput.,* Vol. 38, Apr. 1989, pp. 555-565.

[4] Akl, S.G., K. Qiu, and I. Stojmenovic, "Fundamental algorithms for the star and pancake interconnection networks with applications to computational geometry," *Networks,* Vol. 23, Jul. 1993, pp. 215-225.

[5] Akl, S.G. and K.A. Lyons, *Parallel Computational Geometry,* Prentice Hall, Englewood Cliffs, NJ, 1993.

[6] Azevedo, M.M., S. Latifi, and N. Bagherzadeh, "Low expansion packings and embeddings of hypercubes into star graphs," *Proc. IEEE Int'l Phoenix Conf. Computers and Communications,* 1996, 115-122.

[7] Bagherzadeh, N., M. Dowd, and S. Latifi, "Faster column operations in star graphs," Dept. Elec. & Comput. Engr., Univ. California, Irvine, CA, Tech. Rep. ECE-94-02-01.

[8] Bagherzadeh, N., M. Dowd, and S. Latifi, "A well-behaved enumeration of star graphs," *IEEE Trans. Parallel Distrib. Sys.,* Vol. 6, no. 5, May 1995, pp. 531-535.

[9] Bertsekas, D.P. and J. Tsitsiklis, *Parallel and Distributed Computation – Numerical Methods,* Prentice Hall, New Jersey, 1989.

[10] Bhandarkar, S.M., "A reconfigurable 1-factor hypercubic embedding interconnection network for parallel processing," *Proc. IEEE Int'l Phoenix Conf. Computers and Communications,* 1995, pp. 152-158.

[11] 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.

[12] 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.

[13] Fragopoulou, P. and S.G. Akl, "Optimal communication algorithms on star graphs using spanning tree constructions," *J. Parallel Distrib. Computing.,* Vol. 24, 1995, pp. 55-71.

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

[15] Huang, J.-P., S. Lakshmivarahan, and S.K. Dhall, "Analysis of interconnection networks based on simple Cayley coset graphs," *Proc. IEEE Symp. Parallel and Distributed Processing,* 1993, pp. 150-157.

[16] Knuth, D.E., *The Art of Computer Programming,* Addison-Wesley, Reading, MA, 1973.

[17] Lakshmivarahan, S., J.-S. Jwo, and S.K. Dhall, "Symmetry in interconnection networks based on Cayley graphs of permutation groups: a survey," *Parallel Computing,* Vol. 19, no. 4, Apr. 1993, pp. 361-407.

[18] 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, 1993, pp. 91-95.

[19] Latifi, S. and N. Bagherzadeh, "Incomplete star: an incrementally scalable network based on the star graph," *IEEE Trans. Parallel Distrib. Sys.,* Vol. 5, no. 1, Jan. 1994, pp. 97-102.

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

[21] Menn, A. and Somani A.K., "An efficient sorting algorithm for the star graph interconnection networks," *Proc. Int'l Conf. Parallel Processing,* Vol. III, 1990, pp. 1-8.

[22] Mišić, J. and Z. Jovanović, "Communication aspects of the star graph interconnection network," *IEEE Trans. Parallel Distrib. Sys.,* Vol. 5, no. 7, Jul. 1994, pp. 678-687.

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

[24] Saikia, D.K. and R.K. Sen, "Order preserving communication on a star network," *Parallel Computing,* Vol. 21, 1995, pp. 771-782.

[25] Saikia, D.K. and R.K. Sen, "Two ranking schemes for efficient computation on the star interconnection networks," *IEEE Trans. Parallel Distrib. Sys.,* Vol. 4, Apr. 1996, pp. 321-327.

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

[27] Sheu, J.-P., C.-T. Wu, and T.-S. Chen, "An optimal broadcasting Algorithm without message redundancy in star graphs," *IEEE Trans. Parallel Distrib. Sys.,* Vol. 6, no. 6, Jun. 1995, pp. 653-658.

[28] Varvarigos, E.A. and D.P. Bertsekas, "Communication algorithms for isotropic tasks in hypercubes and wraparound meshes," *Parallel Computing,* Vol. 18, no. 11, Nov. 1992, pp. 1233-1257.

[29] Varvarigos, E.A. and D.P. Bertsekas, "Multinode broadcast in hypercubes and rings with randomly distributed length of packets," *IEEE Trans. Parallel Distrib. Sys.,* Vol. 4, no. 2, Feb. 1993, pp. 144-154.

[30] Yeh, C.-H. and B. Parhami, "Recursive hierarchical fully-connected networks: a class of low-degree small-diameter interconnection networks," *Proc. IEEE Int'l Conf. Algorithms and Architectures for Parallel Processing,* 1996, pp. 163-170.

[31] Yeh, C.-H. and B. Parhami, "Recursive hierarchical swapped networks: versatile interconnection architectures for highly parallel systems," *Proc. IEEE Symp. Parallel and Distributed Processing*, 1996, to appear.