Appears in *Proceedings of the 22nd International Conference on Parallel Processing, St. Charles, Illinois, August 16-20, 1993, Vol. 1, pp. 91-95.*

THE STAR CONNECTED CYCLES: A FIXED-DEGREE NETWORK FOR PARALLEL PROCESSING

Shahram Latifi
Department of Electrical and Computer Engineering
University of Nevada, Las Vegas
Las Vegas, NV 89154-4026
latifi@jb.ee.unlv.edu

Marcelo Moraes de Azevedo * *and Nader Bagherzadeh*
Department of Electrical and Computer Engineering
University of California, Irvine - Irvine, CA 92717
mazevedo@balboa.eng.uci.edu
nader@balboa.eng.uci.edu

Abstract: *This paper introduces a new interconnection network for massively parallel systems referred to as star connected cycles (SCC) graph. The SCC presents a fixed degree structure that results in several advantages over variable degree graphs like the star graph and the n-cube.*

The description of the SCC graph given in this paper includes issues such as labeling of nodes, degree, diameter, symmetry, fault tolerance and Cayley graph representation. The paper also presents an optimal routing algorithm for the SCC and a comparison with other interconnection networks. Our results indicate that for even n, an n-SCC and a CCC of similar sizes have about the same diameter.

1 Introduction

Over the past years, many interesting graphs such as the n-star and the n-cube have been proposed as interconnection networks for parallel processing applications. Some important properties shown by these graphs are node and edge symmetry, hierarchical structure, maximal fault tolerance and strong resilience [1]. However, the n-star is superior to the n-cube in several areas.

Most graphs studied so far offer a high processor density while keeping the diameter as low as possible. Nevertheless, graphs such as the n-star and the n-cube present a variable degree structure and have low scaleability from the viewpoint of network growth. More specifically, since both the degree of the n-star and the n-cube is $O(n)$, a growing number of communication links is required as n increases. Hence, one disadvantage of variable degree interconnection networks is the large number of I/O communication ports required at each processor in massively parallel systems.

Variable degree interconnection networks also present more complex physical lay-outs and require additional communication ports at each processor to be expanded. In other words, if we want to increase the number of nodes of an existing variable degree parallel system, it might be necessary to substitute all processors in the system, unless unused communication ports are available at each node.

To overcome these difficulties, we propose a new type of interconnection network: the *star connected cycles* (*SCC*) graph. The SCC offers a fixed-degree structure and can be viewed as an evolution of its counterpart, the cube connected cycles or CCC [2]. The SCC and CCC graphs are formed by connecting cycles or rings of nodes through a particular network communication topology. The underlying topology used to connect the cycles in an n-SCC graph is an n-star, while that of the n-CCC graph is the n-cube. As expected, this results in a fixed-degree interconnection network that is superior to the CCC in several areas.

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

2 Description of the SCC

The SCC interconnection network is based on the well known star graph [1]. An *n-SCC* graph is obtained by substituting each node of an n-star with a ring of $(n-1)$ nodes. Each ring may be viewed as a *supernode* that can be implemented with a cluster of individual processors or with a single multiprocessor VLSI device. A supernode in the n-SCC graph is connected to $(n-1)$ adjacent supernodes, using lateral links according to the topology of the n-star graph. The nodes inside each ring are identified by a pair of labels (I_j, P_i), where:

- P_i is a permutation obtained using the generators of the n-star graph [3]. We consider that the nodes in an n-star are labeled with a permutation of the digits $\{1, 2, \ldots, n\}$, which allows the labeling of $n!$ different nodes in the n-star. Therefore, each permutation P_i of the n-star labels $(n-1)$ nodes belonging to the corresponding supernode in the n-SCC.
- I_j is a single digit that identifies each particular node inside a ring. The labeling method proposed for SCC consists of assigning to each I_j a label in the range $\{2, 3, \ldots, n\}$, such that I_j corresponds to the label of the lateral link used to connect each node within a ring to other rings in the n-SCC graph. The label of the lateral link is chosen so as to represent the position of the digit in the permutation P_i that is swapped with the first digit of P_i when the lateral link is traversed.

As an example, consider the *4*-SCC graph shown in Figure 1. Node 1234 of a *4*-star graph is substituted for 3 nodes, labeled respectively as (2,1234), (3,1234) and (4,1234). These nodes are connected to other rings using lateral links 2, 3 and 4 (e.g. node (2,1234) is connected to node (2,2134) via lateral link 2).

Number of nodes: An n-SCC graph can be seen as an n-star graph connecting $n!$ supernodes. Since each supernode contains $(n-1)$ nodes, the total number of nodes in an n-SCC graph is $N = (n-1)n!$.

Appears in *Proceedings of the 22nd International Conference on Parallel Processing, St. Charles, Illinois, August 16-20, 1993, Vol. 1, pp. 91-95.*

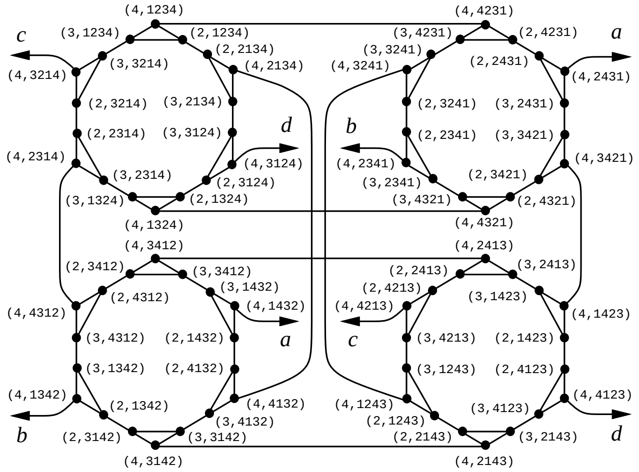

Figure 1: *4*-SCC graph

Degree: The degree of the n-SCC graph is shown in Table 1. Since the degree of every node in the graph is the same, the n-SCC graph is regular. For $n > 3$, every node in a n-SCC graph connects to exactly 3 adjacent nodes using two local links within the same supernode and one lateral link to a node belonging to an adjacent supernode.

A fixed and low degree reduces the communication costs at each node. If each node is implemented as an individual processor, we can use a standard building block with 3 communication links to build any n-SCC graph.

Table 1: Degree and fault tolerance of the n-SCC

Property	$n=2$	$n=3$	$n>3$
Degree (δ)	1	2	3
Fault tolerance (f)	0	1	2

Fault tolerance: A graph is *f-fault-tolerant* if it remains connected when any set of f or fewer nodes are removed from the graph [1]. The *fault tolerance* of a graph corresponds to the largest f for which the graph is f-fault-tolerant. The fault tolerance of a graph with degree δ can be at most equal to $(\delta - 1)$, since if we remove all the neighbors of a node the graph will be disconnected. A graph whose fault tolerance is exactly $(\delta - 1)$ is said to be *maximally fault-tolerant.*

The fault tolerance of the n-SCC graph is shown in Table 1. Clearly, the n-SCC graph is maximally fault-tolerant.

Symmetry: The SCC graph is node symmetric. This is true both from the viewpoint of a single node and from the viewpoint of a supernode. So, given any two nodes a and b there is an automorphism of the graph that maps a to b.

Since not all SCC edges look the same, the SCC graph is not edge symmetric. This implies that the communication load is not uniformly distributed over all communication links. However, if we consider only the lateral links and view the n-SCC as an n-star of supernodes, the edge symmetry properties of the n-star still hold.

Thus, every lateral link in an n-SCC graph is edge symmetric with any other lateral link in the graph. The local links within a ring or supernode are also transitive with any other local link in the graph.

Lemma 1: The n-SCC is not a Cayley graph.

Proof: The generators of a Cayley graph labeled with n digits can either generate S_n or a subgroup of S_n [1], where S_n is the set of all $n!$ possible permutations that can be created with n digits. Each node in an n-SCC graph is actually labeled with $(n+1)$ digits, which means that the labels of an n-SCC graph actually belong to S_{n+1}.

S_{n+1} has $(n+1)!$ different permutations and the n-SCC graph uses only $(n-1)n!$ permutations of S_{n+1}. The relation between these numbers is equal to $(n+1)/(n-1)$.

According to LaGrange's theorem on finite groups [1], the order of any subgroup always divides the order of the group. That's not the case with the ratio above. □

Although the n-SCC graph is not a Cayley graph, its node symmetry property allows that any n-SCC graph can be represented as the quotient of two Cayley graphs [3]. More specifically, we may obtain the quotient graph of an n-SCC graph by identifying subgraphs in the n-SCC and reducing such subgraphs to nodes. The nodes in the resulting quotient graph are connected iff there existed an edge between elements of the corresponding subgraphs. In the case of the n-SCC, each subgraph corresponds to a ring of nodes. Reducing each subgraph to a node results in a Cayley quotient graph: the n-star.

3 Routing in the n-SCC

Routing in the n-SCC is an extension of the routing in the n-star graph and can be seen as two different problems: routing in the lateral links and routing in the local links.

3.1 Routing in the lateral links

Routing in the lateral links uses the same routing techniques already developed for the n-star [1]. Suppose that we want to route from P_s to P_d in an n-star. To find the lateral links connecting P_s to P_d, we can instead find the path from P_{ds} to the identity permutation [3], where $P_{ds} = P_d^{-1} P_s$.

Algorithm 1 (Routing in the n-star):

1. If the first digit in the P_{ds} permutation is 1, move it to any position not occupied by the correct digit.
2. If x (i.e. any digit other than 1) is first, move it to its position.

We may organize the digits of permutation P_{ds} as a set of cycles – i.e. cyclically ordered sets of digits with the property that each digit's desired position is that occupied by the next digit in the set. A permutation $P_{ds} = 26543187$ belonging to an *8*-star graph, for instance, consists of the following cycles: (2 6 1), (5 3), (8 7), (4). Note that any digit already in its correct position appears as a *1*-cycle.

Appears in *Proceedings of the 22nd International Conference on Parallel Processing, St. Charles, Illinois, August 16-20, 1993, Vol. 1, pp. 91-95.*

Let $C = (i_1\ i_2\ \dots\ i_k)$ be a cycle of length $k \leq n$ in P_{ds}, where $1 \leq i_1 < i_2 < \dots < i_k \leq n$. The *execution of cycle C* corresponds to a path R in the n-star and can be expressed as a sequence of lateral links as follows [4]:

$$\begin{array}{ll} R = (i_2, i_3, \dots, i_k) & , \ if\ i_1 = 1 \\ R = (i_1, i_2, \dots, i_{k-1}, i_k, i_1) & , \ if\ i_1 \neq 1 \end{array}$$

Note that in an n-star there are[1] $N_c = c!$ different choices that can be used for an optimal order of execution of cycles of length at least 2 in P_{ds}. If the number of digits in cycle C_i is K_i, $K_i \geq 2$, then there are also N_i different ways to minimally execute cycle C_i, where:

$$N_i = \begin{cases} K_i & , \ if\ C_i\ does\ not\ include\ digit\ 1 \\ 1 & , \ if\ C_i\ includes\ the\ digit\ 1 \end{cases}$$

If the cycles C_i for which $K_i \geq 2$ are $C_1, C_2, \dots, C_c$ then the total number of optimal routing paths in the n-star from P_s to P_d is:

$$T_p = c! \prod_{i=1}^{i=c} N_i$$

3.2 Routing in the local links

Assume that the nodes belonging to the same ring are labeled with $I = 2, \dots, n$, going counterclockwise. Also, suppose that the lateral link entering the supernode is I_i and the lateral link leaving the supernode is I_j. If D_n is the minimum number of local links between I_i and I_j and $D' = |I_j - I_i|$, then:

$$D_n = min\,(D', n - 1 - D') \qquad (1)$$

Algorithm 2 (Routing in the local links):

1. Evaluate $L = I_j - I_i$ and $R = |I_j - I_i|\, div \left\lfloor \frac{n+1}{2} \right\rfloor$, where *div* is the integer division operator.
2. If ($L > 0$ and $R = 0$) or ($L < 0$ and $R = 1$), then take the ring counterclockwise traversing D_n local links; else take the ring clockwise traversing D_n local links.

3.3 Routing algorithm for the n-SCC

We now present an algorithm for routing in the n-SCC graph. Such algorithm is actually a combination of Algorithms 1 and 2 and provides a sequence of lateral and local links as a result.

We recall that Algorithm 1 allows for T_p different optimal paths in an n-star graph. The edges of the n-star graph are the lateral links of the corresponding n-SCC graph. However, not all of the T_p different optimal paths that exist in the n-star result in optimal paths in the n-SCC, since the order of execution of the lateral links affects the number of local links in the routing.

The routing algorithm for the n-SCC graph performs a depth-first search on a weighted tree structure. The algorithm builds the tree by expanding at each step those cycle orderings that seem to result in a minimal number of local links. Backtracking is also performed to enable expansion of previous cycle orderings that seem to be equivalent or better than the most recently expanded orderings.

In the description of the routing algorithm for the n-SCC, we denote the source node by (I_s, P_s) and the destination node by (I_d, P_d). We also define the following:

- S_c is the set of cycles of length at least 2 in P_{ds}.
- S_d is a subset of the digits of P_{ds}, such that:
  - If $(1\ i_2\ i_3\ \dots\ i_k)$ is a cycle of S_c, then $i_2 \in S_d$ and $1, i_3, \dots, i_k \notin S_d$.
  - If $(i_1\ i_2\ \dots\ i_k)$ is a cycle of S_c that does not include digit 1, then $i_1, i_2, \dots, i_k \in S_d$.

The tree structure generated by the routing algorithm has the following characteristics:

- If the number of cycles in S_c is c, then the first level of the tree is 0 and the deepest level is $(c+1)$.
- The label of any vertex in the tree is a pair of digits (f, ℓ), belonging either to S_d or to $\{I_s, I_d\}$. Each vertex located between levels 1 and c in the tree represents one of the cycles of S_c. The label (f, ℓ) is chosen so as to represent the first (f) and last (ℓ) lateral link used during the execution of the cycle.

  If the cycle represented by the vertex is $(1\ i_2\ i_3\ \dots\ i_k)$, then the vertex is labeled $(f, \ell) = (i_2, i_k)$. If the cycle represented by the vertex does not include digit 1, then the vertex may be labeled as $(f, \ell) = (i_i, i_i)$, where i_i is any of the digits of the cycle.
- The weight of an edge connecting any two vertices (f_i, ℓ_i) and (f_j, ℓ_j) corresponds to the number of local links required to move from ℓ_i to f_j within the same supernode and is given by Equation (1).
- Each vertex (f, ℓ) has an associated data structure consisting of its distance to the root (D_r) and a reduced set of digits $S_d^c = S_d^p - S_i$. The distance D_r is obtained by summing the weights of all edges in the path from the root to the vertex. S_d^p is the set of digits stored in the parent of each vertex and S_i is the set of digits belonging to the cycle that includes digit f.
- The root vertex is $(f, \ell) = (I_s, I_s)$ and has $D_r = 0$ and $S_d^c = S_d$. The vertices located at level (c+1) in the tree are labeled $(f, \ell) = (I_d, I_d)$ and have $S_d^c = \{\}$. The vertices located at level c in the tree have $S_d^c = \{I_d\}$.
- Each vertex also stores an enable/disable bit that informs whether the tree should continue to be expanded from that vertex or not. The root vertex is created with an enabled bit, but all other child vertices are created with a disabled bit. Vertices that have already been expanded also have a disabled bit.

[1] Note that this equation is valid even if the first digit in P_{ds} is not 1. Although Algorithm 1 indicates that the cycle including digit 1 should be executed first, we may actually choose any order to execute the cycles, one at a time. This is possible because the execution of any cycle leaves the position of digits that do not belong to that cycle unchanged.

Appears in *Proceedings of the 22nd International Conference on Parallel Processing, St. Charles, Illinois, August 16-20, 1993, Vol. 1, pp. 91-95.*

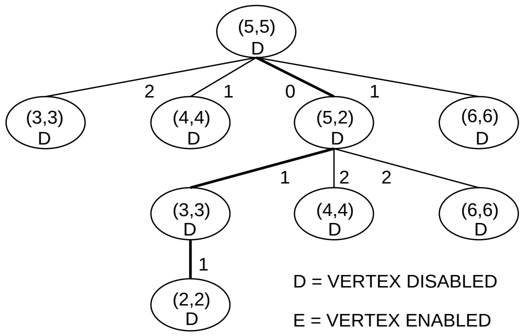

Figure 2: Example of routing tree in the n-SCC

Given the definitions above, the routing algorithm for the n-SCC is as follows :

Algorithm 3 (Routing algorithm for the n-SCC):

1. If $P_s = P_d$, then route inside the ring using Algorithm 2 and exit.
2. If $P_s \neq P_d$, then calculate permutation P_{ds} such that $P_{ds} = P_d^{-1} P_s$. Also identify the cycles of length at least 2 that exist in P_{ds} and create the sets S_c and S_d.
3. Create an enabled root vertex labeled (I_s, I_s) such that $S_d^c = S_d$.
4. Generate child vertices for all enabled vertices, such that the label f for each child corresponds to exactly one of the digits stored in the set S_d^p of each parent vertex. The label ℓ for each child vertex is chosen according to the definitions of the tree structure given earlier in this section.
5. Evaluate D_r and S_d^c for all child vertices. Check if any recently generated child vertex has a distance $D_n = 0$ to its parent. If such vertex exists, enable it. Otherwise, enable all recently generated child vertices that have a distance $D_n = 1$ to their parents. If $D_n > 1$ for all recently generated child vertices, then it is necessary to perform a backtracking search in the tree. Such search enables all vertices that have the smallest *virtual distance* (D_v) to the end of the tree, where $D_v = D_r + c + 1 - h$ and h is the level at which the vertex is located in the tree $(0 \leq h \leq c+1)$.
6. If all enabled vertices are at level $c+1$ of the tree, then an optimal order of execution for the cycles in S_c has already been found. Otherwise, return to Step 5.
7. The optimal order of execution for the cycles in S_c is given by the intermediate vertices existing between the root and any enabled vertex at level $c+1$ in the tree. This optimal order of execution is actually a sequence of lateral links. Additional routing is required to move between lateral links, but that can be easily done with Algorithm 2.

As an example, consider the routing in a 7-SCC for the case $I_s = 5$, $I_d = 2$ and $P_{ds} = (1\ 5\ 2)\,(3\ 6\ 4)\,(7)$. Figure 3 shows the tree built by the routing algorithm. An optimal order of execution for P_{ds} is therefore 5 - 2 – 3 - 6 - 4 - 3.

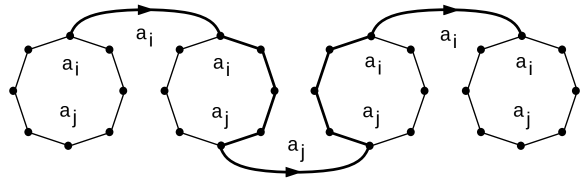

Figure 3: Execution of a cycle $(a_i\ a_j)$ in the n-SCC

4 Diameter

The diameter of the n-SCC graph can be calculated by finding its antipode permutations. An antipode is the farthest node from a given node along the shortest path [4]. We recall that in the n-SCC graph there are $(n-1)$ nodes (I_i, P_1), where I_i is a digit of the set $\{\ 2,\ 3,\ 4,\ \ldots,\ n\ \}$ and P_1 is the identity permutation $123\ldots n$. We must therefore choose one of these $(n-1)$ nodes as the identity node for the n-SCC graph. Let such node be $(I_1, P_1) = (2, 123\ldots n)$. Now, we define an antipode (I_a, P_a) in the n-SCC graph as follows:

- P_a is a permutation chosen such that the node (I_a, P_a) is located $\lfloor 3(n-1)/2 \rfloor$ lateral links away from (I_1, P_1), while keeping the number of local links in the path to the identity node to a maximum.
- I_a is a digit chosen so as to include a maximum number of additional local links in the path to (I_1, P_1).

Lemma 2: The following nodes are antipodes in the n-SCC graph:

For odd n: $(2, (1)\,(2\ a{+}1)\,(3\ a{+}2)\ldots(a{-}1\ n{-}1)\,(a\ n))$
For even n: $(2, (1\ b{+}1)\,(2\ b{+}2)\ldots(b{-}1\ n{-}1)\,(b\ n))$

where $a = (n+1)/2$ and $b = n/2$.

Reference [5] gives a proof of correctness for these antipodes and lists other possible antipodes.

Theorem 1: The diameter of the n-SCC, $n \geq 2$, is :

$$d_{SCC} = \begin{cases} 2\left(\left\lfloor \frac{n-1}{2} \right\rfloor\right)^2 + \left\lfloor \frac{3(n-1)}{2} \right\rfloor + 2\left\lfloor \frac{n}{2} \right\rfloor - 2 & , n \neq 3 \\ 6 & , n = 3 \end{cases}$$

Proof: Routing from any of the above antipodes requires $d_{lat} = \lfloor 3\,(n-1)/2 \rfloor$ lateral links. The number of local links can be calculated as follows:

- Permutation P_a has $\lfloor (n-1)/2 \rfloor$ cycles that does not include the digit 1. Execution of each of these cycles requires $2\lfloor (n-1)/2 \rfloor$ local links, as can be seen in Figure 4. Thus, the total number of local links required for execution of all cycles in P_a that do not include digit 1 is $d_{loc}(1) = 2\left(\lfloor (n-1)/2 \rfloor\right)^2$.
- Permutation P_a has $\lfloor n/2 \rfloor$ cycles of length 2 that must be executed in the route to the identity node. The cycles in P_a may be ordered such that only one local link is required to move between the execution of adjacent cycles. This adds $d_{loc}(2) = \lfloor n/2 \rfloor - 1$ local links to the routing from the antipode to the identity.

Appears in *Proceedings of the 22nd International Conference on Parallel Processing, St. Charles, Illinois, August 16-20, 1993, Vol. 1, pp. 91-95.*

- Digit I_a in the antipode is such that at most $d_{loc}(3) = \lfloor n/2 \rfloor - 1$ local links may be added to the routing.
- The diameter of the n-SCC graph is the sum of d_{lat}, $d_{loc}(1)$, $d_{loc}(2)$ and $d_{loc}(3)$, so the result follows. □

5 Comparison to other graphs

A comparison between different interconnection network graphs is shown in Table 2. Due to its fixed-degree structure, processors with just 3 communication links can be used to build any n-SCC graph. Clearly, the n-SCC graph has higher scaleability than variable degree graphs like the n-cube and the n-star. Such graphs require a growing number of communication links at each processor as we increase the number of nodes. The result is increased complexity and higher pin count at each processor than required by fixed-degree graphs.

Table 2: Comparison of interconnection networks

Graph	*n*	*Size*	*Degree*	*Diameter*
n-cube	7	128	7	7
	8	256	8	8
	9	512	9	9
n-star	5	120	4	6
	6	720	5	7
	7	5040	6	9
n-CCC	4	64	3	8
	5	160	3	10
	6	384	3	13
	8	2048	3	18
	9	4608	3	20
n-SCC	4	72	3	8
	5	480	3	16
	6	3600	3	19

One of the trade-offs of fixed-degree graphs is an increased diameter. However, the n-SCC can be built with very high speed buses in the local links. The resulting communication delays may be comparable to the n-star, if we consider that the lateral links often use serial transmission for making their lay-out simpler.

Table 2 also shows another type of fixed-degree interconnection network, namely the cube connected cycles or CCC. An n-CCC graph can be built by substituting each node of an n-cube with a ring of n or more nodes. Table 2 shows typical values for n-CCC graphs containing n nodes in each ring. The number of nodes and diameter of an n-CCC graph formed under such structure are given respectively by $N = n2^n$ and $d_{CCC} = 2n + \lfloor n/2 \rfloor - 2$ [6].

Compared to a CCC graph of similar size, the n-SCC graph presents about the same diameter for the cases where n is even. The diameter of the n-SCC graph shows a sharp discontinuity when n changes from an even to an odd value. Such behavior is due to the presence of the quadratic component in the n-SCC diameter expression.

The diameter of the CCC compares favorably with the n-SCC graph for odd n. However, the underlying topology or quotient Cayley graph used to connect the cycles in the n-SCC (i.e., the n-star) has several advantages over that used in the CCC (i.e., the n-cube) [1]. Among these advantages, we may cite a smaller degree from the viewpoint of the supernodes, as well as a shorter average distance and fault diameter. More specifically, an n-SCC graph requires fewer lateral links and fewer nodes at each ring than a CCC graph with similar number of nodes. Such characteristic reduces the complexity of the supernodes and makes their implementation simpler.

6 Conclusion

The SCC is a fixed-degree graph and has been proposed as an evolution of the previous cube connected cycles or CCC. Aspects such as labeling of nodes, degree, diameter, symmetry, fault tolerance and Cayley graph representation have been presented. We have also developed an optimal routing algorithm for the n-SCC.

We have compared the n-SCC with variable degree graphs such as the star graph and the n-cube. We claim that the n-SCC is an attractive alternative for parallel systems, since it overcomes some disadvantages of variable degree graphs such as the increased requirement of communication ports at each node and the reduced scaleability from the viewpoint of network growth.

We have also compared the n-SCC with another fixed-degree graph, namely the CCC graph. We have shown that the diameter of the n-SCC is close to that of a CCC graph containing a similar number of nodes whenever n is even. However, even for the cases where n is odd, the n-SCC shows some superiority over the CCC, due to the use of the n-star as its quotient graph.

References

[1] S. B. Akers, D. Harel and B. Krishnamurthy, "The Star graph: An Attractive Alternative to the n-cube," *Int'l Conf. on Parallel Processing*, 1987, pp. 393-400.

[2] F. P. Preparata and J. Vuillemin, "The Cube-Connected Cycles: A Versatile Network for Parallel Computation," *Communications of the ACM*, Vol. 24, No. 5, May 1981, pp. 300-309.

[3] S. B. Akers and B. Krishnamurthy, "A Group-Theoretic Model for Symmetric Interconnection Networks," *IEEE Transactions on Computers*, Vol. 38, No. 4, April 1989, pp. 555-566.

[4] S. Latifi, "Parallel Dimension Permutations on Star Graph," *1993 Working Conf. on Architectures and Compilation Techniques for Fine and Medium Grain Parallelism*, January 1993.

[5] S. Latifi, M.M. Azevedo and N. Bagherzadeh, *The Star Connected Cycles: a Fixed Degree Interconnection Network for Massively Parallel Systems*, Dept. of Electrical and Computer Engineering, Univ. of Calif., Irvine, Technical Report ECE 93-02, March 1993.

[6] D.S. Meliksetian and C.Y.R. Chen, "Communication Aspects of the Cube-Connected Cycles," *Int'l. Conf. Parallel Processing*, Vol. 1, 1990, pp. 579-580.