

# Average Distance and Routing Algorithms in the Star-Connected Cycles Interconnection Network

M.M. Azevedo<sup>1</sup>, N. Bagherzadeh<sup>1</sup>, M. Dowd<sup>1</sup> and S. Latifi<sup>2</sup>

The star-connected cycles (SCC) graph was recently proposed as an attractive interconnection network for parallel processing, using a star graph to connect cycles of nodes. This paper presents an analytical solution for the problem of the average distance of the SCC graph. The cost of a route in the SCC graph is divided into three components and it is shown that one of such components is affected by the routing algorithm being used. Three routing algorithms for the SCC graph are discussed, which respectively employ random, greedy and minimal routing rules. The computational complexities of the algorithms and the average costs of the paths they produce, are compared. Finally, the problem of how each of the algorithms can be used in association with wormhole routing is considered.

## INTRODUCTION

An interconnection network is characterized by four distinct aspects: topology, routing, flow control and switching [1]. The topology of a network defines how the nodes are interconnected by links and is usually modeled by a graph. Routing determines the path selected by a packet to reach its destination and is usually specified by means of a routing algorithm. Flow control deals with the allocation of links and buffers to a packet as it is routed through the network. Switching determines the mechanism by which data is moved from an incoming link to an outgoing link of a node (store-and-forward, circuit switching, virtual cut-through and wormhole routing are examples of switching techniques found in parallel architectures).

This paper continues the study of topological and routing aspects of the star-connected cycles (SCC) interconnection network [2], which has been recently proposed as an attractive extension of the star graph [3]. An SCC graph is related to a star graph in the same way a cube-connected cycles graph [4] is related to a hypercube [5]. Namely, an SCC graph is formed from a star graph by replacing the nodes of the latter with cycles or rings of nodes. The SCC graph constitutes an efficient architecture for execution of parallel algorithms, which includes broadcasting [6] and FFT [7]. Mesh algorithms are also supported in SCC graphs via embeddings [8]. The SCC graph inherits many of the interesting properties of the star graph, while employing, at most, three I/O ports per node. This last aspect categorizes

- 
1. Department of Electrical and Computer Engineering, University of California, Irvine, CA 92697-2625, USA.
  2. Department of Electrical and Computer Engineering, University of Nevada, Las Vegas, NV 89154-4026, USA.

the SCC graph as a bounded-degree network (other examples are in [4,9]). Networks with bounded degree favor area-efficient VLSI layouts and scale more easily than variable-degree networks.

Previously known topological aspects of SCC graphs included degree, symmetry, diameter and fault-diameter and were derived in [10,2]. Here, the study of these aspects is continued by investigating the average distance (or average diameter) of SCC graphs. The interest in this property is twofold: 1) to obtain a metric for comparing the performance of routing algorithms and 2) to provide continued characterization for the graph theoretical aspects of SCC networks.

In the absence of other network traffic, modern switching techniques (e.g., wormhole routing [11]) achieve a communication latency which is virtually independent of the selected path length [1]. In this ideal environment, the two factors which contribute to the communication latency experienced by a packet are the start-up latency and the network latency. In a realistic environment in which congestion occurs, however, a third factor known as blocking time also contributes to the communication latency.

Regardless of the flow control and switching mechanisms being used in the network, congestion can usually be minimized if fewer links are used when routing a packet [12]. For communication-intensive parallel applications, the blocking time (and, consequently, the communication latency) is expected to grow with path length. In such cases, a routing algorithm should ideally compute paths whose average cost matches the average distance of the network.

In this paper, it is shown that routes in an SCC graph may contain up to three classes of links. These are referred to as lateral links, *MI* local links and *MB* local links and are defined later. Exact expressions for the average number of lateral links and *MI* local links between two nodes in an SCC graph and an upper bound on the average number of *MB* local links, are derived. When combined,

these expressions produce a tight upper bound on the average distance of the SCC graph.

It will be shown that the number of *MB* local links is affected by the routing algorithm being used and three different algorithms for the SCC graph will be presented, including random, greedy and minimal routing. The algorithms are compared according to criteria such as computational complexity (which affects their implementation in hardware) and average routing cost, for which figures were obtained by means of simulation. The results obtained with the minimal routing algorithm provide exact numeric solutions for the average distance of the SCC graphs. These simulations indicate that the greedy routing algorithm performs close to the minimal routing algorithm, while requiring a smaller complexity. It will be shown that the random routing algorithm presents the smallest complexity among the three algorithms described in this paper, also average and worst-case routing cost metrics for it will be provided. Finally, it is considered how the three algorithms can be implemented in combination with wormhole routing.

## BACKGROUND

### The Star Graph

An  $n$ -dimensional star graph, denoted by  $S_n$ , contains  $n!$  nodes which are labeled with the  $n!$  possible permutations of  $n$  distinct symbols. In this paper, the integers  $\{1, \dots, n\}$  are used to label the nodes of  $S_n$ . A node  $\pi = p_1 \dots p_i \dots p_n$  is connected to  $(n-1)$  distinct nodes, respectively labeled with permutations  $\pi_i = p_i \dots p_{i-1} p_1 p_{i+1} \dots p_n$ ,  $2 \leq i \leq n$  (i.e.,  $\pi_i$  is the permutation resulting from exchanging the symbols occupying the first and the  $i^{\text{th}}$  position in  $\pi$ ) [3]. Each of these  $(n-1)$  possible exchange operations is referred to as a generator of  $S_n$ . Two nodes  $\pi$  and  $\pi_i$  of  $S_n$  are connected by a link iff there is a generator  $g_i$  such that  $\pi \cdot g_i = \pi_i$ . The link connecting  $\pi$  and  $\pi_i$  is referred to as an  $i^{\text{th}}$ -dimension link and is labeled  $i$ .  $S_n$  has  $(n-1) \cdot (n!/2)$  links.  $S_n$  is a regular graph with degree  $\delta(S_n) = n-1$  and diameter  $\phi(S_n) = \lfloor 3(n-1)/2 \rfloor$ .  $S_n$  is vertex- and edge-symmetric

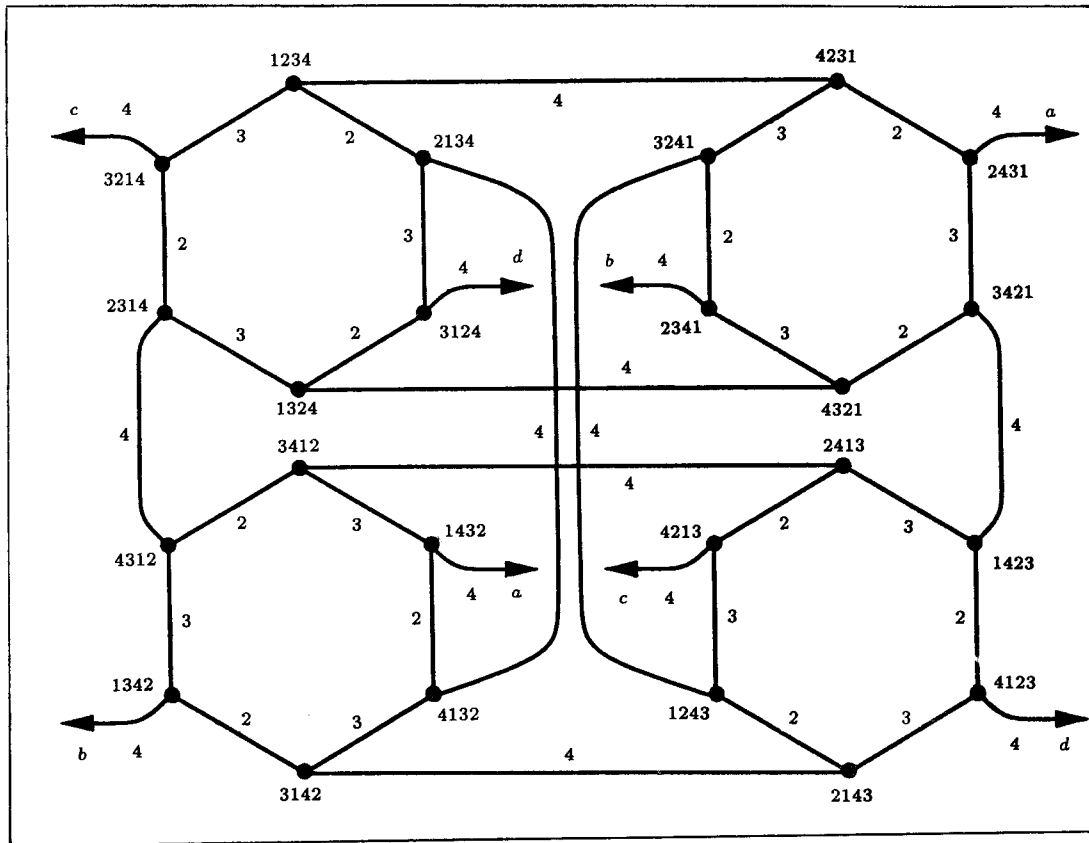


Figure 1. A 4-star graph ( $S_4$ ).

and has hierarchical structure. The degree and diameter of  $S_n$  are sublogarithmic on the size of the graph [3], which makes the star graph compare favorably with the hypercube. Figure 1 shows  $S_4$ .

**The Star-Connected Cycles (SCC) Graph**

An  $n$ -dimensional SCC graph, denoted by  $SCC_n$ , is a bounded-degree variant of  $S_n$  [2].  $SCC_n$  is formed by replacing each node of  $S_n$  with a supernode, i.e., a ring of  $(n - 1)$  nodes. The connections between nodes inside the same supernode are referred to as local links. Each supernode is connected to  $(n - 1)$  adjacent supernodes, using lateral links inherited from  $S_n$ . Figure 2 shows  $SCC_4$ .

Nodes in  $SCC_n$  are identified by a label  $\langle i, \pi \rangle$ , where  $i$  is an integer, such that  $2 \leq i \leq n$  and  $\pi$  is a permutation of  $n$  symbols. Two nodes  $\langle i, \pi \rangle$  and  $\langle i', \pi' \rangle$  are connected by a link  $(\langle i, \pi \rangle, \langle i', \pi' \rangle)$  in  $SCC_n$  iff either: 1)

$(\langle i, \pi \rangle, \langle i', \pi' \rangle)$  is a local link, i.e.,  $\pi = \pi'$  and  $\min(|i - i'|, n - 1 - |i - i'|) = 1$ , or 2)  $(\langle i, \pi \rangle, \langle i', \pi' \rangle)$  is a lateral link, i.e.,  $i = i'$  and  $\pi$  differs from  $\pi'$  only in the first and the  $i^{th}$  symbols, such that  $\pi(1) = \pi'(i)$  and  $\pi(i) = \pi'(1)$ .

For similarity with  $S_n$ , the label of the supernode containing nodes  $\langle 2, \pi \rangle, \dots, \langle n, \pi \rangle$  is  $\pi$ . Also, the lateral link connected to node  $\langle i, \pi \rangle$  is labeled  $i$ . For simplicity, supernode and lateral link labels are not shown in Figure 2.

$SCC_n$  contains  $(n - 1) \cdot n!$  nodes,  $(n - 1) \cdot n!$  local links and  $(n - 1) \cdot (n!/2)$  lateral links. Thus, the size of  $SCC_n$  is comparable to that of  $S_{n+1}$ . Local links account for 2/3 of the links of  $SCC_n$  and can be laid out very efficiently due to the ring topology of the supernodes. Moreover,  $SCC_n$  has about  $n$  times fewer lateral links than  $S_{n+1}$ , which further reduces the complexity of a VLSI layout for  $SCC_n$  when compared to  $S_{n+1}$ .  $SCC_n$  is vertex-symmetric and has degree  $\delta(SCC_n) = 2$

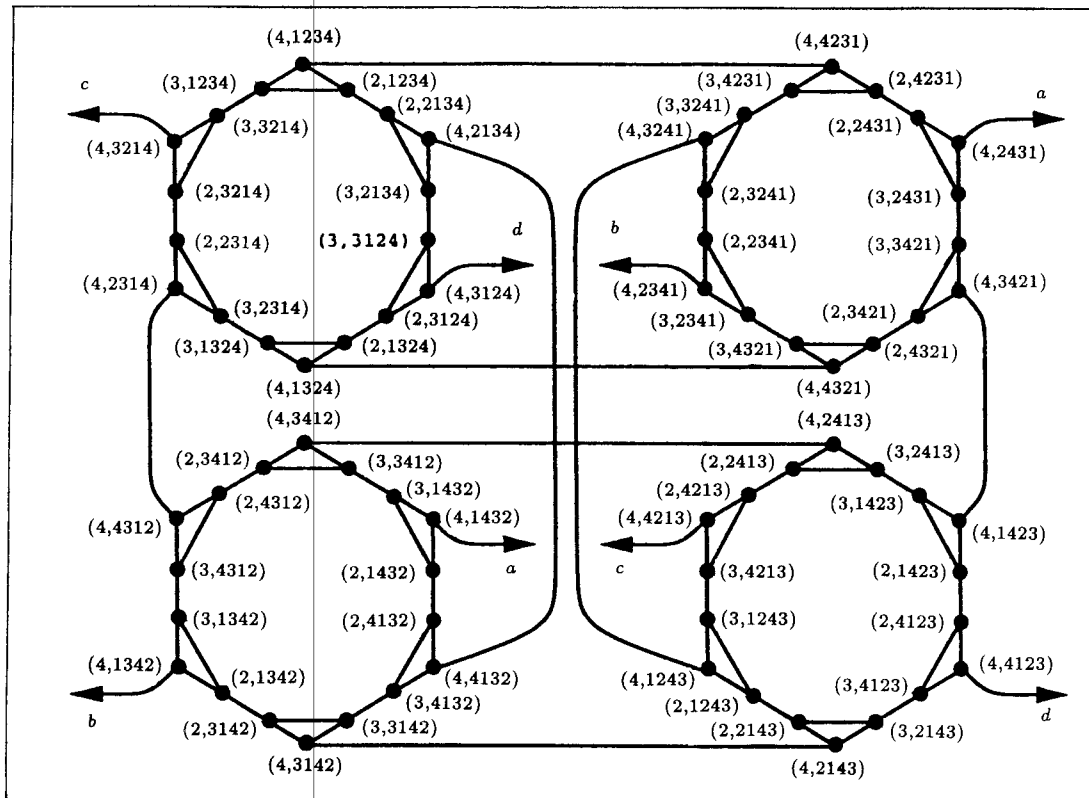


Figure 2. The  $SCC_4$  graph.

(for  $n = 3$ ) and  $\delta(SCC_n) = 3$  (for  $n \geq 4$ ). In addition, the diameter of  $SCC_n$  is given by [2]:

$$\phi(SCC_n) = \begin{cases} 6, & \text{for } n = 3 \\ \frac{1}{2}(n^2 + n - 4), & \text{for even } n \\ \frac{1}{2}(n^2 + 3n - 8), & \text{for odd } n \geq 5. \end{cases} \quad (1)$$

### AVERAGE DISTANCE OF THE SCC GRAPH

#### Preliminaries

Let the cost of a route  $P$  between node  $\langle i, \pi \rangle$  and the identity node  $\langle i_0, \pi_0 \rangle = \langle 2, 12 \dots n \rangle$  in  $SCC_n$  be  $d = lat + loc$ , where  $lat$  and  $loc$ , respectively, denote the number of lateral links and the number of local links in  $P$ . Because  $SCC_n$  is vertex-symmetric, its average distance can be computed by finding minimal cost routes to the identity from every node in the graph and averaging those over  $(n - 1) \cdot n!$ .

Before the average distance of  $SCC_n$  can be derived, some definitions related to lateral

links are needed. The symbols of permutation  $\pi$  may be organized as a set of  $r$ -cycles, i.e., cyclically ordered sets of symbols with the property that each symbol's desired position is that occupied by the next symbol in the set. These  $r$ -cycles provide a convenient means to represent permutations [13] and should not be confused with physical cycles or rings, which constitute the supernodes of  $SCC_n$ . In this paper, all  $r$ -cycles are written in canonical form [13] (i.e., the smallest symbol appears first in each  $r$ -cycle). For example, a permutation  $\pi = 265431$  can be written in cyclic format as  $(1 \ 2 \ 6)(3 \ 5)(4)$ . Note that a symbol already in its correct position appears as a 1-cycle.

Let  $C_i = (i_0 \dots i_{r-1})$  be an  $r$ -cycle in  $\pi$ ,  $2 \leq r \leq n$ . Let  $\pi \cdot R_i$  be the permutation produced from  $\pi$  by moving the symbols in  $C_i$  to their correct positions. The execution of an  $r$ -cycle  $C_i$  is, by definition, a minimal sequence of lateral links  $R_i$ , leading from supernode  $\pi$  to supernode  $\pi \cdot R_i$  (note that local links are not an issue here). Throughout the paper, the

notation of an  $r$ -cycle is distinguished from that of a sequence of lateral links by using commas in the latter.  $R_i$  can be expressed by [14,15]:

$$R_i = \begin{cases} (i_1, i_2, \dots, i_{r-1}), & \text{if } i_0 = 1 \\ (i_0, i_1, \dots, i_{r-1}, i_0), & \text{if } i_0 \neq 1. \end{cases} \quad (2)$$

In the case  $i_0 \neq 1$ ,  $C_i$  can actually be executed with  $r$  different sequences of lateral links [14,15]. Hence, for  $j : 0 \rightarrow r - 1$ , such sequences can be expressed as:

$$(i_{j \bmod r}, i_{(j+1) \bmod r}, \dots, i_{(j+r-1) \bmod r}, i_{j \bmod r}). \quad (3)$$

The minimum number of lateral links in a route from supernode  $\pi$  to  $\pi_0$  does not depend on the order chosen to execute the  $r$ -cycles in  $\pi$  and is given with [3]:

$$lat = \begin{cases} c + m, & \text{if } \pi\text{'s first symbol is } 1 \\ c + m - 2, & \text{if } \pi\text{'s first symbol is not } 1, \end{cases} \quad (4)$$

where  $c$  is the number of  $r$ -cycles of length at least 2 in  $\pi$  and  $m$  is the total number of symbols in these  $r$ -cycles.

Routes in  $SCC_n$  often consist of sequences of lateral links interleaved with local links. In what follows, some definitions that relate to local links are given.

Recall that  $loc$  denotes the contribution of the local links to the total cost of a route  $P$  from  $\langle i, \pi \rangle$  to  $\langle i_0, \pi_0 \rangle$ .  $loc$  can be further divided into two components, which are denoted by  $MI(loc)$  and  $MB(loc)$  and are defined as follows:

- $MI(loc)$  – the number of move-in ( $MI$ ) local links existing in the route from  $\langle i, \pi \rangle$  to  $\langle i_0, \pi_0 \rangle$ . By definition, these are local links that must be traversed between two lateral links belonging to the execution sequence of an  $r$ -cycle in  $\pi$ .
- $MB(loc)$  – the number of move-between ( $MB$ ) local links existing in the route from  $\langle i, \pi \rangle$  to  $\langle i_0, \pi_0 \rangle$ . By definition,  $MB$  local links are: 1) local links that must be traversed between the executions of two consecutive  $r$ -cycles in  $\pi$ , 2) local links that must

be traversed in supernode  $\pi$  and are required to move from  $\langle i, \pi \rangle$  to the lateral link that initiates the execution of the first  $r$ -cycle of  $\pi$  and 3) local links that must be traversed in supernode  $\pi_0$  and are required to move from the lateral link that finishes the execution of the last  $r$ -cycle of  $\pi$  to  $\langle i_0, \pi_0 \rangle$ .

Thus,  $d = lat + loc = lat + MI(loc) + MB(loc)$ . As an example, consider routing from  $\langle 3, 34125 \rangle$  to  $\langle 2, 12345 \rangle$  in  $SCC_5$ . The cyclic representation of permutation 34125 is  $(1\ 3)(2\ 4)(5)$ . One possible route uses the sequences of lateral links  $(2, 4, 2)$  and  $(3)$ . Figure 3 shows the  $MI$  local links and the  $MB$  local links in such a route.

Note that from the topological viewpoint there is no distinction between  $MI$  and  $MB$  local links. A particular local link used by a route in  $SCC_n$  is considered to be either an  $MI$  or an  $MB$  local link, depending on the conditions stated above. Therefore, the same local link can be classified as an  $MI$  local link for some routes and as an  $MB$  local link for others.

The cost components  $lat$ ,  $MI(loc)$  and  $MB(loc)$  exist in any route in  $SCC_n$  (although in some short routes one or more of these components may be null). Due to vertex symmetry, one can derive the average distance of  $SCC_n$  by computing the average numbers of lateral links,  $MI$  local links and  $MB$  local links in a route from  $\langle i, \pi \rangle$  to  $\langle i_0, \pi_0 \rangle$ . Such average numbers are denoted by  $\overline{lat}$ ,  $\overline{MI(loc)}$  and  $\overline{MB(loc)}$ , respectively. The average distance of  $SCC_n$ , denoted by  $\overline{\phi(SCC_n)}$ , can then be expressed by:

$$\overline{\phi(SCC_n)} = \overline{lat} + \overline{MI(loc)} + \overline{MB(loc)}. \quad (5)$$

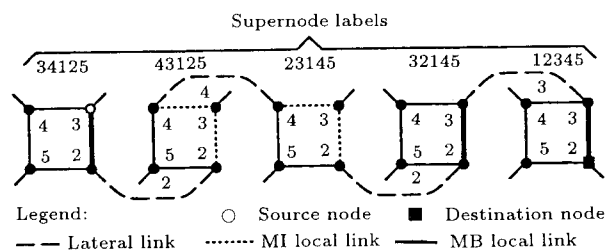


Figure 3. Types of links in a route in  $SCC_5$ .

Finally, the average number of local links existing in a route from  $\langle i, \pi \rangle$  to  $\langle i_0, \pi_0 \rangle$  in  $SCC_n$  is, by definition,  $\overline{loc} = \overline{MI(loc)} + \overline{MB(loc)}$ .

**Average Number of Lateral Links**

The number of lateral links in the route between any node of  $SCC_n$  and the identity node is exactly equal to the cost of the corresponding route in the underlying  $n$ -star graph [2]. Therefore,  $\overline{lat}$  is exactly equal to the average distance of  $S_n$ , which is given by [16]:

$$\overline{lat} = n + H_n + \frac{2}{n} - 4, \tag{6}$$

where  $H_n = \sum_{k=1}^n \frac{1}{k}$  is the  $n$ th Harmonic number [13].

**Average Number of MI Local Links**

The number of  $MI$  local links in a route in  $SCC_n$  can be calculated as follows. Consider routing from  $\langle i, \pi \rangle$  to the identity node  $\langle i_0, \pi_0 \rangle$  and let the number of  $r$ -cycles of length at least 2 in  $\pi$  be  $c$ . Let  $C_i = (i_0 \dots i_{r-1})$  be one of these  $r$ -cycles and let  $R_i$  be an execution sequence for  $C_i$  (Equation 2). Moving between two consecutive lateral links  $i_a, i_b$  in  $R_i$  requires  $d(i_a, i_b)$   $MI$  local links, where [2]:

$$d(i_a, i_b) = \min(|i_a - i_b|, n - 1 - |i_a - i_b|). \tag{7}$$

The total number of  $MI$  local links that must be traversed during the execution of  $C_i$ , denoted by  $MI(loc, C_i)$ , is, therefore, the sum of the distances  $d(i_a, i_b)$  between all pairs of consecutive lateral links  $(i_a, i_b)$  in  $R_i$ :

$$MI(loc, C_i) = \begin{cases} \sum_{j=2}^{r-1} d(i_{j-1}, i_j), & \text{if } i_0 = 1 \\ \sum_{j=1}^r d(i_{j-1}, i_{j \bmod r}), & \text{if } i_0 \neq 1. \end{cases} \tag{8}$$

**Lemma 1**

The number of  $MI$  local links that must be traversed in a route between any two nodes of  $SCC_n$  is independent of the order chosen to execute the  $r$ -cycles existing between those nodes.

*Proof*

It is first shown that  $MI(loc, C_i)$  does not depend on the sequence of lateral links  $R_i$  chosen to execute  $C_i$ . If  $i_0 = 1$ , there is only one such sequence (Equation 2). If  $i_0 \neq 1$ , there are  $r$  different possible sequences (Equation 3). However, due to the cyclic nature of these sequences, they all have the same cost  $MI(loc, C_i)$  (Equation 8). By extension, the total number of  $MI$  local links in the route,  $MI(loc)$ , must also be an invariant.  $\square$

An immediate consequence of Lemma 1 is that the number of  $MI$  local links between two nodes of  $SCC_n$  can be derived without further considerations about routing. (Assuming, of course, that routing is accomplished in adherence to Equations 2 and 3, as is the case with all routing algorithms presented in this paper.) As an example, consider an  $r$ -cycle  $C_i = (2 \ 6 \ 4)$  and let  $n = 7$ .  $C_i$  can be executed with a sequence of lateral links  $R_i = (2, 6, 4, 2)$ . The number of  $MI$  local links required in the execution of this sequence is  $MI(loc, C_i) = d(2, 6) + d(6, 4) + d(4, 2) = 2 + 2 + 2 = 6$ .

**Theorem 1**

The average number of  $MI$  local links that must be traversed in a route in  $SCC_n$  is:

$$\overline{MI(loc)} = \frac{(n - 1) \lfloor \frac{n}{2} \rfloor \lfloor \frac{n-1}{2} \rfloor}{n}. \tag{9}$$

*Proof*

The average number of local links that must be traversed between two adjacent lateral links is:

$$\overline{d(loc)} = \frac{\sum_{i=3}^n d(i, 2)}{n - 2} = \frac{\lfloor \frac{n}{2} \rfloor \lfloor \frac{n-1}{2} \rfloor}{n - 2}. \tag{10}$$

The average number of local links that must be traversed in the execution of an  $r$ -cycle  $C_i = (i_0 \dots i_{r-1})$  is:

$$\overline{MI(loc, C_i)} = \begin{cases} \overline{d(loc)} \cdot (r - 2), & \text{if } i_0 = 1 \\ \overline{d(loc)} \cdot r, & \text{if } i_0 \neq 1. \end{cases} \tag{11}$$

Over all  $n!$  possible permutations of  $n$  symbols and for each integer  $r, 2 \leq r \leq n$ ,

there is a total of  $(n-1)!$   $r$ -cycles that include symbol 1 ( $i_0 = 1$ ) and  $n!/r - (n-1)!$   $r$ -cycles that do not include symbol 1 ( $i_0 \neq 1$ ). The average number of  $MI$  local links over all  $n!$  permutations is therefore:

$$\begin{aligned} & \frac{\overline{d(loc)} \cdot \sum_{r=2}^n ((n-1)! \cdot (r-2) + (\frac{n!}{r} - (n-1)! \cdot r))}{n!} \\ &= \frac{\overline{d(loc)} \cdot (n-1)(n-2)}{n} = \frac{(n-1) \lfloor \frac{n}{2} \rfloor \lfloor \frac{n-1}{2} \rfloor}{n}. \quad \square \end{aligned}$$

### Average Number of $MB$ Local Links

Recall that  $MB$  local links are needed to move between execution sequences of adjacent  $r$ -cycles ( $2 \leq r \leq n$ ), to move into the first lateral link and to move out of the last lateral link in a route in  $SCC_n$ .

#### Theorem 2

The average number of  $MB$  local links that must be traversed in a route in  $SCC_n$ , under a random ordering of  $r$ -cycles, is:

$$\overline{MB(loc, rand)} = \left\lfloor \frac{n}{2} \right\rfloor \left\lfloor \frac{n-1}{2} \right\rfloor \left( \frac{H_n(n-1) - 2}{n^2 - 3n + 2} \right). \quad (12)$$

#### Proof

Over all  $n!$  possible permutations of  $n$  symbols and for each integer  $r$ ,  $2 \leq r \leq n$ , there is a total of  $n!/r$   $r$ -cycles. The total number of  $r$ -cycles of length at least 2 in the  $n!$  possible permutations of  $n$  symbols is, therefore,  $N_r = \sum_{r=2}^n (n!/r) = n! \cdot (H_n - 1)$ .

The average number of  $r$ -cycles,  $2 \leq r \leq n$ , in a permutation of  $n$  symbols is  $\bar{r} = N_r/n! = H_n - 1$ . The average number of  $MB$  local links that must be traversed between these  $r$ -cycles is  $\overline{MB(loc, mid)} = (\bar{r} - 1) \cdot \overline{d(loc)} = \frac{\lfloor \frac{n}{2} \rfloor \lfloor \frac{n-1}{2} \rfloor (H_n - 2)}{n-2}$ .

Let  $\langle i, \pi \rangle$  be the source node and let the first lateral link in the route be  $i_k$ ,  $2 \leq i_k \leq n$ . The average number of local links that must be traversed between  $\langle i, \pi \rangle$  and  $\langle i_k, \pi \rangle$  is  $\overline{d(in)} = \frac{1}{n-1} \sum_{i=2}^n d(i, 2) = \frac{\lfloor \frac{n}{2} \rfloor \lfloor \frac{n-1}{2} \rfloor}{n-1}$ .

Note that  $\overline{d(in)}$  differs from  $\overline{d(loc)}$  (Equation 10), since to compute  $\overline{d(in)}$  the case  $i = i_k$

must be considered. Similarly, the average number of local links that must be traversed between the last lateral link in the route and the destination node is  $\overline{d(out)} = \overline{d(in)}$ . Then, the average number of  $MB$  local links that must be traversed in a route in  $SCC_n$ , assuming a random ordering of  $r$ -cycles, is  $\overline{MB(loc, rand)} = \overline{d(in)} + \overline{MB(loc, mid)} + \overline{d(out)}$ . The theorem follows.  $\square$

As will be described in the next section, a properly designed routing algorithm can optimize the ordering of the  $r$ -cycles and reduce the average number of  $MB$  local links further below the value provided by a random ordering of  $r$ -cycles (Equation 12). The average number of  $MB$  local links, considering that the shortest route between any two nodes of an SCC graph is determined by a minimal routing algorithm, is, therefore, bounded by:

$$\overline{MB(loc)} \leq \overline{MB(loc, rand)}. \quad (13)$$

### Average Distance in the SCC Graph

#### Theorem 3

The average distance of  $SCC_n$  is bounded by:

$$\begin{aligned} \overline{\phi(SCC_n)} &\leq n + H_n + \frac{2}{n} - 4 \\ &+ \left\lfloor \frac{n}{2} \right\rfloor \left\lfloor \frac{n-1}{2} \right\rfloor \left( \frac{n^2 + 1}{n^2 - n} + \frac{H_n - 2}{n - 2} \right). \quad (14) \end{aligned}$$

#### Proof

Proof follows directly from Equations 5, 6, 9, 12 and 13.  $\square$

## ROUTING ALGORITHMS IN THE SCC GRAPH

### Ordering of $r$ -Cycles

Consider the problem of routing from  $\pi_s$  to  $\pi_d$  in  $S_n$ . Because  $S_n$  is vertex-symmetric, there is an automorphism that fixes  $\pi_d$  to the identity node  $\pi_0 = 12 \dots n$ . The same automorphism maps  $\pi_s$  by conjugation to a permutation  $\pi_{ds}$ , such that finding a path from  $\pi_s$  to  $\pi_d$  becomes equivalent to routing from  $\pi_{ds}$  to  $\pi_0$ . The desired automorphism can be expressed as a

symbol renaming operation, which is selected to fix the destination node to  $\pi_0$ . For example, let  $\pi_s = 45231$  and  $\pi_d = 31245$ . To map  $\pi_d$  into  $\pi_0$ , symbols 3, 1 and 2 are renamed as 1, 2 and 3, respectively. This symbol renaming operation, when applied to  $\pi_s$ , yields  $\pi_{ds} = 45312$ .

This discussion can be extended to  $SCC_n$  as follows. Permutations  $\pi_s$ ,  $\pi_d$ ,  $\pi_{ds}$  and  $\pi_0$  are viewed as labels of supernodes in  $SCC_n$ , such that routing between two nodes  $\langle i_s, \pi_s \rangle$  and  $\langle i_d, \pi_d \rangle$  in  $SCC_n$  becomes equivalent to routing from  $\langle i_s, \pi_{ds} \rangle$  to  $\langle i_d, \pi_0 \rangle$ .

Let  $P(\ell_1 \mapsto \ell_f)$  denote a route from  $\langle i_s, \pi_s \rangle$  to  $\langle i_d, \pi_d \rangle$  in  $SCC_n$ , which traverses a sequence of  $f$  lateral links  $R(\ell_1 \mapsto \ell_f) = (\ell_1, \ell_2, \dots, \ell_f)$ . The total cost of  $P(\ell_1 \mapsto \ell_f)$  is given with:

$$|P(\ell_1 \mapsto \ell_f)| = f + d(i_s, \ell_1) + \sum_{j=1}^{f-1} d(\ell_j, \ell_{j+1}) + d(\ell_f, i_d). \quad (15)$$

Depending on the order chosen to execute the  $r$ -cycles in  $\pi_{ds}$ , different routes  $P(\ell_1 \mapsto \ell_f)$  are produced. As explained in the previous section, a common feature to any of these routes is that they all have the same number of lateral links ( $lat$ ) and  $MI$  local links ( $MI(loc)$ ). Finding the shortest route from  $\langle i_s, \pi_s \rangle$  to  $\langle i_d, \pi_d \rangle$  is, therefore, a matter of choosing an  $r$ -cycle ordering which minimizes the number of  $MB$  local links ( $MB(loc)$ ). In this section, two non-minimal routing algorithms are described. A minimal (but more complex) routing algorithm is given in [17] and is briefly discussed in this paper.

To illustrate the different cost components in a route and how they are affected by the order chosen to execute the  $r$ -cycles, assume routing from node  $\langle 3, 34125 \rangle$  to node  $\langle 2, 12345 \rangle$  in  $SCC_5$ . A route along the sequence  $R(2 \mapsto 3) = (2, 4, 2, 3)$  contains four lateral links, four  $MI$  local links and three  $MB$  local links (i.e.,  $|P(2 \mapsto 3)| = 4 + 4 + 3 = 11$ ). However, if the sequence of lateral links  $R(3 \mapsto 2) = (3, 2, 4, 2)$  is used, a route with four lateral links, four  $MI$  local links and one  $MB$  local link results (i.e.,  $|P(3 \mapsto 2)| = 4 + 4 + 1 = 9$ ).

In some cases, the number of  $MB$  local links in a route from  $\langle i_s, \pi_s \rangle$  to  $\langle i_d, \pi_d \rangle$  can be further reduced by interleaving (rather than executing separately) the  $r$ -cycles in  $\pi_{ds}$ . For example, some possible sequences of lateral links from supernode  $\pi_{ds} = 23154 = (1\ 2\ 3)(4\ 5)$  to supernode  $\pi_0 = 12345$  in  $SCC_5$  are  $(2, 3, 4, 5, 4)$ ,  $(2, 3, 5, 4, 5)$ ,  $(4, 5, 4, 2, 3)$ ,  $(5, 4, 5, 2, 3)$ ,  $(2, 4, 5, 4, 3)$  and  $(2, 5, 4, 5, 3)$ . The last two of these sequences interleave  $r$ -cycles  $(1\ 2\ 3)$  and  $(4\ 5)$ . All of the routing algorithms discussed in this paper account for the possibility of interleaving  $r$ -cycles.

### Random Routing Algorithm

A simple routing algorithm for  $SCC_n$  consists of choosing a random order to execute the  $r$ -cycles in  $\pi_{ds}$ . Particularly, a possible algorithm that can be used for this purpose is the routing algorithm of the star graph [14]:

#### Algorithm 1 (Non-Deterministic Routing in the Star Graph)

Repeat until  $\pi_{ds} = \pi_0$ :

1. If the first symbol in  $\pi_{ds}$  is 1, then exchange it with any symbol not in its correct position.
2. If the first symbol in  $\pi_{ds}$  is  $x \neq 1$ , then either exchange it with the symbol at position  $x$ , or exchange it with any symbol in an  $r$ -cycle of length at least two, other than the  $r$ -cycle containing  $x$ .

Algorithm 1 requires at most  $c + m$  steps of complexity  $O(1)$  each, and, therefore, its complexity is  $O(c + m)$ , or  $O(n)$ , since  $0 \leq c \leq \lfloor n/2 \rfloor$  and  $1 \leq m \leq n$ .

### Greedy Routing Algorithm

A simple approach to minimizing the number of  $MB$  local links in the route between nodes  $\langle i_s, \pi_s \rangle$  and  $\langle i_d, \pi_d \rangle$  consists of using a greedy algorithm. Such an algorithm uses the following data structures and variables:

- $\mathcal{S}_c$  – the set of  $r$ -cycles of length at least 2 in  $\pi_{ds}$ .
- $\mathcal{S}_s$  – a subset of the symbols of  $\pi_{ds}$ , such that:
  - 1) if  $(1\ i_1 \dots i_{r-1})$  is an  $r$ -cycle of  $\pi_{ds}$ ,  $2 \leq$



$r \leq n$ , then  $i_1 \in \mathcal{S}_s$  and  $1, i_2, \dots, i_{r-1} \notin \mathcal{S}_s$  and 2) if  $(i_0 \dots i_{r-1})$  is an  $r$ -cycle of  $\pi_{ds}$ ,  $2 \leq r \leq n$ , such that  $i_0, \dots, i_{r-1} \neq 1$ , then  $i_0, \dots, i_{r-1} \in \mathcal{S}_s$ .

- $i_j$  – an integer variable initialized to  $i_j = i_s$ .

### Algorithm 2 (Greedy Routing in the SCC Graph)

1. If  $\pi_{ds} = \pi_0$ , then route inside the supernode and exit.
2. Identify the  $r$ -cycles of length at least 2 that exist in  $\pi_{ds}$  and initialize  $\mathcal{S}_c$ ,  $\mathcal{S}_s$  and  $i_j$ .
3. Choose a symbol  $i_\alpha \in \mathcal{S}_s$  such that  $d(i_j, i_\alpha)$  is minimal. Let  $C_\alpha$  be the  $r$ -cycle that contains symbol  $i_\alpha$ . Once  $i_\alpha$  is chosen, make  $i_j = i_\alpha$ .
4. If  $C_\alpha$  has the form  $(1 i_\alpha i_\beta \dots i_{r-1})$ , then make  $\mathcal{S}_c = \mathcal{S}_c - \{(1 i_\alpha i_\beta \dots i_{r-1})\} + \{(1 i_\beta \dots i_{r-1})\}$  and  $\mathcal{S}_s = \mathcal{S}_s - \{i_\alpha\} + \{i_\beta\}$ . Otherwise, make  $\mathcal{S}_c = \mathcal{S}_c - \{C_\alpha\}$  and  $\mathcal{S}_s = \mathcal{S}_s - \{\text{symbols}(C_\alpha)\}$ , where  $\text{symbols}(C_\alpha)$  denote a function that returns the set of symbols in  $r$ -cycle  $C_\alpha$ .
5. Repeat Steps 3 and 4 until  $\mathcal{S}_c = \emptyset$ .
6. If  $i_j \neq i_d$ , then route locally inside the destination supernode.

The greedy approach used by Algorithm 2 consists of choosing the  $r$ -cycle that has the minimum distance from  $i_j$  as the next one to be executed. If the selected  $r$ -cycle  $C_\alpha$  includes symbol 1, then only the first lateral link of  $C_\alpha$  is taken, which allows for an interleaved execution of that  $r$ -cycle. If  $C_\alpha$  does not include symbol 1, then  $C_\alpha$  is executed completely. The complexity of the greedy routing algorithm is  $O(cm)$  or  $O(n^2)$  since  $0 \leq c \leq \lfloor n/2 \rfloor$  and  $0 \leq m \leq n$ . The ordering of  $r$ -cycles chosen by this algorithm, however, may not produce a minimal route.

### Minimal Routing Algorithm

A detailed description of a minimal routing algorithm for  $SCC_n$  is given in [17] and is avoided in this paper due to space constraints. Because the minimal routing algorithm achieves exact numerical solutions for the average distance of

SCC graphs, its simulation results are included in this paper. Furthermore, these results constitute a useful reference for comparisons with the random and greedy routing algorithms.

The main characteristics of the minimal routing algorithm can be summarized as follows. The algorithm performs search on a weighted tree structure and employs selective heuristics to constrain tree branching. To guarantee that a minimal path is always found, the heuristics are constantly monitored against a path cost threshold. A backtracking mechanism is triggered when active branching options do not meet the current path cost threshold. In this case, it is possible that a previously discarded branching option will be actively pursued by the algorithm again. Backtracking also causes the path cost threshold to be reevaluated. The algorithm has an average computational complexity of  $O(n^3)$ .

## SIMULATION RESULTS

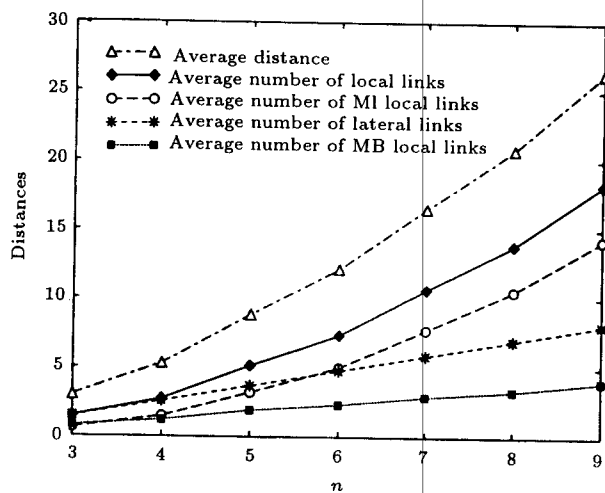
The performance of routing algorithms for  $SCC_n$  was evaluated with simulation programs which compute the route of all  $(n-1)n!$  nodes of the graph to the identity. The routing algorithms that were tested are: 1) a random routing algorithm that generates all possible routes to the identity with equal probability, which is based on Algorithm 1, 2) Algorithm 2 and 3) the minimal routing algorithm given in [17]. The simulations were carried out for  $3 \leq n \leq 9$ . A log of worst-case routes that may result from the random routing algorithm was also made.

Table 1 and Figure 4 show the simulation results obtained with the minimal routing algorithm. Values for  $\overline{lat}$  and  $\overline{MI(loc)}$  match exactly the theoretical values provided by Equations 6 and 9. Also, the simulation results obtained for  $\overline{MB(loc)}$  under a minimal routing algorithm are closely bounded by Equation 12.

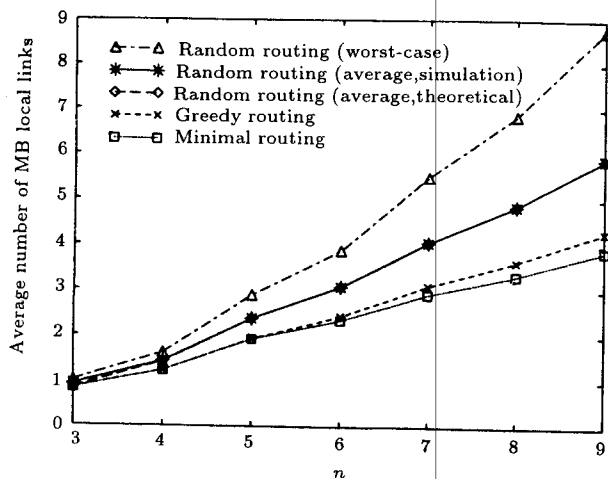
As expected, only the average number of  $MB$  local links varied among the different routing algorithms that were tested. Figure 5 compares simulation results for  $\overline{MB(loc)}$ . Note that the results for the random routing algo-

**Table 1.** Average distance of SCC graphs under minimal routing.

$n$	3	4	5	6	7	8	9
Graph Size $((n - 1) \cdot n!)$	12	72	480	3600	30240	282240	2903040
Graph Diameter $(\phi(SCC_n))$	6	8	16	19	31	34	50
Average Number of Lateral Links $(\overline{lat})$	1.500	2.583	3.683	4.783	5.879	6.968	8.051
Average Number of MI Local Links $(\overline{MI(loc)})$	0.667	1.500	3.200	5.000	7.714	10.500	14.222
Average Number of MB Local Links $(\overline{MB(loc)})$	0.833	1.222	1.925	2.337	2.924	3.334	3.873
Average Number of Local Links $(\overline{loc})$	1.500	2.722	5.125	7.337	10.638	13.834	18.096
Average Distance $(\phi(SCC_n))$	3.000	5.306	8.808	12.121	16.517	20.802	26.147



**Figure 4.** Average distances under minimal routing.



**Figure 5.**  $\overline{MB(loc)}$  vs routing algorithms.

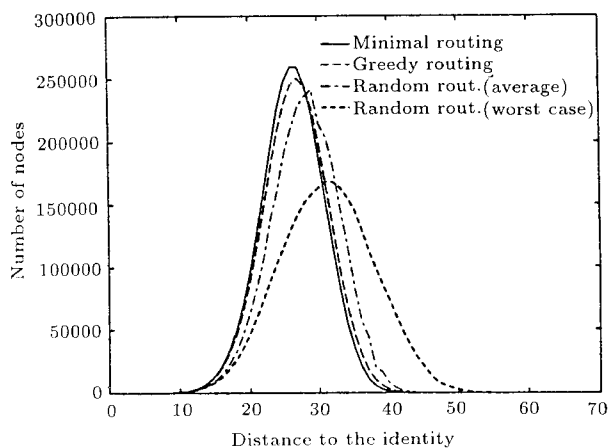
rithm are very close to the theoretical values provided by Equation 12. The model used to derive that equation seems to result in an error proportional to  $1/n!$ , which is negligible considering that Equation 12 is still a close upper bound for  $\overline{MB(loc)}$ . As expected, both the greedy and the minimal routing algorithm outperform the random routing algorithm, as far as the average number of MB local links is concerned. Also observe that, for  $3 \leq n \leq 4$ , the greedy routing algorithm performs as well as the minimal routing algorithm. Besides, the results presented here indicate that the performance of these algorithms is quite similar for  $5 \leq n \leq 9$ , which makes the less complex greedy routing algorithm particularly attractive.

Average costs of paths produced by the three routing algorithms are summarized in Table 2. The random routing algorithm has a complexity of  $O(n)$  and performs reasonably well on the average. Utilization of such an algorithm may, however, result in variations in the average cost of routes up to the worst-case values shown in Table 2.

Figure 6 shows distribution curves comparing the three routing algorithms in the case of an  $SCC_9$  graph. A point  $(D_I, N_I)$  in one of these curves indicates that the corresponding routing algorithm will compute a route of cost  $D_I$  to the identity for  $N_I$  nodes in the SCC graph. The average distribution for the random routing algorithm is shown, but the results

**Table 2.** Average costs vs routing algorithms.

$n$	Minimal Routing	Greedy Routing	Random Routing		
			Theoretical	Simulation	Worst-Case
3	3.000	3.000	3.000	3.084	3.167
4	5.306	5.305	5.500	5.514	5.694
5	8.808	8.812	9.261	9.264	9.775
6	12.121	12.215	12.858	12.858	13.662
7	16.517	16.707	17.660	17.660	19.100
8	20.802	21.109	22.332	22.332	24.324
9	26.147	26.570	28.168	28.168	31.043

**Figure 6.**  $D_I \times N_I$  distribution curves for  $SCC_9$ .

for that algorithm may actually vary from the minimal to the worst-case distribution curves due to the non-deterministic nature of the algorithm. It is also interesting to observe that the greedy routing algorithm provides a distribution curve which is close to that of the minimal routing algorithm, presenting, however, a smaller complexity.

## CONSIDERATIONS ON WORMHOLE ROUTING

In this section, it is briefly described how the algorithms presented in the paper can be combined with wormhole routing [11], which is a popular switching technique used in parallel computers.

All three algorithms can be used with wormhole routing, when implemented as source-based routing algorithms [1]. In source-

based routing, the source node selects the entire path before sending the packet. Because the processing delay for the routing algorithm is incurred only at the source node, it adds only once to the communication latency and can be viewed as part of the start-up latency. Source-based routing, however, has two disadvantages: 1) each packet must carry complete information about its path in the header, which increases the packet length and 2) the path cannot be changed while the packet is being routed, which precludes incorporating adaptivity into the routing algorithm.

Distributed routing eliminates the disadvantages of source-based routing by invoking the routing algorithm in each node to which the packet is forwarded [1]. Thus, the decision on whether a packet should be delivered to the local processor or forwarded on to an outgoing link is done locally by the routing circuit of a node. Because the routing algorithm is invoked multiple times while a packet is being routed, the routing decision must be taken as fast as possible. From this viewpoint, it is important that the routing algorithm can be easily and efficiently rendered in hardware, which favors the random routing algorithm over the greedy and minimal routing algorithms.

Besides being the most complex algorithm discussed in this paper, the minimal routing algorithm includes a feature which precludes its distributed implementation in association with wormhole routing, namely its backtracking mechanism. Distributed versions of the random and greedy algorithms, however, can be used in

combination with wormhole routing. A near-minimal distributed routing algorithm which supports wormhole routing can be obtained by removing the backtracking mechanism from the minimal algorithm given in [17]. Such an algorithm is likely to have computational complexity and average cost that lie between those of the greedy and the minimal routing algorithm.

Due to its non-deterministic nature, the random routing algorithm also seems to be a good candidate for the SCC networks employing distributed adaptive routing [1]. Adaptivity is desirable, for example, if the routing algorithm must dynamically respond to network conditions such as congestion and faults. Some degree of adaptivity is also possible in the greedy and minimal routing algorithms, which in some cases can decide between paths of equal cost.

## CONCLUSION

This paper compared the average cost and the complexity of three different routing algorithms for the SCC graph. Routes were divided into three components (lateral links, *MI* local links and *MB* local links) and it was shown that only the number of *MB* local links may be affected by the routing algorithm being considered. Exact expressions for the average number of lateral links and the average number of *MI* local links were presented. Also, an upper bound for the average number of *MB* local links was derived, considering a random routing algorithm. As a result, a tight upper bound on the average distance of the SCC graph was obtained.

Simulation results for a random, a greedy and a minimal routing algorithm were presented and compared with theoretical values. The complexity of the proposed algorithms is respectively  $O(n)$ ,  $O(n^2)$  and  $O(n^3)$ , where  $n$  is the dimensionality of the  $SCC_n$  graph. The results under minimal routing produce exact numerical values for the average distance of  $SCC_n$ , for  $3 \leq n \leq 9$ .

Results for the greedy algorithm match

those of the minimal algorithm for  $3 \leq n \leq 4$ . The greedy algorithm also performs close to minimality for  $5 \leq n \leq 9$  and is an interesting choice due to its  $O(n^2)$  complexity. The random routing algorithm has an  $O(n)$  complexity and performs fairly well on the average, but may introduce additional *MB* local links in the route under worst-case conditions.

Finally, it was discussed how each of the routing algorithms can be used in association with the wormhole routing switching technique. Directions for future research in this area include an evaluation of requirements for deadlock avoidance (e.g., number of virtual channels).

## ACKNOWLEDGMENTS

This research was supported in part by Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq - Brazil), under grant No. 200392/92-1.

The authors would also like to thank the anonymous referees for their thorough suggestions and many insightful comments which assisted in improving the presentation of this article.

## REFERENCES

1. Ni, L.M. and McKinley, P.K. "A survey of wormhole routing techniques in direct routing networks", *Computer*, pp 62-76 (Feb. 1993).
2. Latifi, S., Azevedo, M.M. and Bagherzadeh, N. "The star-connected cycles: A fixed-degree interconnection network for parallel processing", *Proc. Int'l Conf. Par. Proc.*, **1**, pp 91-95 (1993).
3. Akers, S.B., Harel, D. and Kishnamurthy, B. "The star graph: An attractive alternative to the  $n$ -cube", *Proc. Int'l Conf. Par. Proc.*, pp 393-400 (1987).
4. Preparata, F.P. and Vuillemin, J. "The cube-connected cycles: A versatile network for parallel computation", *Comm. ACM*, **24**(5), pp 300-309 (1981).
5. Saad, Y. and Schultz, M.H. "Topological properties of hypercubes", *IEEE Trans. Comp.*, **37**(7), pp 867-872 (1988).

6. Azevedo, M.M., Bagherzadeh, N. and Latifi, S. "Broadcasting algorithms for the star-connected cycles interconnection network", *J. Par. Dist. Comp.*, **25**, pp 209–222 (1995).
7. Shoari, S. and Bagherzadeh, N. "Computation of the fast Fourier transform on the star-connected cycle network", *Comp. & Elec. Engr.*, **22**(4), pp 235–246 (1996).
8. Azevedo, M.M., Bagherzadeh, N. and Latifi, S. "Embedding meshes in the star-connected cycles interconnection network", to appear in *Math. Mod. and Sci. Comp.*
9. Vadapalli P. and Srimani, P.K. "Two different families of fixed degree regular Cayley networks", *Proc. Int'l Phoenix Conf. Comp. Comm.*, pp 263–269 (1995).
10. Azevedo, M.M., Bagherzadeh, N. and Latifi, S. "Fault-diameter of the star-connected cycles interconnection network", *Proc. 28th Annual Hawaii Int'l Conf. Sys. Sci.*, **II**, pp 469–478 (1995).
11. Dally, W.J. and Seitz, C.I. "The torus routing chip", *Dist. Comp.*, **1**(4), pp 187–196 (1986).
12. Chen, W.-K., Stallmann, M.F.M. and Gehring, E.F. "Hypercube embedding heuristics: An evaluation," *Int'l J. Par. Prog.*, **18**(6), pp 505–549 (1989).
13. Knuth, D.E. *The Art of Computer Programming*, **1**, Addison-Wesley, pp 73, 176–177 (1968).
14. Day, K. and Tripathi, A. "A comparative study of topological properties of hypercubes and star graphs", *IEEE Trans. Par. Dist. Sys.*, **5**(1), pp 31–38 (1994).
15. Latifi, S. "Parallel dimension permutations on star graph", *IFIP Trans. A: Comp. Sci. Tech.*, **A23**, pp 191–201 (1993).
16. Akers, S.B. and Krishnamurthy, B. "A group-theoretic model for symmetric interconnection networks", *Proc. Int'l Conf. Par. Proc.*, pp 216–223 (1986).
17. Azevedo, M.M. *Star-Based Interconnection Networks and Fault-Tolerant Clock Synchronization for Large Multicomputers*, Ph.D. Dissertation, University of California, Irvine, USA (1997).