Macro-Star Networks: Efficient Low-Degree Alternatives to Star Graphs

Chi-Hsiang Yeh, *Member*, *IEEE*, and Emmanouel A. Varvarigos, *Member*, *IEEE*

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 sublogarithmic and 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. This enables us to obtain through emulation a variety of efficient algorithms for the macro-star network, thus proving its versatility. Basic communication tasks, such as the multinode broadcast and the total exchange, can be executed in macro-star networks in asymptotically optimal time under both the single-port and the all-port communication models. Moreover, no interconnection network with similar node degree can perform these communication tasks in time that is better by more than a constant factor than that required in a macro-star network. We show that macro-star networks can embed trees, meshes, hypercubes, as well as star, bubble-sort, and complete transposition graphs with constant dilation. We introduce several variants of the macro-star network that provide more flexibility in scaling up the number of nodes. We also discuss implementation issues and compare the new topology with the star graph and other popular topologies.

Index Terms—Interconnection networks, Cayley graphs, star graphs, routing, algorithm emulation, multinode broadcast, total exchange, parallel architectures.

1 Introduction

A large variety of topologies have been proposed and analyzed in the literature [3], [11], [13], [19], [21], [24], [31], [32], [33], [40], [41], [44], [50], [53] for the interconnection of processors in parallel computing systems. Among them, the star graph [3], [4] 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 [5], 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 [6], [7], [8], [9], [10], [20], [36], [38], [42], [43], [45].

Even though the hypercube and the star graph have many desirable topological, algorithmic, and fault tolerance properties, their node degrees are large for networks of large size. To overcome this problem, constant-degree variants of these topologies, such as the cube connected cycles (CCC) [41], the de Bruijn graph [35], and the star connected cycles (SCC) [32], have been proposed and shown to have several desirable properties. Other graphs proposed as alternatives to the hypercube include hypernets [25], hierarchical cubic networks (HCN) [21], hierarchical folded-hypercube networks (HFN) [19], recursively connected complete (RCC) networks [23], hierarchical swapped networks (HSN) [51], and cyclic networks (CN) [52], all of which have small degrees and diameters and can efficiently emulate hypercube algorithms.

• *The authors are with the Department of Electrical and Computer Engineering, University of California, Santa Barbara, CA 93106-9560. E-mail: yeh@engineering.ucsb.edu, manos@ece.ucsb.edu.*

Manuscript received 20 Jan. 1997; revised 18 Dec. 1997.
For information on obtaining reprints of this article, please send e-mail to: tpds@computer.org, and reference IEEECS Log Number 100388.

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 because numerous topologies have been proposed in the literature and it is impractical, if not impossible, to develop all the useful algorithms for each of them. Therefore, the emulation of popular topologies, such as trees, meshes, hypercubes, and star graphs, 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 modular, and their node degrees can be considerably smaller than those of similar-sized star graphs. MS networks come at various sizes and degrees, which are determined by parameters l and n. An MS(l, n) network has $N = (nl + 1)!$ nodes, degree $n + l - 1$, and diameter at most equal to $\lfloor 2.5nl \rfloor + 2l - 2 = \Theta\left(\frac{\log N}{\log \log N}\right)$. Routing in an MS network can be viewed as a simple game involving "balls" and "boxes," leading to a considerable conceptual simplification of the routing algorithm. The diameter of an MS(l, n) network with $l = \Theta(n)$ (we refer to such networks as *balanced* MS networks) is asymptotically

within a factor of 1.25 from a universal lower bound (given its node degree).

We show that an MS network can emulate a star graph of the same size with asymptotically optimal slowdown under several communication models. As a consequence, we obtain through emulation many efficient algorithms for the MS network, thus indicating its versatility. In particular, we derive asymptotically optimal algorithms to execute basic communication tasks, such as the multinode broadcast (MNB) and the total exchange (TE) [12], [48], [49], assuming either single-port or all-port communication. We also show that the MNB and the TE tasks cannot be performed in an interconnection network of similar node degree in time that is asymptotically better by more than a constant factor than the time required in a balanced MS network, under both the single-port and the all-port communication models. The traffic on all the links of balanced MS networks is shown to be uniform within a constant factor for all algorithms considered in this paper. We show that MS networks can embed a variety of topologies, such as trees, meshes, hypercubes, star graphs, bubble-sort graphs [5], and complete transposition graphs [34], [35] with constant dilation.

We introduce several variants of MS networks and present ways to scale up MS networks using a smaller step size, while preserving many of their original properties. These variant topologies give us more flexibility in choosing the number of nodes in the network, without sacrificing performance or modularity in the design. We also focus on implementation issues and make a detailed comparison with star graphs. For example, we find that, when several processors are placed on a single module, MS networks require a considerably smaller number of off-module links and pins than star graphs. Because the theoretical properties of a topology do not always predict its usefulness in practice, we look into particular special cases and their suitability for building multiprocessor networks of practical size. MS networks compare favorably to many other popular topologies in terms of node degree, diameter, symmetry, and algorithmic properties, and appear to be efficient low-degree alternatives to star graphs for the construction of parallel systems at reasonable cost.

The remainder of this paper is organized as follows. In Section 2, we define MS networks and discuss their structural properties. In Section 3, we present algorithms to perform routing and compare MS networks to other popular topologies. In Section 4, we present simple and efficient algorithms for emulating star graph algorithms, and obtain optimal algorithms to execute certain prototype communication tasks in them. We also present $O(1)$-dilation embeddings of several important topologies on MS networks. In Section 5, we introduce several variants of MS networks and compare MS networks and star graphs with respect to several implementation considerations. Finally, in Section 6, we conclude the paper.

2 Macro-Star Networks

In this section, we define the macro-star (MS) network topology and introduce some related notation. To provide some intuition and better visualize the topology, we first relate the MS network to a game involving "boxes" and "balls." The reader could visualize each distinct state of the game as a different node of the MS network, each possible movement in the game as a link connecting two nodes of the MS network.

2.1 A Balls-to-Boxes Game

We are given l boxes, each of which is assigned a distinct color in $\{1, 2, ..., l\}$, and $k = nl + 1$ balls. $k - 1$ of the balls are partitioned into l groups of size n, each of which is assigned a distinct color in $\{1, 2, ..., l\}$, while the remaining ball is assigned color 0 and does not belong to any group (see also Fig. 2). Initially, $k - 1$ of the balls are mixed together in the l boxes so that each box contains n balls (of different colors, in general) and one ball is left outside the boxes. The goal of the game is to rearrange the balls and the boxes so that each ball ends up in a specific position in the box that has the same color, except for the ball with color 0 that ends up outside the boxes. Also, the boxes should be sorted so that the box of color i, $i \in \{1, 2, ..., l\}$, appears in the ith position from the left. At any time in the game, the ball that is currently outside all boxes will be called the *outside ball*, while the box currently at position 1 will be called the *leftmost box*. At each step, the player can take one of the following actions:

1) Exchange the outside ball with one of the balls in the leftmost box, or
2) Exchange the leftmost box with any of the other boxes.

Note that there are $N = (nl + 1)!$ distinct placements (configurations) of balls to boxes and $n + l - 1$ possible movements from one configuration to another.

A ball that is currently in a box of color different than its own color, or a ball that is at the wrong position in a box of the same color, will be referred to as a *dirty ball*. A box that contains at least one dirty ball will be referred to as a *dirty box*. A ball or box that is not dirty will be called *clean*. It is easy to verify that the following algorithm solves the Balls-to-Boxes game.

Balls-to-Boxes Algorithm

- Phase 1
  - Case 1.1: If the outside ball has color 0:
    - 1.1.1: If all boxes are clean, go to Phase 2; if the leftmost box is clean, exchange it with a dirty box and go to Step 1.1.2.
    - 1.1.2: Exchange the outside ball (which has color 0) with any dirty ball in the leftmost box and go to Case 1.2.
  - Case 1.2: If the outside ball has color c different than 0:
    - 1.2.1: If the color of the leftmost box is different than c, then swap the leftmost box with the box of color c and go to Step 1.2.2.
    - 1.2.2: If the outside ball is of the same color as the leftmost box, then put the outside ball at its correct position in the leftmost box, take the ball occupying that position outside, and go to Phase 1.

- Phase 2: Now, all boxes are clean (they contain balls of the correct color, placed at their correct positions), but they may not be in the correct order. To put them in the correct order so that box of color i is placed at position i, the following algorithm is run:
  - 2.1: If the leftmost box has color 1, then exchange it with any box that is not at its correct position.
  - 2.2: If the leftmost box has color $i \neq 1$, exchange it with the box at the ith position.
  - 2.3: If all boxes appear in the correct order, then stop (the goal of the algorithm has been accomplished); otherwise, go to Step 2.1.

2.2 Definition of Macro-Star Networks

Recall that there are $(nl + 1)!$ distinct configurations (states) of balls to boxes for the game involving l boxes, each having n balls, and an outside ball. The macro-star network, MS(l, n), is obtained by drawing the state transition graph for the game. In other words, each of the $(nl + 1)!$ states corresponds to a vertex in an MS(l, n) network, and two vertices are connected if and only if one of their corresponding states can be obtained from the other by performing one of the $n + l - 1$ possible actions. In what follows, we formally define the MS(l, n) network, each node of which will be represented as a permutation of $k = nl + 1$ symbols.

A permutation of k distinct symbols in the set $\{1, 2, ..., k\}$ is represented by $U = u_{1:k} = u_1u_2 \cdots u_k$, 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.

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 $i = 2, 3, ..., k$,

$$T_i(U) = u_iu_{2:i-1}u_1u_{i+1:k},$$

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

$$S_{n,i}(u_{1:k}) = u_1u_{(i-1)n+2:in+1}u_{n+2:(i-1)n+1}u_{2:n+1}u_{in+2:k}.$$

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

$$T_2(I) = 2\ 13456789,\ T_5(I) = 5\ 23416789,\ T_8(I) = 8\ 23456719,$$

and

$$S_{2,2}(I) = 1\ 45\ 23\ 67\ 89,\ S_{2,3}(I) = 1\ 67\ 45\ 23\ 89,$$
$$S_{2,4}(I) = 1\ 89\ 45\ 67\ 23.$$

The k-star graph [3] has $k!$ nodes, each represented by a permutation of the symbols in $\{1, 2, ..., k\}$, whose links are defined by the application of generators $T_2, T_3, ..., T_k$ on the node label. It can be seen that the sequence of generators $S_{n,i}T_jS_{n,i}$, which stands for the chain function

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

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

A macro-star network MS(l, n) has l levels of hierarchy and uses the $(n + 1)$-star as a basic building module (to be referred to as the nucleus). In this paper, the integer "n" is exclusively used to signify the n in the nucleus "$(n + 1)$-star"; the integer "l" is exclusively used to signify the number of hierarchical levels in the MS(l, n) network. We also let $k = nl + 1$ be the number of symbols in the permutation labeling a node of the MS(l, n) network. In what follows, we will use S_i instead of $S_{n,i}$, suppressing the dependence on n, unless explicitly stated otherwise.

DEFINITION 2.3 (Macro-Star MS(l, n) Networks). *An l-level macro-star network based on an $(n + 1)$-star is defined as the graph* MS(l, n) = $(\mathcal{V}, \mathcal{E})$, *where*

$$\mathcal{V} = \{U = u_{1:k} \mid u_i, u_j \in \{1, 2, ..., k\}, u_i \neq u_j \text{ for } i \neq j, 1 \leq i, j \leq k\}$$

is the set of vertices and

$$\mathcal{E} = \{(U, V) \mid U, V \in \mathcal{V} \text{ satisfying } U = T_j(V) \text{ or } U = S_i(V) \text{ for } 2 \leq j \leq n + 1, 2 \leq i \leq l\}$$

is the set of edges.

We define the ith 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 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 jth symbol of U for some $j \in \{2, 3, ..., n + 1\}$, or by swapping the first and the ith 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 a dimension-j nucleus link (or T_j link). Similarly, a link connecting node U to node $S_i(U)$, $2 \leq i \leq l$, will be referred to as the level-i intercluster link (or S_i link). Therefore, each node in an MS(l, n) network is connected to $l + n - 1$ neighboring nodes through n nucleus links and $l - 1$ intercluster links.

Clearly, the MS(l, n) network is a degree-$(l + n - 1)$ Cayley graph [5] 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 and regular.

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

DEFINITION 2.4 (Subgraph 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* MS(l, n) *network, where*

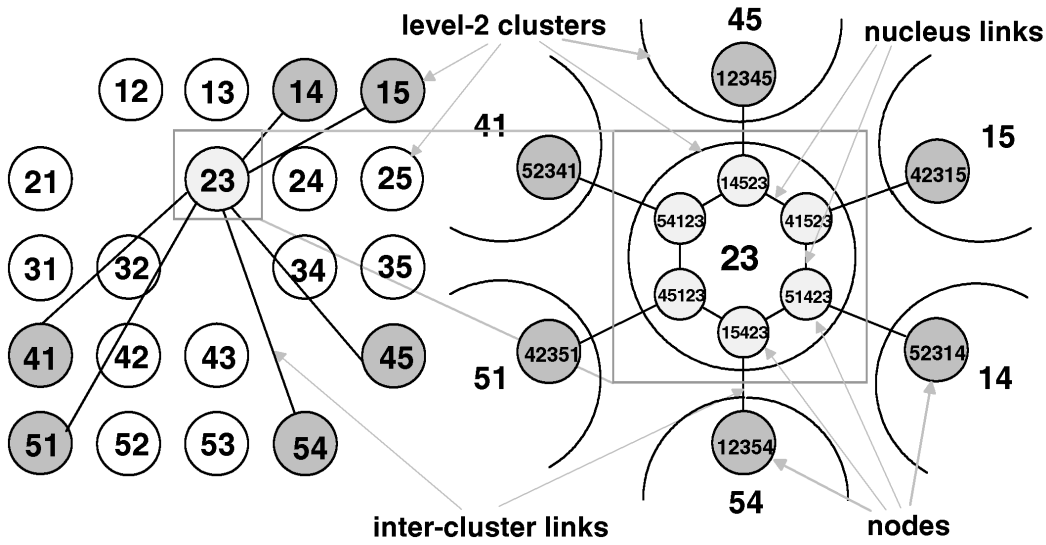


Fig. 1. The structure of an MS(2, 2) network. In Fig. 1a, each circle corresponds to a level-2 cluster and consists of the set of all nodes whose two rightmost symbols are equal to those indicated in the circle. All clusters that do not contain symbols 2 and 3 in their permutations have a node that is connected to some node in cluster MS(2, 2, 23). Fig. 1a illustrates only the intercluster links connecting cluster MS(2, 2, 23) to other clusters, while Fig. 1b illustrates the nucleus (internal) links of cluster MS(2, 2, 23).

$\mathcal{V}_{u_{j:k}}$ is the set of nodes of 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* 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 MS(l, n, $u_{j:k}$) within the MS(l, n) network.

DEFINITION 2.5 (Level-i Clusters). *A level-i cluster of the* MS(l n) *network,* $i = 2, 3, ..., l$, *is defined as the subgraph* 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* $\{1, 2, ..., k\}$.

In other words, MS(l, n, $u_{j:k}$) is the subgraph consisting of nodes whose symbols in blocks i, $i + 1$, ..., l form the sequence $u_{j:k}$. By the definition of the MS network, a level-i cluster MS(l, n, $u_{(i-1)n+2:k}$) is itself an MS($i - 1$, n) network. A nucleus of an MS(l, n) network is a level-2 cluster, which is an MS(1, n) network and is identical to an $(n + 1)$-star. The nucleus links of an MS(l, n) network correspond to the links within its nucleus $(n + 1)$-stars. Also, a level-i intercluster link of an 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 MS(l, n) network itself. An MS(l, n) network has $k!/(k - n)!$ MS($l - 1$, n) subgraphs as its level-l clusters, each of which has $(k - n)!/(k - 2n)!$ MS($l - 2$, n) subgraphs as its level-$(l - 1)$ clusters, and so on. Thus, an MS network can be constructed in a hierarchical way from identical copies of smaller MS networks.

Fig. 1a shows a "top view" of an MS(2, 2) network, while Fig. 1b illustrates the details of the level-2 cluster MS(2, 2, 23), which is itself a 3-star (and also a ring), and the way it is connected to other level-2 clusters of the MS(2, 2) network. The level-2 clusters of the MS(2, 2) network are arranged along the points of a 5×5 grid. Clusters labeled by permutations that have two or more identical symbols are not present in the MS(2, 2) network. All clusters that do not contain symbols 2 and 3 in their permutations (that is, all the shaded clusters in Fig. 2) have a node that is connected to some node in cluster MS(2, 2, 23).

2.3 The Routing Problem as a Game

A little thought shows that the routing problem in an MS network is essentially equivalent to the Balls-to-Boxes game described in Section 2.1. Symbols of the node label in the routing problem correspond to balls, while blocks of symbols corresponds to boxes. The leftmost symbol of the node label corresponds to the ball that is currently outside the boxes, and the ball of color 0 corresponds to symbol 1 in the routing problem. Transmissions over intercluster links correspond to the swapping of two boxes, while transmissions over nucleus links correspond to the interchange of the outside ball with a ball inside the leftmost box.

To illustrate the Balls-to-Boxes routing algorithm through an example, we show in Fig. 2 how to route a packet from node $X^{(0)}$ = 6 57 23 41 to destination node I = 1234567 in an MS(3, 2) network. We arbitrarily assign the colors for blocks (boxes) 1, 2, and 3 of the source node to be colors 3, 1, and 2, respectively. The color of each symbol (ball) is defined to be the color $i \in \{1, 2, ..., l\}$ of the block (box) at which it appears in the destination address. In the example, the destination is node I = 1234567 and, therefore, the color of symbol x is equal to $\lceil (x - 1)/n \rceil$. Since symbol 6 has the same color 3 with the leftmost box (Step 1.2.2 of the Balls-to-Boxes Algorithm), we first interchange symbol 6 with symbol 5, which occupies the current desired position of symbol 6. The node holding the packet after the first hop is $X^{(1)}$ = 5 67 23 41. To place symbol 5 at its desired position, we then have to interchange it with symbol 1. Since there is no generator in the 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 (Step 1.2.1), moving the packet to node $X^{(2)}$ = 5 41 23 67.

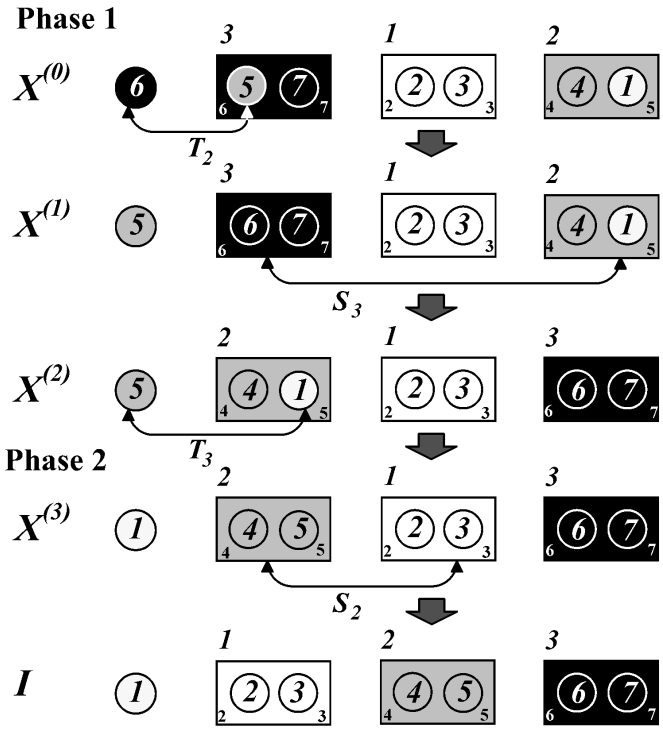


Fig. 2. Packet routing from source node $X^{(0)}$ = 6572341 to destination node I. The small integers p that appear at the corners of the blocks indicate the desired positions for symbols p, $p \in \{2, 3, ..., 7\}$.

(Note that the second position of the gray block in Fig. 2 is the desired position of symbol 5 and should remain the desired position of symbol 5 after the boxes are swapped.) We then interchange symbols 5 and 1 using generator T_3 (Step 1.2.2), and the new location of the packet is node $X^{(3)}$ = 1 45 23 67. All symbols (balls) have now been placed at their correct position in the block (box) that has the same color, and Phase 1 of the algorithms has been completed. In Phase 2, the blocks (boxes) 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 (Step 2.2). The receiving node has label I = 1 23 45 67, which is the intended destination.

3 Routing and Topological Properties

In this section, we present algorithms for performing routing and derive some basic properties of the MS network.

3.1 Routing in an MS Network Based on Any Star-Graph Routing Algorithm

In this subsection, we develop MS routing algorithms based on (any) corresponding star-graph routing algorithms. Since the MS network is vertex symmetric, we will assume, without loss of generality, that the destination is node $I = 123 \cdots (k-1)k$.

Recall that any permutation π of {1, 2, ..., k} can be viewed as a set of *cycles* [3], [29], where a cycle is defined as an ordered set of symbols $(s_1 s_2 \cdots s_c)$ such that $\pi(s_1) = s_2$, $\pi(s_2) = s_3$, ..., $\pi(s_c) = s_1$, where $\pi(x)$ is the xth symbol of π. In other words, in *cycle representation*, each symbol's position is that occupied by the next symbol in the same cycle (cyclically). For example, (6425371) and (371)(245)(6) are cycle representations of permutations 6572341 and 3475261, respectively. It is well-known [3], [5] that the routing algorithm in a star graph (or, more generally, a Cayley graph) can be viewed as "sorting" the symbols in the label of the source node so that symbol j appears at position j when the destination node is $I = 123 \cdots (k-1)k$. To describe the routing algorithm for a star graph, let a cycle representation of the source node be

$$\left(s_{1,1}s_{1,2} \ \ldots \ s_{1,l_1-1}1\right)\left(s_{2,1}s_{2,2} \ \ldots \ s_{2,l_2}\right) \cdots \left(s_{c,1}s_{c,2} \ \ldots \ s_{c,l_c}\right),$$

where $s_{1,1}$ is the first symbol of the source label. The sorting of symbols can be performed by first interchanging symbol $s_{1,1}$ with symbol $s_{1,2}$, and then interchanging symbol $s_{1,2}$ with symbol $s_{1,3}$, ..., symbol s_{1,l_1-2} with symbol s_{1,l_1-1}, and symbol s_{1,l_1-1} with symbol 1. For i = 2, 3, 4, ..., c and $l_i > 1$, we interchange symbol 1 (which always appears at position 1 at the end of an iteration) with symbol $s_{i,1}$ and, then, interchange symbol $s_{i,1}$ with symbol $s_{i,2}$, symbol $s_{i,2}$ with symbol $s_{i,3}$, ..., symbol s_{i,l_i-1} with symbol s_{i,l_i}, and symbol s_{i,l_i} with symbol 1. Note that the cycles, except for the first one, can be permuted, and the symbols in each of them can be cyclically shifted to obtain alternative routing paths that have the same length. For example, to route a packet from source 3475261 = (371)(245)(6) to node I in a star graph, we can interchange symbol 3 with symbol 7, symbol 7 with symbol 1 (for iteration 1) and, then, interchange symbol 1 with symbol 2, symbol 2 with symbol 4, symbol 4 with symbol 5, and, finally, symbol 5 with symbol 1 (for iteration 2). An $(ln + 1)$-star has ln generators that can interchange the first symbol with any other symbol of the node label, while an MS(l, n) network has only $l + n - 1$ generators. Therefore, some symbol interchanges that are possible in a star graph cannot be performed in an MS network in a single step.

The following theorem shows that the MS(l, n) network can emulate an $(nl + 1)$-star with a slowdown factor not exceeding three, assuming the *single-dimension communication* (SDC) model, where the nodes are allowed to use only links of the same dimension at any given time.

THEOREM 3.1. *Any algorithm in an* $(ln + 1)$*-star under the SDC model can be emulated on the* MS(l, n) *network with a slowdown factor of three.*

PROOF. The dimension-j links T_j in an $(ln + 1)$-star can be emulated by the paths consisting of links

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

in an MS(l, n) network, where $j_0 = j - 2 \bmod n$ and $j_1 = \lfloor (j-2)/n \rfloor$, when $j_1 \neq 0$. 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. When $j_1 = 0$, emulating the T_j links requires only one step. □

Note that the *dilation* for embedding an $(ln + 1)$-star in an MS(l, n) network is also equal to 3. That is, if we map each node of the $(ln + 1)$-star onto a node in an MS(l, n) network, and map each link of the $(ln + 1)$-star onto a path in the MS(l, n) network, the maximum length of such paths is equal to 3. The maximum number of such paths that are mapped onto a link in the MS(l, n) network is called the *congestion* of the embedding. The congestion for embedding an $(ln + 1)$-star in an MS(l, n) network is equal to $\max(2n, l)$. However, the congestion for embedding all the links of a certain dimension i in an MS(l, n) network is only 2 when $i > n + 1$ and is equal to 1 otherwise. Therefore, the slowdown factor for an MS network to emulate a star-graph algorithm under the SDC model is approximately equal to 2 if the network uses wormhole or cut-through routing or if it uses packet switching and each node has many packets to be sent along a certain dimension.

Through the emulation of routing algorithms developed for the star graph, we can obtain simple algorithms to route a packet between any pair of nodes in an MS(l, n) network in at most $3\lfloor 3nl/2 \rfloor \leq 4.5nl$ steps. In what follows, we present a slightly more complicated algorithm, to be referred to as the MS routing algorithm, that significantly reduces the routing time.

The proposed MS routing algorithm consists of two phases, both of which have similarities with routing algorithms developed for star graphs. Consider any particular path from a source node to the destination node I in an $(ln + 1)$-star or, equivalently, any sequence of transposition generators that sort the source label into the destination label I. In Phase 1 of the MS routing algorithm, we interchange symbols in exactly the same order as in the $(ln + 1)$-star. If the next symbol to be interchanged belongs to block 1, the transposition can be performed in one step using a transposition generator (that is, sending the packet over a nucleus link); if it belongs to block $i > 1$, we first swap block 1 with block i using generator S_i (that is, sending the packet over an S_i link) and, then, transpose the two symbols using a transposition generator. When Phase 1 is completed, symbol 1 will appear at position 1, and each block will contain symbols belonging to a given block of destination I in ascending order. In Phase 2 of the MS routing algorithm, we use swap generators to rearrange the blocks so that symbols appear in the order in which they appear at destination node I. This can be done by using any routing algorithm developed for an l-star, and viewing the sequence of symbols belonging to block i of destination node I as a "super-symbol" i.

Phase 1 of the MS routing algorithm essentially removes the third transmission (on an S_{j_1+1} link) from the direct emulation of Theorem 3.1 and, thus, improves the execution time by a factor of approximately 1.5. This modification may result in blocks appearing in incorrect order, which can be corrected in at most $\lfloor 3(l - 1)/2 \rfloor$ steps during Phase 2 of the algorithm.

In the following section, we show that by adding some restrictions on the star-graph routing algorithm that is emulated, the required time can be further improved.

3.2 Faster MS Routing Algorithms

As we have pointed out in Section 2.3, when the balls are viewed as symbols in the permutation label of a node, and the boxes as blocks of symbols, any algorithm that solves the Balls-to-Boxes game immediately gives rise to a corresponding algorithm for performing routing in an MS network. The Balls-to-Boxes algorithm introduced in Section 2.1 is actually equivalent to imposing a special order for the cycles of the source node and for the symbols within each cycle such that the last element of a cycle has the same color as the first element of the next cycle (that is, both elements belong to the same block in the destination label), unless no such cycle or element exists. As we will show later, this restriction reduces the maximum number of swap generators (and, thus, the corresponding intercluster links) required, reducing the routing time by approximately $nl/2$ steps for the worst case scenario.

We analyze the required time for the Balls-to-Boxes algorithm as follows: Again, we can assume without loss of generality that the destination is node $I = 123 \cdots k$. The number of steps required for Phase 2 of the Balls-to-Boxes algorithm is at most $\lfloor 1.5(l - 1) \rfloor$, which is equal to the diameter of an l-star [3]. If $k = ln + 1$ is odd, the worst case for Phase 1 occurs when the first symbol of the source label is 1, and the cycle representation of the source label consists of cycles of two symbols from different blocks of the source label. If k is even, one of the worst cases occurs when all the symbols form cycles consisting of two symbols from different blocks of the source label. Thus, the total number of steps required by this routing algorithm is at most equal to

1) $2nl$ steps (which is two times the maximum number of symbols that have to be interchanged), corresponding to the total count for executing Step 1.2 of the Balls-to-Boxes algorithm, plus
2) $\lfloor nl/2 \rfloor$ steps (which is -1 plus the maximum number of cycles that have length larger than 1), corresponding to the total count for Step 1.1.2, plus
3) $l - 1$ steps (which is the number of blocks minus 1), corresponding to the total count for Step 1.1.1, plus
4) $\lfloor 1.5(l - 1) \rfloor$ steps for Phase 2,

for a total of at most $\lfloor 2.5(nl + l - 1) \rfloor$ steps.

Further improvements in the routing time can be obtained by merging Phases 1 and 2 of the Balls-to-Boxes algorithm. More precisely, after a box becomes clean, we place it to its final position right away, rather than waiting until Phase 2 of the Balls-to-Boxes algorithm. If this can be done then, when all the boxes are cleaned and placed at their correct positions, they will appear in the correct order and we will not need the $\lfloor 1.5l \rfloor$ steps required by Phase 2 for reordering the boxes. This is equivalent to adding a restriction on Phase 1.1.1 of the algorithm in Section 2.1 that the box that is exchanged is the one occupying the desired position of the current leftmost box. Note that the box to be exchanged may be either dirty or clean. An exception occurs when the leftmost box that was just cleaned or exchanged should finally appear in the leftmost position, in which case we exchange it with a dirty box or a clean box that is not at its final position. At most $l - 1$ exchanges of the latter type may have to be made, for a total of at most

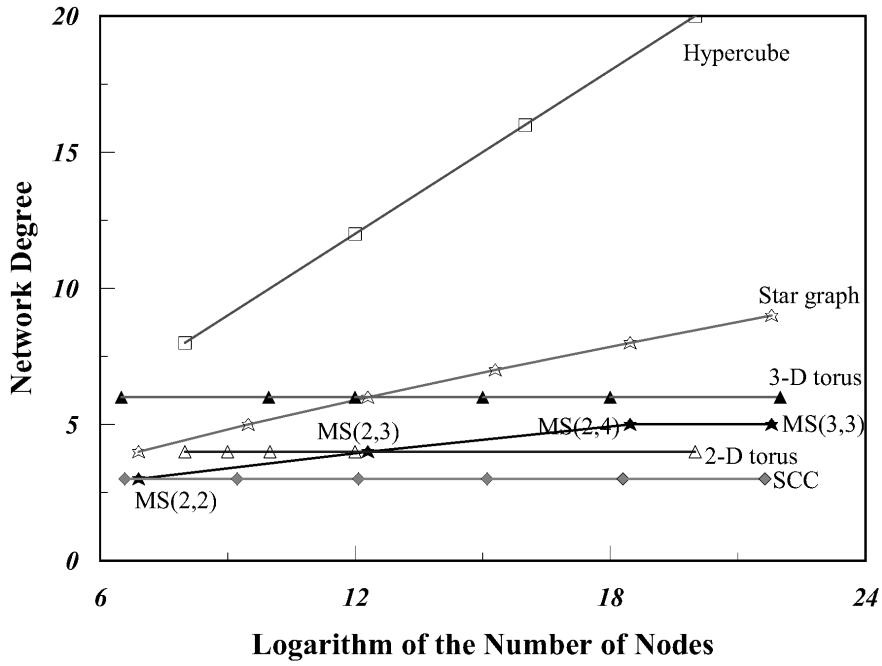


Fig. 3. Comparison of the node degrees of various interconnection networks.

$2l - 2$ steps. Therefore, this improved algorithm requires $2nl + \lfloor nl/2 \rfloor + 2l - 2$ steps, where $2nl + \lfloor nl/2 \rfloor$ is the number of steps required for Parts 1 and 2 in the previous analysis.

3.3 Basic Properties

In this section, we derive the diameter, node degree, and other basic properties of MS networks and compare them with several popular topologies.

THEOREM 3.2. *The diameter of the* MS(l, n) *network is at most equal to* $\lfloor 2.5nl \rfloor + 2l - 2 = \Theta\left(\frac{\log N}{\log \log N}\right)$, *where N is the number of nodes.*

PROOF. The upper bound $\lfloor 2.5nl \rfloor + 2l - 2 = \lceil 2.5k \rceil + 2l - 5$ on the diameter of MS networks follows from the analysis at the end of Section 3.2. Since an MS(l, n) network has $N = k! = (nl + 1)!$ nodes, we have

$$\log_2 N = \log_2 k! = k \log_2 k - O(k),$$

where we have used Stirling's approximation [22]. Therefore,

$$\frac{\log_2 N}{\log_2 \log_2 N} = \frac{k \log_2 k - O(k)}{\log_2\left(k \log_2 k - O(k)\right)} = \frac{k \log_2 k - O(k)}{\log_2 k + O\left(\log \log k\right)} = k - O\left(\frac{k}{\log k}\right).$$

Thus,

$$k = \frac{\log_2 N}{\log_2 \log_2 N} + o\left(\frac{\log N}{\log \log N}\right). \quad (1)$$

Since $l = O(k)$ for any MS(l, n) network, the diameter is at most equal to

$$\lceil 2.5k \rceil + 2l - 5 = \Theta\left(\frac{\log N}{\log \log N}\right).$$

□

Figs. 3 and 4 show the node degrees and diameters of various network topologies as a function of the network size. Degree at most equal to five seems to be sufficient for network sizes that are expected to be practical in the near future. Although the HCN [21], RCC [23], and HSN [51] topologies also have small node degree and diameter, and possess several desirable algorithmic properties, they are asymmetric and irregular, and their node degrees are in general larger than those of MS networks. CCC [41] and SCC [32] have degree equal to three, but their diameter and algorithmic properties are not as good as those of MS networks. MS networks are also competitive in terms of the cost measure, defined as the diameter times node degree [13], as shown in Fig. 5.

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 close to uniform, at least when the sources and destinations of the packets are uniformly distributed over all network nodes. The following corollary 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 algorithms described in Sections 2.1, 3.1, and 3.2 are used, the expected traffic on the network links is equally balanced within a constant factor.*

PROOF (ABBREVIATED). This can be shown by computing the probability with which each link is used. The expected

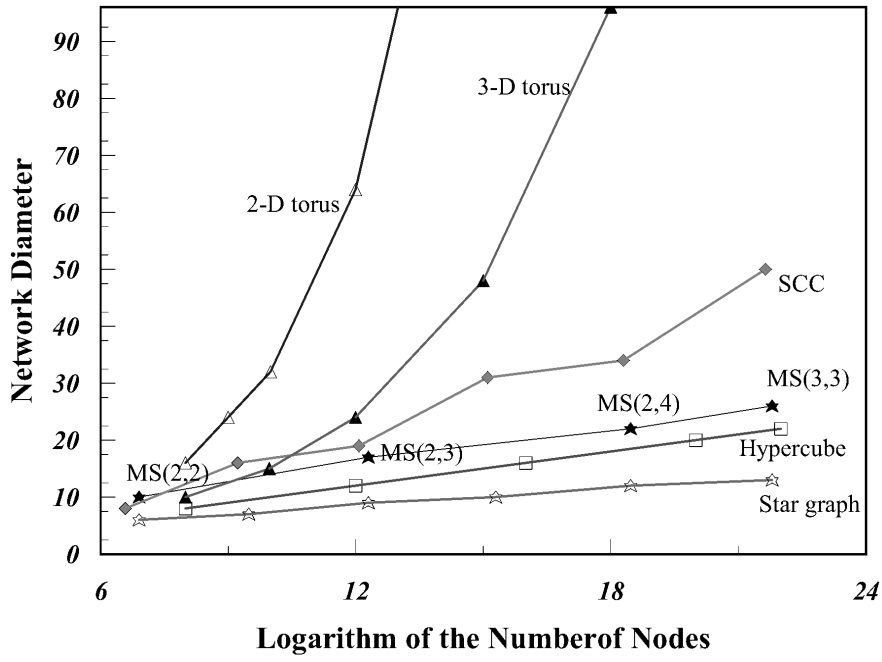


Fig. 4. Comparison of the diameters of various interconnection networks.

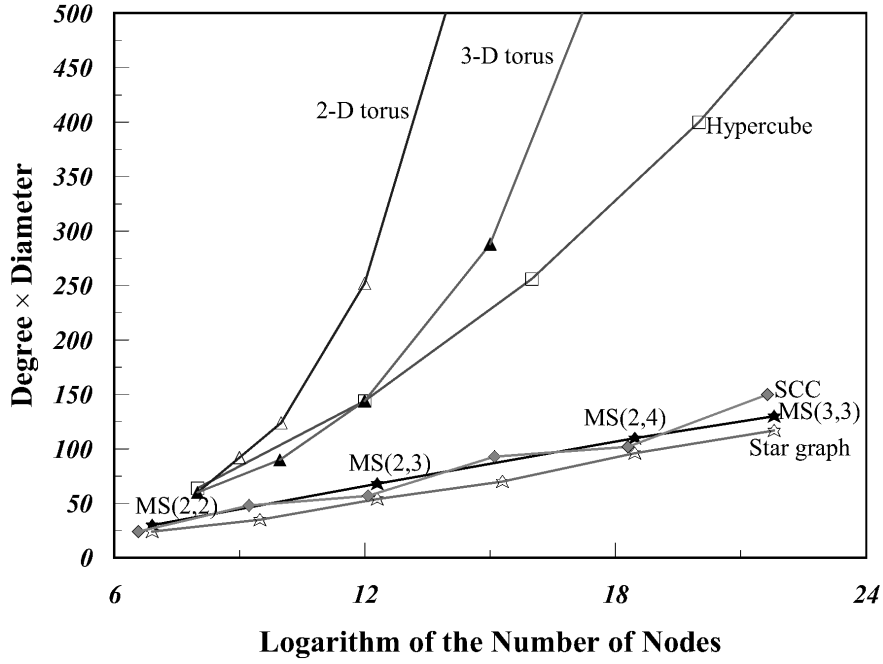


Fig. 5. Comparison of the product of degree and diameter for various interconnection networks.

traffic on all intercluster links (also, on all nucleus links) in an MS network is the same due to symmetry. The expected traffic on an intercluster 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 within a constant factor on all network links when $l = \Theta(n)$. □

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

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

Equation (1) together with (2) imply that the node degree $d = n + l - 1$ of an MS(l, n) network is minimized when $l = \Theta(n) = \Theta\left(\sqrt{\frac{\log N}{\log\log N}}\right)$, 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* $\Theta\left(\sqrt{\frac{\log N}{\log\log N}}\right)$ *if and only if* $l = \Theta(n)$.

Since $n + l - 1 \le nl = O\left(\frac{\log N}{\log\log N}\right)$ for any positive integers n and l, the node degree of an N-node MS(l, n) network can take values in the range from $\Omega\left(\sqrt{\frac{\log N}{\log\log N}}\right)$ to $O\left(\frac{\log N}{\log\log N}\right)$, 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), also results in the best diameter to lower bound ratio (Section 3.4), and optimal emulation of the star graph under both the single dimension and the all-port communication models (Section 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 $l = \Theta(n)$. In what follows, we will informally refer to an MS(l, n) network with $l = \Theta(n)$ as a *balanced MS network*.

3.4 Near Optimal Diameter

Let G be an undirected interconnection network that has N nodes, degree $d \ge 3$, and diameter $D(G)$. We then have

$$\begin{aligned} N &\le 1 + d\sum_{i=1}^{D(G)} (d-1)^{i-1} \\ &= \frac{d(d-1)^{D(G)} - 2}{d-2} \text{(Moore's bound)} \\ &< \frac{d(d-1)^{D(G)}}{d-2}. \end{aligned}$$

yielding the lower bound

$$D(G) > \log_{d-1} N + \log_{d-1}(1 - 2/d) \tag{3}$$

on the diameter $D(G)$. Note that the lower order term $c(d) \stackrel{\Delta}{=} \log_{d-1}(1 - 2/d)$ satisfies $c(d) \ge -\log_2 3$ for any $d \ge 3$ and approaches 0 very fast when d increases (for example, $c(4) \approx -0.631$ and $c(7) \approx -0.19$). We define the universal lower bound $D_L(N, d)$ on the diameter of a static undirected interconnection network that has N nodes and degree $d \ge 3$ as

$$D_L(N,d) \stackrel{\text{def}}{=} \log_{d-1} N + \log_{d-1}\left(1 - \frac{2}{d}\right). \tag{4}$$

For a given graph G, we define the asymptotic diameter to lower bound ratio

$$a(G) = \lim_{N\to\infty} \frac{D(G)}{D_L(N,d)}.$$

Note that $a(G) \ge 1$, and small values of $a(G)$ are desirable. A graph G will be said to have (asymptotically) optimal diameter if the ratio $a(G)$ is a constant or, equivalently, if the diameter of G is asymptotically within a constant factor from the lower bound.

THEOREM 3.5. *Any MS network has asymptotically optimal diameter.*

PROOF. By substituting the node degree $d < k = O\left(\frac{\log N}{\log\log N}\right)$ (1) into (4), we get

$$D_L(N,d) = \Omega\left(\frac{\log N}{\log\log N}\right),$$

which is of the same order of magnitude with the diameter of any N-node MS network (Theorem 3.2). □

COROLLARY 3.6. *The asymptotic diameter to lower bound ratio of the* MS(l, n) *network is* a(MS(l, n)) = 1.25, *when* $l = \Theta(n)$ *(that is, when the* MS *network is balanced).*

PROOF. Using (4) together with Stirling's approximation [22], we have

$$D_L(N,d) = \frac{\log_2 k!}{\log_2(n+l-2)} - o(1) = \frac{k\log_2 k - O(k)}{\log_2(n+l-2)}.$$

For $l = \Theta(n)$, we have $\log_2 l = \log_2 n \pm O(1)$, and the universal lower bound on the diameter becomes

$$D_L(N,d) = \frac{k(2\log_2 n \pm O(1))}{\log_2 n + O(1)} = 2k \pm O\left(\frac{k}{\log n}\right).$$

By Theorem 3.2, the diameter of the MS(l, n) network is equal to $2.5k + O(l)$, which gives an asymptotic diameter to lower bound ratio of

$$a(\text{MS}(l, n)) = 2.5/2 = 1.25,$$

when $l = \Theta(n)$. □

For $l = \Theta(n)$, the N-node MS(l, n) network has node degree

$$d = \Theta\left(\sqrt{\frac{\log N}{\log\log N}}\right)$$

and diameter

$$2.5k + O(\sqrt{k}) = \frac{2.5\log_2 N}{\log_2\log_2 N} + o\left(\frac{\log N}{\log\log N}\right)$$

(from (1)), both of which are sublogarithmic. The asymptotic diameter to lower bound ratios for the balanced MS network and several other interconnection networks of interest are summarized in Table 1.

Although diameter and average distance may be less important for networks using wormhole routing under light traffic, they are crucial for network performance under heavy load. The maximum throughput of a network is inversely proportional to these parameters for any switching technology. In [2], [17], it has been shown that lower-dimensional k-ary n-cubes perform better than higher-dimensional ones under the constraint of constant bisection bandwidth. In [1], Abraham and Padmanabhan examined network performance under pin-out constraints and showed that higher-dimensional networks performed better. Generally speaking, low-dimensional k-ary n-cubes outperform MS networks under the bisection-bandwidth constraint; while MS networks outperform k-ary n-cubes and hypercubes under constant pin-out constraint. Detailed comparisons based on such considerations are outside the scope of this paper.

4 Parallel Algorithms in MS Networks

In this section, we show how to emulate algorithms developed for a k-dimensional star graph on an MS(l, n) network. In our emulation algorithms, a node in the k-star is one-to-one

TABLE 1
ASYMPTOTIC DIAMETER TO LOWER BOUND RATIOS OF VARIOUS INTERCONNECTION NETWORKS WITH N NODES

Graph	Degree	Diameter	$a(G)^{\dagger}$
Hypercube	$\log_2 N$	$\log_2 N$	$\log_2 \log_2 N$
Star Graph	$\frac{\log_2 N}{\log_2 \log_2 N} + o\left(\frac{\log N}{\log \log N}\right)$	$\frac{1.5 \log_2 N}{\log_2 \log_2 N} + o\left(\frac{\log N}{\log \log N}\right)$	1.5
d-dimensional mesh	$2d$	$\Theta(N^{1/d})$	$\Theta(\frac{N^{1/d} \log d}{\log N})$
de Bruijn Graph‡	4	$\log_2 N$	1.73
CCC	3	$2.5 \log_2 N - O(\log \log N)$	2.5
SCC	3	$\Theta\left(\frac{\log^2 N}{(\log \log N)^2}\right)$	$\Theta\left(\frac{\log N}{(\log \log N)^2}\right)$
balanced MS(l,n)	$2\sqrt{\frac{\log_2 N}{\log_2 \log_2 N}} + o\left(\sqrt{\frac{\log N}{\log \log N}}\right)$	$\frac{2.5 \log_2 N}{\log_2 \log_2 N} + o\left(\frac{\log N}{\log \log N}\right)$	1.25

†*When N is not large, the actual diameter to lower bound ratio $D(G) / D_L(N, d)$ is larger than the asymptotic ratio $a(G)$ indicated in the table for star graphs and MS networks, and smaller than that for CCC networks.*

‡*The binary de Bruijn graph is viewed as an undirected degree-4 network.*

mapped on the node that has the same permutation label in the MS(l, n) network. We also present constant-dilation embeddings of several important topologies on MS networks.

4.1 Parallel Algorithms under the Single-Dimension Communication Model

In this section, 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. Many algorithms developed for the star graph fall into this category [38].

Two basic communication tasks that arise often in applications are the multinode broadcast (MNB) and the total exchange (TE) [12], [27], [48], [49]. 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. Misic and Jovanovic [38] have proposed strictly optimal algorithms to execute both tasks in time $k! - 1$ and $(k + 1)! + o((k + 1)!)$,[1] respectively, in a k-star with single-dimension communication. Using Theorem 3.1, the algorithms proposed in [38] give rise to corresponding asymptotically optimal algorithms for the MS(l, n) network.

COROLLARY 4.1. *The total exchange task can be performed in time $\Theta\left(\frac{N \log N}{\log \log N}\right)$ 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 3.1 through emulating the TE algorithm given in [38]. The lower bound can be proved by arguing that any interconnection network with N nodes of degree $O(\log^c N)$ has mean internodal distance of at least

$$\Omega\left(\frac{\log N}{\log \log N}\right).$$

(The derivation is similar to that given in Section 3.4 for the lower bound on the diameter.) Therefore, the total number of packet transmissions required to execute the TE task is $\Omega\left(\frac{N^2 \log N}{\log \log N}\right)$. 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, often with a slowdown factor of 1 asymptotically. We present this result formally in the following theorem.

THEOREM 4.2. *Any algorithm in a k-star using links of consecutive dimensions $j, j + 1, \ldots, j + s - 1$ in s consecutive steps can be emulated on the* MS(l, n) *network in time $s - 2a_1 + 2b_1 + 2c_1$, where $a_1 = \lfloor (j - 2)/n \rfloor$, $b_1 = \lfloor (j + s - 2)/n \rfloor$, and*

$$c_1 = \begin{cases} 1 & \text{if } a_1 > 0; \\ 0 & \text{otherwise.} \end{cases}$$

PROOF. If we emulate the transmissions over two successive dimensions $j + t$ and $j + t + 1$ using the algorithm given in Theorem 3.1, each node U has to route packets via links (from left to right)

$$S_{t_1+1}, T_{t_0+2}, S_{t_1+1}, S_{t_1+1}, T_{t_0+3}, S_{t_1+1},$$

where $t_0 = (j + t - 2) \bmod n$ and $t_1 = \lfloor (j + t - 2)/n \rfloor$, assuming $t_0 \neq n - 1$. Clearly, the third and fourth routing steps cancel each other and can be removed, and the emulation requires transmissions over links

$$S_{t_1+1}, T_{t_0+2}, T_{t_0+3}, S_{t_1+1}.$$

Note that each node $S_{t_1+1}(U)$, which receives data from node U in the first step, has to emulate the required computation on node U.

When $t_0 = n - 1$, different intercluster links will be used in the third and fourth routing steps above, for a total of exactly $2b_1 - 2a_1$ steps that cannot be eliminated by the above method. Two steps (routing on S_{a_1} links) are required at the beginning and end of the

1. The notation $f(N) = o(g(N))$ means that $\lim_{N \to \infty} f(N)/g(N) = 0$.

emulation algorithm when $a_1 \neq 0$. Thus, $2b_1 - 2a_1 + 2c_1$ extra steps are required to emulate the s steps of the algorithm on the $(nl + 1)$-star, which completes the proof. □

Using Theorem 4.2, we can emulate a number of star graph algorithms on an MS network with a slowdown factor smaller than three. In particular, we can obtain an asymptotically optimal algorithm (within a factor of one) to execute the multinode broadcast task in an MS network by emulating the corresponding algorithm given in [38] for star graphs, leading to the following corollary.

COROLLARY 4.3. *The multinode broadcast task can be performed in $N + o(N)$ time in an N-node* MS *network under the SDC model.*

4.2 Emulation of Star Graphs under the All-Port Communication Model

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. Given two graphs G_1 and G_2 of similar sizes, and node degrees d_1 and d_2, a lower bound on the time required for G_1 to emulate G_2 is $T(d_1, d_2) = \lceil d_2/d_1 \rceil$. When G_1 can emulate G_2 with a slowdown factor of $\Theta(T(d_1, d_2))$, we will say that graph G_1 can (asymptotically) optimally emulate graph G_2. In what follows, we show that an MS(l, n) network can emulate a star graph of the same size with asymptotically optimal slowdown.

THEOREM 4.4. *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 Theorem 3.1, we have shown that an MS(l, n) network can emulate an $(nl + 1)$-star with a slowdown factor of three 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. In particular, a packet for a dimension-j neighbor, $j \geq n + 2$, in the emulated star graph will be sent through links $S_{j_1+1}, T_{j_0+2}, S_{j_1+1}$, where $j_0 = j - 2 \bmod n$ and $j_1 = \lfloor (j - 2)/n \rfloor$. There exist several schedules that guarantee the desired slowdown factor. In what follows, we present such a possible schedule.

We first consider the special case where $l = rn + 1$ for some positive integer r.

- At time 1, each node sends the packets for its dimension-j neighbors (in the emulated k-star), $j = 2, 3, 4, \ldots, n + 1$, through links T_j.
- At time t, $t = 1, 2, 3, \ldots, n$, each node sends the packets for its dimension-$u_i(t)$ neighbors, $i = 2, 3, 4, \ldots, l$, through links S_i, where $u_i(t) = (i - 1)n + 2 + (i + t - 3 \bmod n)$.
- At time t, $t = sn + 2, sn + 3, sn + 4, \ldots, (s + 1)n + 1$ for $s = 0, 1, 2, \ldots, r - 1$, each node forwards the packets for dimension-$v_i(t)$ neighbors, $i = sn + 2, sn + 3, sn + 4, \ldots, (s + 1)n + 1$, through links $T_{v_i(t)-(i-1)n}$, where $v_i(t) = (i - 1)n + 2 + (i + t - 4 \bmod n)$.
- At time t, $t = n + 1, n + 2, \ldots, 2n$, each node forwards the packets for its dimension-$u_i(t)$ neighbors, $i = 2, 3, 4, \ldots, n + 1$, through links S_i, where $u_i(t) = (i - 1)n + 2 + (i + t - 3 \bmod n)$.
- At time t, $t = sn + 3, sn + 4, sn + 5, \ldots, (s + 1)n + 2$ for $s = 1, 2, 3, \ldots, r - 1$, each node forwards the packets for dimension-$u_i(t)$ neighbors, $i = sn + 2, sn + 3, sn + 4, \ldots, (s + 1)n + 1$, through links S_i, where $u_i(t) = (i - 1)n + 2 + (i + t - 5 \bmod n)$.

Fig. 6a shows such a schedule for emulating a 13-star on an MS(4, 3) network.

In what follows we extend the previous schedule to the general case where l is not of the form $l = rn + 1$. The schedule for $l \leq n$ can be easily obtained by removing the unused part of the schedule for an MS$(n + 1, n)$ network. Other possible cases can be formulated by assuming that $l = rn - w$ for some integers $r \geq 2$ and $0 \leq w \leq n - 2$, in which case we can modify the schedule as follows. We initially start with the schedule for an MS$(rn + 1, n)$ network. Clearly, the transmissions in the schedule that correspond to the emulation of dimensions $j > ln + 1$ are not used by the MS(l, n) network. Therefore, we can now perform each of the transmissions over links T_{j_0+2} originally scheduled for time $l + 1$ through $rn + 1$ at time earlier than $l + 1$ by rescheduling these transmissions to the unused part of the schedule. Note that the modified part of the schedule are for the emulation of some dimensions larger than $(r - 1)n^2 + n + 1$ (that is, some of the dimensions that correspond to the last $l - (r - 1)n - 1 = n - w - 1$ blocks). We then swap generators T_{j_0+2} in the modified part of the schedule with part of the schedule for the emulation of dimensions smaller than $(r - 1)n^2 + n + 2$ (that is, for some of the dimensions that correspond to the first $(r - 1)n + 1$ blocks). Due to the previous modifications, we also have to move the schedule for some generators S_{j_1+1}. In particular, we will move the final generator S_{j_1+1} in each of the three-step single-dimension emulations one time step after the use of T_{j_0+2} generators when possible. When $l + 1 < 2n$, the schedule for some generators S_{j_1+1} cannot be moved before time $2n$. As a result, the time required for emulation under the all-port communication model is equal to $l + 1$ if $l + 1 \geq 2n$, and is equal to $2n$ otherwise. Fig. 6b shows such a schedule for emulating a 16-star on an MS(5, 3) network. □

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 4.5. *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* $\Theta\left(\sqrt{\frac{\log N}{\log \log N}}\right)$*).*

generators of an MS(4,3) network	dimension j of the 13-star being emulated												
	2	3	4	5	6	7	8	9	10	11	12	13	
Step 1	T_2	T_3	T_4	S_2	—	—	—	S_3	—	—	—	S_4	
Step 2				T_2	S_2	—	—	T_3	S_3	S_4	—	T_4	
Step 3				—	T_3	S_2	S_3	—	T_4	T_2	S_4	—	
Step 4				S_2	—	T_4	T_2	S_3	—	—	T_3	S_4	
Step 5					S_2	—	—		S_3	S_4	—		
Step 6						S_2	S_3				S_4		

(a)

generators of an MS(5,3) network	dimension j of the 16-star being emulated															
	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	
Step 1	T_2	T_3	T_4	S_2	—	—	—	S_3	—	—	—	S_4	—	S_5	—	
Step 2				T_2	S_2	—	—	—	S_3	S_4	—	T_4	—	T_3	S_5	
Step 3				—	T_3	S_2	S_3	—	T_4	T_2	S_4	—	S_5	—	—	
Step 4				S_2	—	—	T_2	—	S_3	—	T_3	S_4	—	S_5	T_4	
Step 5					S_2	T_4	S_3	T_3		S_4	—		T_2		S_5	
Step 6						S_2		S_3			S_4		S_5			

(b)

Fig. 6. Schedules for emulating star graphs on MS networks, under the all-port communication model. Note that a generator appears at most once in a row, and each column $j > 4$ consists of generators $S_{j_1+1}, T_{j_0+2}, S_{j_1+1}$, where $j_0 = j - 2 \bmod 3$ and $j_1 = \lfloor (j-2)/3 \rfloor$. (a) Emulating a 13-star on an MS(4, 3) network. (b) Emulating a 16-star on an MS(5, 3) network. The links in the MS network are fully used during Steps 1 to 5 and are 93 percent used on the average.

PROOF. It follows from Lemma 3.4, Theorem 4.4, and the fact that a graph of degree $\Theta\left(\sqrt{\frac{\log N}{\log\log N}}\right)$ cannot emulate a graph of degree $\Theta\left(\frac{\log N}{\log\log N}\right)$ with a slowdown smaller than $\Theta\left(\sqrt{\frac{\log N}{\log\log N}}\right)$, under the all-port communication model. □

Note that the slowdown factor of $\Theta\left(\sqrt{\frac{\log N}{\log\log N}}\right)$ 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 $\Theta\left(\sqrt{\frac{\log N}{\log\log N}}\right)$ 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 [20] have given optimal algorithms to execute the multinode broadcast (MNB) and the total exchange (TE) communication tasks in a k-star with all-port communication in time $\Theta((k-1)!) = \Theta(N \log\log N / \log N)$ and $\Theta(k!) = \Theta(N)$, respectively. Emulating their algorithms leads to the following asymptotically optimal algorithms for MS networks.

COROLLARY 4.6. *The multinode broadcast task can be performed in 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* $\Theta\left(\sqrt{\frac{\log N}{\log\log N}}\right)$, *under the all-port communication model.*

COROLLARY 4.7. *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* $\Theta\left(\sqrt{\frac{\log N}{\log\log N}}\right)$, *under the all-port communication model.*

PROOF. Since the TE can be performed in an N-node star graph in time $\Theta(N)$ [20], it can be completed in time $O\left(N\sqrt{\frac{\log N}{\log\log N}}\right)$ in an MS(l, n) network with N nodes of degree $\Theta\left(\sqrt{\frac{\log N}{\log\log N}}\right)$ through emulation (Theorem 4.4), assuming all-port communication and $l = \Theta(n)$. By arguing, as in the derivation of the universal diameter lower bound $D_L(d, N)$, we can show that the mean internodal distance of an N-node graph with degree $\Theta\left(\sqrt{\frac{\log N}{\log\log N}}\right)$ is at least $\Omega\left(\frac{\log N}{\log\log N}\right)$. The total number of packets that have to be exchanged to perform a TE is $N^2 - N$, for a total of $\Omega\left(\frac{N^2 \log N}{\log\log N}\right)$ packet transmissions. Since at most $O\left(N\sqrt{\frac{\log N}{\log\log N}}\right)$ transmissions can take place simultaneously in an N-node interconnection network of degree $\Theta\left(\sqrt{\frac{\log N}{\log\log N}}\right)$ under the all-port communication model, the time required to complete the TE is at least

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

□

4.3 Embeddings of Trees, Meshes, Hypercubes, and Complete Transposition Graphs

In this section, we present constant-dilation embeddings of several important graphs in MS networks. The following corollary follows directly from Theorem 3.1.

COROLLARY 4.8. *A k-star graph can be one-to-one embedded in an* MS(l, n) *network with load 1, expansion 1, and dilation 3.*

A k-dimensional complete transposition graph CT(k) [34], [35] is a Cayley graph defined with a generator set consisting of all the generators that interchange any two of

the k symbols in the label of a node. A CT(k) graph has $k!$ nodes, degree $k(k-1)/2$, and diameter $k-1$. It contains a k-star or a k-dimensional bubble-sort network [5] as a subgraph and has been shown to be a rich topology that can efficiently embed many other popular topologies, including hypercubes, meshes, and trees. The following theorem provides $O(1)$-dilation embedding of complete transposition graphs in macro-star networks.

THEOREM 4.9. *A k-dimensional complete transposition graph can be one-to-one embedded in an* MS(l, n) *network with load 1, expansion 1, and dilation 5 when $l = 2$, or dilation 7 when $l \geq 3$.*

PROOF. Similar to Theorem 3.1, we map each node in the CT(k) graph onto the node with the same label in the MS(l, n) network. Therefore, the load and expansion of the embedding are both equal to 1. We let $T_{i,j}$ be the generator that interchanges the ith and jth symbols in the label of a node, where $1 \leq i < j$. Then, the generator set for a CT(k) graph consists of generators $T_{i,j}$ for any combination of integers i, j satisfying $1 \leq i < j \leq k$. Letting $i_0 = i - 2 \bmod n$, $i_1 = \lfloor (i-2)/n \rfloor$, $j_0 = j - 2 \bmod n$, and $j_1 = \lfloor (j-2)/n \rfloor$, it is easy to verify the following equivalence

$$T_{i,j} = \begin{cases} T_j \\ S_{j_1+1} T_{j_0+2} S_{j_1+1} \\ T_i T_j T_i \\ T_i S_{j_1+1} T_{j_0+2} S_{j_1+1} T_i \\ S_{i_1+1} T_{i_0+2} T_{j_0+2} T_{i_0+2} S_{i_1+1} \\ S_{i_1+1} T_{i_0+2} S_{j_1+1} T_{j_0+2} S_{j_1+1} T_{i_0+2} S_{i_1+1} \end{cases} \text{ when } \begin{cases} i = 1, j_1 = 0; \\ i = 1, j_1 > 0; \\ i_1 = j_1 = 0; \\ i_1 = 0, j_1 > 0; \\ i_1 = j_1 > 0; \\ i_1 \neq j_1, i_1, j_1 > 0. \end{cases}$$

As a result, the dilation for embedding a CT(k) graph in an MS(l, n) network is at most equal to seven. When $l = 2$, only the first five cases are possible so that the dilation is equal to five for an MS(2, n) network. □

Since a k-dimensional bubble-sort network is a subgraph of a CT(k) graph, it can also be embedded in an MS(l, n) network with dilation 5 when $l = 2$, and dilation 7 when $l \geq 3$.

A variety of embedding results are available for star graphs, bubble-sort graphs, and complete transposition graphs [14], [28], [34], [37]. These results, when combined with Theorem 4.9 and Corollary 4.8, give rise to a variety of $O(1)$-dilation embeddings for MS networks. The following corollaries summarize some of the results.

COROLLARY 4.10. *There exists a dilation-3 embedding of the complete binary tree of height 5 into an* MS(2, 2) *network. For $k \geq 7$, there exists a dilation-3 embedding of the complete binary tree of height at least equal to* $(1/2 + o(1))k \log_2 k$ *into an* MS(l, n) *network.*

PROOF. In [14], it has been shown that, for $k = 5$ or 6, there exists a dilation-1 embedding of the complete binary tree of height $2k-5$ into the k-star. For $k \geq 7$, there exists a dilation-1 embedding of the complete binary tree of height at least equal to $(1/2 + o(1))k \log_2 k$ into the k-star. The rest of the proof follows from Corollary 4.8. □

COROLLARY 4.11. *There exists a dilation-$O(1)$ embedding of the d-dimensional hypercube into an* MS(l, n), *provided*

$$d \leq k \log_2 k - \frac{3k}{2} + o(k).$$

PROOF. In [37], it has been shown that there exists a dilation-$O(1)$ embedding of the d-dimensional hypercube into a k-star, provided that $d \leq k \log_2 k - (3/2 + o(1))k$. This, combined with Corollary 4.8, completes the proof. □

COROLLARY 4.12. *There exists a load-1, expansion-1, and dilation-5 embedding of the $M_1 \times M_2$ mesh into an* MS(2, n) *network, where $M_1 \times M_2 = (2n+1)!$. There exists a dilation-7 and expansion-1 embedding of the $M_1 \times M_2$ mesh into an* MS(l, n) *network, where $M_1 \times M_2 = k!$ and $l \geq 3$.*

PROOF. It follows from Theorem 4.9 and the fact that there exists a dilation-1 expansion-1 embedding of $M_1 \times M_2$ mesh into a CT(k) graph, where $M_1 \times M_2 = k!$ [34]. □

COROLLARY 4.13. *There exists a dilation-$O(1)$ expansion-1 embedding of the $2 \times 3 \times 4 \times \cdots \times (k-1) \times k$ mesh into an* MS(l, n) *network.*

PROOF. In [28], it has been shown that there exists a dilation-3 expansion-1 embedding of the $2 \times 3 \times 4 \times \cdots \times (k-1) \times k$ mesh into a k-star. This, combined with Corollary 4.8, completes the proof. □

5 IMPLEMENTATION CONSIDERATIONS

In previous sections, we have shown various theoretical advantages of MS networks. In this section, we address some practical and implementation issues, and provide a detailed comparison between MS networks and star graphs.

5.1 Scaling Up MS Networks

An MS(l, n) network has $N = (nl+1)!$ nodes and degree equal to $n + l - 1$. For a given $N = k!$, there may be more than one MS(l, n) network with N processors, because there may be more than one pair (l, n) for which $k = nl + 1$. Two networks MS(l_1, n_2) and MS(l_2, n_2) that have the same number of nodes (i.e., $l_1 n_1 = l_2 n_2$) may have different degrees $l_1 + n_1 - 1$ and $l_2 + n_2 - 1$ and different diameters.

For a given number of nodes $N = k! = (nl+1)!$, it is generally preferable to have $l = \Theta(n)$ so that the network is balanced (see Sections 3 and 4). Clearly, this is not always possible (e.g., if $k-1$ is prime), which implies that when scaling up an MS(l, n) network with $k! = (ln+1)!$ nodes to obtain an MS(l', n') network with $(k+1)! = (l'n'+1)!$ nodes, many of the original properties may not be preserved. For example, consider scaling up an MS(3, 4) network that has 13! nodes to the (only) MS(13, 1) network that has 14! nodes (the immediately next possible number of nodes). The MS(3, 4) network is close to being balanced and has degree equal to 6, while the MS(13, 1) network has degree equal to 13. Furthermore, the MS(13, 1) network does not contain the MS(3, 4) network as a subgraph. If modularity in the design is to be preserved, the MS network should be scaled up by increasing the number of levels l by one, keeping parameter n constant.

To obtain an MS($l+1$, n) network from an MS(l, n) network, n additional symbols have to be used in the node

label, increasing the number of nodes by a factor of $\frac{(nl+n+1)!}{(nl+1)!}$. Such a step size may be too large for practical applications. This problem is common to other hierarchical modular networks, such as hypernets [25], RCC [23], and HSN [51]. In what follows, we briefly present alternative ways to scale up an MS network with a smaller step size, while preserving most of its desirable properties, and ensuring modularity in the design.

The first method is to allow more flexible connectivity at the top level. For example, consider a network that consists of $(k + m)!/k!$, $1 \le m \le n$, identical copies of an MS(l, n) network. A node in this variant network is represented by $k + m$ symbols, where $1 \le m \le n$, and is connected through an additional link to a new neighbor, obtained by swapping the first m symbols with the last (additional) m symbols of the node label. We call this an incomplete MS network and we denote it by IMS((m), $l + 1$, n). For example, the IMS((1), 3, 3) network has 8! nodes and generators

$$T_2, T_3, T_4, S_{3,2}(I) \stackrel{\text{def}}{=} 1\ 567\ 234\ 8$$

and

$$S_{(1),3,3}(I) \stackrel{\text{def}}{=} 1\ 8\ 34\ 567\ 2$$

IMS((m), $l + 1$, n) networks have properties similar to those of MS($l + 1$, n) networks, and most of the results derived in this paper can be applied to them either directly (for example, the emulation algorithms presented in Section 4) or after minor modifications (for example, the routing algorithms). In particular, to extend the MS routing algorithm introduced in Section 3 to IMS networks, we have to perform a complete SDC emulation (involving three steps) when an exchange with a symbol within the rightmost block is involved, while the rest of the algorithm remains the same. An upper bound on the diameter of IMS((m), l, n) networks can be shown to be $2.5nl + O(n + l)$. The technique used above can be further generalized to provide more flexibility at lower levels. For example, we can use generators T_2, T_3, T_4, $S_{(2,2),3,2}(I) = 1\ 56\ 4\ 23\ 78$, and $S_{(2,2),3,3}(I) = 1\ 78\ 4\ 56\ 23$ to obtain an incomplete MS network, IMS((2, 2), 3, 3), that has 8! nodes and degree 5. The integer numbers in the inner parenthesis indicate that the incomplete blocks 2 and 3 consist of two symbols each, rather than three symbols. It can also be seen that a Cayley graph [5] using generators T_2, T_3, T_4, $S_{(1),3,2}(I) = 1\ 567\ 234\ 8$, and $S'_3(I) = 1\ 678\ 5\ 234$, as well as other variant topologies, also have algorithms and properties similar to those of MS networks. Since the variants introduced so far belong to the class of Cayley graphs, they are all vertex-symmetric and regular. Similar to Theorem 3.1, it can be easily shown that all of them can embed a star graph of the same size with dilation 3. Similar to Theorem 4.9, it can be shown that all the variants introduced so far can embed a complete transposition graph of the same size with dilation 5 when the number of hierarchy levels is equal to two, and with dilation 7 otherwise. As a result, the parallel algorithms presented in Sections 4.1 and 4.3 can be directly applied to all these variants. Also, the results of Section 4.2 can be extended to them by performing SDC emulations for all dimensions with proper scheduling.

A second method for scaling up an MS network with an even smaller step size uses a strategy similar to that used in the construction of the *clustered star* and *incomplete star* [33]. More precisely, some of the level-l clusters and the corresponding links are removed from the MS(l, n) network, leaving a total of $c \in \left\{2, 3, \ldots, \frac{(nl+1)!}{(nl-n+1)!}\right\}$ level-l clusters that are not removed. We call the resultant topology a *clustered MS network*, and we denote it by CMS((c), l, n). Clearly, all the algorithms presented in this paper can be applied to each of the c level-l clusters in the CMS((c), l, n) network. The number of nodes in a CMS network can be increased by a factor of $1 + \frac{1}{c}$ by adding an additional level-l cluster. This construction can be further generalized by allowing clusters at lower levels to be removed. Also, the techniques used in the construction of the IMS and the CMS networks can also be combined to obtain even more flexible variant topologies. More precisely, an MS(l, n) network can first be scaled up to obtain an IMS((m), $l + 1$, n) network and, then, some of its top-level clusters can be removed, leaving a total of $c \in \left\{2, 3, \ldots, \frac{(nl+m+1)!}{(nl+1)!}\right\}$ top-level clusters that are not removed. In general, the variants obtained by the second method are not Cayley graphs, and they are not symmetric or regular.

Techniques similar to those used in the derivation of Cayley coset graphs [24] can also be used to obtain other variant topologies that have a small step size.

5.2 Mapping of MS Networks onto Parallel Architectures

With the rapid advances in VLSI technologies, the number of transistors and the number of processors that can be put onto a chip are expected to grow significantly. Since the processor-memory bandwidth is one of the major bottlenecks limiting the performance of current and future parallel systems, implementing processors in memory (PIM) [30], [15] or computing in RAM [46], [47] offer a lot of promise for the construction of future parallel computers. Another trend in the synthesis of multicomputers is to use off-the-shelf PC or workstation boards (or processor chips) as building modules. In either case, the number of available off-module (e.g., off-chip or off-board) pins is one of the major constraints limiting the number of processors that can be put on the module. Also, the intermodule bandwidth is a potential bottleneck on the performance of the resultant system. In what follows, we consider the case where several nodes (processors, routers, and their memory banks) of an MS network are implemented on a single chip, or more generally, a single module. EXECUBE [30], [47], hypernets [25], [26], and hierarchical shuffle-exchange networks (HSE) [16] are some of the architectures and networks that use such an approach or similar assumptions.

A natural partition of the processors (and their memory banks and network interfaces) of an MS(l, n) network into chips is to put all processors belonging to the same nucleus (n + 1)-star onto the same chip. The MS(l, n) network can then be built with identical chips, which are connected to other chips through links. This partition also eliminates the "number of parts" problem since it uses identical building modules. We expect that 3-star and 4-star (or at most 5-star)

TABLE 2
COMPARISON OF MS AND IMS NETWORKS AND STAR GRAPHS IN TERMS OF THE NODE DEGREE AND THE NUMBER OF OFF-MODULE LINKS PER NODE

# Nodes	Network	Degree	Basic Module	Number of off-module links per node
7!	7-star	6	3/4-star	4/3
8!	8-star	7	4/5-star	4/3
9!	9-star	8	4/5-star	5/4
10!	10-star	9	4-star	6
7!	MS(3,2) / MS(2,3)	4	3/4-star	2/1
8!	IMS((1),3,3) / IMS((2),2,4)	5	4/5-star	2/1
9!	IMS((2),3,3) / MS(2,4)	5	4/5-star	2/1
10!	MS(3,3)	5	4-star	2

modules will be sufficient to build MS-based multicomputers for most practical purposes (see Table 2). For example, an MS network with $7! \approx 5K$ nodes can be implemented as an MS(3, 2) built from identical 3-star chips, with each node in the chip requiring two off-chip links, or an MS(2, 3) network built from identical four-star chips, with each node in the chip requiring only one off-chip link. For comparison, a 7-star built with 3-star or 4-star chips requires four or three off-chip links per node, respectively. Since many well-known topologies use rings as their basic building modules (e.g., the CCC, SCC, ring of rings, hierarchical ring, and Cayley-graph-connected cycles [39]) or contain rings as subgraphs (e.g., meshes, hypercubes, and tori), it is quite possible that off-the-shelf ring modules (having a limited number of off-module links) will be commercially available in the future. This would make MS networks built from such modules considerably less expensive than parallel computers built from customer-designed modules. Since star graphs are viewed as attractive candidates for future parallel computers, it is also possible that modules used to build small or medium-scale star-graph will be available and could be used to build larger-scale MS networks. For example, multicomputers based on the 6-star or larger star graphs may be built from 4-star modules, with each node having at least two off-module links; these 4-star modules can also be used to build MS(3, 3) network with 10!-node or IMS networks with 7!, 8!, or 9! nodes. Moreover, the 7!, 8!, or 9!-node IMS networks can later be expanded to larger networks using the same basic modules. For comparison, if we want to build a k-star with $k = 8$, 9, or 10, using 4-star chips, the required numbers of off-chip links per node will be 4, 5, or 6, respectively, resulting in larger hardware costs and may not be commercially available. In general, k-stars with $k = 6$, 7, 8, 9 cannot be expanded to larger star graphs in the future using the same modules because their node degrees increase with network size.

Table 2 summarizes the above discussion by comparing several options for building parallel computers based on the MS, IMS, and star graph topologies in terms of implementation considerations, such as node degree, basic modules used, and the number of off-module links per node.

6 CONCLUSIONS

Desirable properties in interconnection networks for parallel systems include small network diameter, symmetry, modularity, ease of mapping efficient algorithms onto them, and reasonable implementation cost. The hypercube and star graph meet most of these requirements, but their node degrees are prohibitively large for networks of large size. The MS networks proposed in this paper form a new class of interconnection networks for the modular construction of parallel computers. MS networks have several desirable algorithmic and topological properties, while using nodes of small degree. We showed that MS networks have asymptotically optimal diameter, and we presented efficient algorithms 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, under both the single-port and the all-port communication models. In all routing algorithms presented, the expected traffic was shown to be balanced on all links of the MS(l, n) network with $l = \Theta(n)$. We presented variants of the MS network that are more flexible to scale up and can be more easily adjusted to the size of a particular application. We also compared MS networks and star graphs with respect to several practical implementation considerations. 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 medium to large-scale parallel architectures.

ACKNOWLEDGMENTS

We would like to thank the anonymous referees for their helpful comments and suggestions that led to an improved presentation of this work. This research was supported in part by the U.S. National Science Foundations under grant NSF-RIA-08930554.

REFERENCES

[1] S. Abraham and K. Padmanabhan, "Performance of Multicomputer Networks Under Pin-Out Constraints," *J. Parallel and Distributed Computing*, vol. 12, no. 3, pp. 237-248, July 1991.

[2] A. Agarwal, "Limits on Interconnection Network Performance," *IEEE Trans. Parallel and Distributed Systems*, vol. 2, no. 4, pp. 398-412, Oct. 1991.

[3] S.B. Akers, 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.

[4] S.B. Akers and B. Krishnamurthy, "The Fault Tolerance of Star Graphs," *Proc. Int'l Conf. Supercomputing*, vol. III, pp. 270-276, 1987.

[5] S.B. Akers and B. Krishnamurthy, "A Group-Theoretic Model for Symmetric Interconnection Networks," *IEEE Trans. Computers*, vol. 38, no. 4, pp. 555-565, Apr. 1989.

[6] S.G. Akl, K. Qiu, and I. Stojmenovic, "Fundamental Algorithms for the Star and Pancake Interconnection Networks with Applications to Computational Geometry," *Networks*, vol. 23, pp. 215-225, July 1993.

[7] S.G. Akl and K.A. Lyons, *Parallel Computational Geometry*. Englewood Cliffs, N.J.: Prentice Hall, 1993.

[8] M.M. de Azevedo, "Star-Based Interconnection Networks and Fault-Tolerant Clock Synchronization for Large Multicomputers," PhD dissertation, Dept. of Electrical and Computer Eng., Univ. of California, Irvine, 1997.

[9] N. Bagherzadeh, M. Dowd, and S. Latifi, "Faster Column Operations in Star Graphs," Technical Report ECE-94-02-01, Dept. of Electrical and Computer Eng., Univ. of California, Irvine, 1994.

[10] N. Bagherzadeh, M. Dowd, and S. Latifi, "A Well-Behaved Enumeration of Star Graphs," *IEEE Trans. Parallel and Distributed Systems*, vol. 6, no. 5, pp. 531-535, May 1995.

[11] D. Basak and D.K. Panda, "Designing Clustered Multiprocessor Systems Under Packaging and Technological Advancements," *IEEE Trans. Parallel and Distributed Systems*, vol. 7, no. 9, pp. 962-978, Sept. 1996.

[12] D.P. Bertsekas and J. Tsitsiklis, *Parallel and Distributed Computation—Numerical Methods*. Englewood Cliffs, N.J.: Prentice Hall, 1989.

[13] L.N. Bhuyan and D.P. Agrawal, "Generalized Hypercube and Hyperbus Structures for a Computer Network," *IEEE Trans. Computers*, vol. 33, no. 4, pp. 323-333, Apr. 1984.

[14] A. Bouabdallah, M.C. Heydemann, J. Opatrny, and D. Sotteau, "Embedding Complete Binary Trees into Star Networks," *Proc. Int'l Symp. Mathematical Foundations of Computer Science*, pp. 266-275, 1994.

[15] D. Clark, *Proc. Nat'l Science Foundation Workshop Critical Issues in Computer Architecture Research*, May 1996.

[16] R. Cypher and J.L.C. Sanz, "Hierarchical Shuffle-Exchange and de Bruijn Networks," *Proc. IEEE Symp. Parallel and Distributed Processing*, pp. 491-496, 1992.

[17] W.J. Dally, "Performance Analysis of k-ary n-Cube Interconnection Networks," *IEEE Trans. Computers*, vol. 39, no. 6, pp. 775-785, June 1990.

[18] G. Della Vecchia and C. Sanges, "A Recursively Scalable Network VLSI implementation," *Future Generations Computer Systems*, pp. 235-243, 1988.

[19] D. Duh, G. Chen, and J. Fang, "Algorithms and Properties of a New Two-Level Network with Folded Hypercubes as Basic Modules," *IEEE Trans. Parallel and Distributed Systems*, vol. 6, no. 7, pp. 714-723, July 1995.

[20] P. Fragopoulou and S.G. Akl, "Optimal Communication Algorithms on Star Graphs Using Spanning Tree Constructions," *J. Parallel and Distributed Computing*, vol. 24, pp. 55-71, 1995.

[21] K. Ghose and R. Desai, "Hierarchical Cubic Networks," *IEEE Trans. Parallel and Distributed Systems*, vol. 6, no. 4, pp. 427-435, Apr. 1995.

[22] R.L. Graham, D.E. Knuth, and O. Patashnik, *Concrete Mathematics: A Foundation for Computer Science*. Menlo Park, Calif.: Addison-Wesley, 1994.

[23] M. Hamdi, "A Class of Recursive Interconnection Networks: Architectural Characteristics and Hardware Cost," *IEEE Trans. Circuits and Systems—I: Fundamental Theory and Applications*, vol. 41, no. 12, pp. 805-816, Dec. 1994.

[24] J.-P. Huang, S. Lakshmivarahan, and S.K. Dhall, "Analysis of Interconnection Networks Based on Simple Cayley Coset Graphs," *Proc. IEEE Symp. Parallel and Distributed Processing*, pp. 150-157, 1993.

[25] K. Hwang and J. Ghosh, "Hypernet: A Communication-Efficient Architecture for Constructing Massively Parallel Computers," *IEEE Trans. Computers*, vol. 36, no. 12, pp. 1,450-1,466, Dec. 1987.

[26] K. Hwang, *Parallel Processing for Supercomputers and Artificial Intelligence*. New York: McGraw-Hill, 1989.

[27] S.L. Johnsson and C.-T. Ho, "Optimum Broadcasting and Personalized Communication in Hypercubes," *IEEE Trans. Computers*, vol. 38, no. 9, pp. 1,249-1,268, Sept. 1989.

[28] J.S. Jwo, S. Lakshmivarahan, and S.K. Dhall, "Embedding of Cycles and Grids in Star Graphs," *Proc. IEEE Symp. Parallel and Distributed Processing*, pp. 540-547, 1990.

[29] D.E. Knuth, *The Art of Computer Programming*. Reading, Mass.: Addison-Wesley, 1973.

[30] P.M. Kogge, "EXECUBE—A New Architecture for Scalable MPPs," *Proc. Int'l Conf. Parallel Processing*, vol. I, pp. 77-84, 1994.

[31] S. Lakshmivarahan, 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, pp. 361-407, Apr. 1993.

[32] S. Latifi, M.M. de 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.

[33] S. Latifi and N. Bagherzadeh, "Incomplete Star: An Incrementally Scalable Network Based on the Star Graph," *IEEE Trans. Parallel and Distributed Systems*, vol. 5, no. 1, pp. 97-102, Jan. 1994.

[34] S. Latifi and P.K. Srimani, "Transposition Networks as a Class of Fault-Tolerant Robust Networks," *IEEE Trans. Parallel and Distributed Systems*, vol. 45, no. 2, pp. 230-238, Feb. 1996.

[35] F.T. Leighton, *Introduction to Parallel Algorithms and Architectures: Arrays, Trees, Hypercubes*. San Mateo, Calif.: Morgan Kaufmann, 1992.

[36] A. Menn and A.K. Somani, "An Efficient Sorting Algorithm for the Star Graph Interconnection Networks," *Proc. Int'l Conf. Parallel Processing*, vol. III, pp. 1-8, 1990.

[37] Z. Miller, D. Pritikin, and I.H. Sudborough, "Bounded Dilation Maps of Hypercubes into Cayley Graphs on the Symmetric Group," *Math. Systems Theory*, vol. 29, no. 6, pp. 551-572, Nov./Dec. 1996.

[38] J. Misic and Z. Jovanovic, "Communication Aspects of the Star Graph Interconnection Network," *IEEE Trans. Parallel and Distributed Systems*, vol. 5, no. 7, pp. 678-687, July 1994.

[39] S. Öhring, F. Sarkar, S.K. Das, and D.H. Hohndel, "Cayley Graph Connected Cycles: A New Class of Fixed-Degree Interconnection Networks," *Proc. Int'l Conf. Systems Sciences*, pp. 479-487, 1995.

[40] B. Parhami, *Introduction to Parallel Processing: Algorithms and Architectures*. Plenum, 1998.

[41] F.P. Preparata and J.E. Vuillemin, "The Cube-Connected Cycles: A Versatile Network for Parallel Computation," *Comm. ACM*, vol. 24, no. 5, pp. 300-309, May 1981.

[42] D.K. Saikia and R.K. Sen, "Order Preserving Communication on a Star Network," *Parallel Computing*, vol. 21, pp. 771-782, 1995.

[43] D.K. Saikia and R.K. Sen, "Two Ranking Schemes for Efficient Computation on the Star Interconnection Networks," *IEEE Trans. Parallel and Distributed Systems*, vol. 4, no. 4, pp. 321-327, Apr. 1996.

[44] I.D. Scherson and A.S. Youssef, *Interconnection Networks for High-Performance Parallel Computers*. Los Alamitos, Calif.: IEEE CS Press, 1994.

[45] J.-P. Sheu, C.-T. Wu, and T.-S. Chen, "An Optimal Broadcasting Algorithm without Message Redundancy in Star Graphs," *IEEE Trans. Parallel and Distributed Systems*, vol. 6, no. 6, pp. 653-658, June 1995.

[46] T. Sterling, P. Messina, and P. Smith, *Enabling Technologies for Peta(FL)OPS Computing*. MIT Press, 1994.

[47] T. Sterling and M.J. MacDonald, *Proc. PetaFLOPS Frontiers Workshop*, 1995.

[48] E.A. Varvarigos and D.P. Bertsekas, "Communication Algorithms for Isotropic Tasks in Hypercubes and Wraparound Meshes," *Parallel Computing*, vol. 18, no. 11, pp. 1,233-1,257, Nov. 1992.

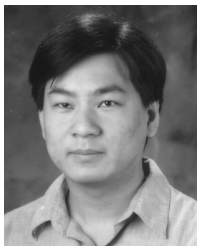
[49] E.A. Varvarigos and D.P. Bertsekas, "Multinode Broadcast in Hypercubes and Rings with Randomly Distributed Length of Packets," *IEEE Trans. Parallel and Distributed Systems*, vol. 4, no. 2, pp. 144-154, Feb. 1993.

[50] C.-H. Yeh and E.A. Varvarigos, "Macro-Star Networks: Efficient Low-Degree Alternatives to Star Graphs for Large-Scale Parallel Architectures," *Proc. Symp. Frontiers of Massively Parallel Computation*, pp. 290-297, Oct. 1996.

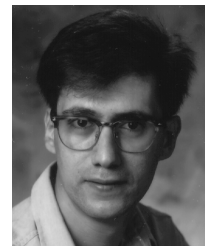
[51] C.-H. Yeh and B. Parhami, "Recursive Hierarchical Swapped Networks: Versatile Interconnection Architectures for Highly Parallel Systems," *Proc. IEEE Symp. Parallel and Distributed Processing*, pp. 453-460, Oct. 1996.

[52] C.-H. Yeh and B. Parhami, "Cyclic Networks—A Family of Versatile Fixed-Degree Interconnection Architectures," *Proc. Int'l Parallel Processing Symp.*, pp. 739-743, Apr. 1997.

[53] C.-H. Yeh, "Efficient Low-Degree Interconnection Networks for Parallel Processing: Topologies, Algorithms, VLSI Layouts, and Fault Tolerance," PhD dissertation, Dept. of Electrical and Computer Eng., Univ. of California, Santa Barbara, Mar. 1998.



Chi-Hsiang Yeh received the BS degree in electrical engineering from National Taiwan University, Taiwan, in 1992, and the MS and PhD degrees in electrical and computer engineering from the University of California, Santa Barbara, in 1996 and 1998, respectively. His research interests include parallel and distributed computation, interconnection networks, fault-tolerant computing, VLSI layouts, computer arithmetic, and threshold circuits. He has published more than 25 papers in these areas. He is a member of the IEEE and the ACM.



Emmanouel A. Varvarigos received a BS in electrical and computer engineering from the National Technical University of Athens, Greece, in 1988, and the MS, Electrical Engineer, and PhD degrees in electrical engineering and computer science from the Massachusetts Institute of Technology in 1990, 1991, and 1992, respectively. In 1990, he conducted research on optical fiber communication at Bell Communications Research, Morristown, New Jersey. He is currently an associate professor in the Department of Electrical and Computer Engineering at the University of California, Santa Barbara. His research interests are in the areas of parallel and distributed computation, high-speed data networks, and mobile communications. Dr. Varvarigos received the first panhellenic prize in the Greek Mathematic Olympiad in 1982, was awarded the Technical Chamber of Greece award four times (1984-1988), and received a U.S. National Science Foundation Research Initiation Award. He is a member of the IEEE.