Available online at www.sciencedirect.com
SCIENCE DIRECT®
Journal of Systems Architecture 51 (2005) 1–14
JOURNAL OF SYSTEMS ARCHITECTURE
www.elsevier.com/locate/sysarc

Hierarchical star: a new two level interconnection network

Wei Shi, Pradip K. Srimani *

Department of Computer Science, Clemson University, 401 Edwards, Clemson, SC 29634, USA

Received 5 October 2003; received in revised form 19 April 2004; accepted 13 May 2004
Available online 24 August 2004

Abstract

We propose a new two level interconnection network topology, hierarchical star networks, HS_n, that uses the star graphs as building blocks. Two level networks have been previously proposed that use hypercube and its variants as building blocks; it has been shown that these two level networks are superior to the networks, that are used as building blocks, in terms of various performance metrics including diameter, cost, fault tolerance, fault diameter etc. Our results show that the proposed family of hierarchical star networks perform very competitively in comparison to star graphs; in addition, the proposed network outperforms all of the two level hierarchical networks proposed earlier that uses hypercubes (or its variations) as building blocks. Thus, our results further reinforce the notion that the star graphs are strong competitors of hypercubes for large multiprocessor design. We also investigate various topological properties of the network including embedding, mapping of parallel algorithms, fault tolerance and broadcasting algorithms.

1. Introduction

A suitable interconnection network is an integral part of any distributed computing system. The network is usually modeled by a symmetric (undirected) graph where the nodes (vertices) denote the processing elements and the edges (arcs) denote the bidirectional communication channels. Interconnection topologies are evaluated in terms of low degree, small diameter, high fault tolerance, low fault diameter etc. One of the most efficient interconnection network has been the well known binary n-cubes or hypercubes; they have been used to design various commercial multiprocessor machines and they have been extensively studied. In search of a viable or even better alternative for hypercubes, another family of regular graphs, called the star graphs [1,2], are being extensively studied; star graphs seem to enjoy most of the desirable properties [5,16,18,19] of the hypercubes

* Corresponding author. Tel.: +1 864 656 7552; fax: +1 864 656 0145.
E-mail address: srimani@cs.clemson.edu (P.K. Srimani).

1383-7621/$ - see front matter
doi:10.1016/j.sysarc.2004.05.003

at considerably less cost; they accommodate more nodes with less interconnection hardware and less communication delay. It has also been shown [8,14,15,17] that many parallel algorithms can be efficiently mapped on these star graphs.

Investigators [4,10,13] have been studying two level interconnection networks which take some known networks and connect them in a complete manner. Authors in [4] have proposed a two level network, called hierarchical folded hypercube network (HFN), using folded hypercubes of [7] as the basic building blocks. In [9,10], authors proposed hierarchical cube networks (HCN) which consist of 2^n basic modules each of which is a hypercube of dimension n and showed that this network is superior to hypercubes. Authors in [4] extended this design concept to propose hierarchical folded hypercube network (HFN) using folded hypercubes of [7] as the basic building blocks; they showed that HFN is superior to HCN in terms of almost all the network parameters as much as folded hypercubes were a topological improvement over the regular hypercubes. Our objective in the present paper is to design such two level hierarchical networks using the star graphs as the basic building blocks and to investigate the topological properties of the resulting family of networks. We show that our proposed network, hierarchical star network (HS) is superior to the original star graphs, hierarchical folded hypercube networks, folded hypercubes in terms of cost of the network, node degree and diameter as well as the HS networks also retain other desired network properties like simple routing strategy, maximal fault tolerance (vertex connectivity) and optimal broadcasting.

2. Hierarchical star network

2.1. Star graph

A star graph S_n, of order n, is defined to be a symmetric graph $G = (V, E)$ where V is the set of $n!$ vertices, each representing a distinct permutation of n elements and E is the set of symmetric edges such that two permutations (nodes) are connected by an edge iff one can be reached from the other by interchanging its first symbol with any other symbol [2]. For example, in S_3, the node representing permutation *abc* have edges to two other permutations (nodes) *bac* and *cba*. Throughout our discussion we denote the nodes by permutations of English alphabets. These star graphs are members of the family of Cayley group graphs. For a star graph S_n of dimension n, there are $n-1$ generators, $\text{swap}_2, \text{swap}_3, \ldots, \text{swap}_n$, where swap_i swaps the first symbol with the i-th symbol of any permutation. Each generator is its own inverse, i.e., the star graph is symmetric. S_n is a $(n-1)$-regular graph with $n!$ nodes and $n!(n-1)/2$ edges. These star graphs have many other interesting topological properties and they compete well with the popular hypercubes in many aspects; see [2,6] for details.

Note: The generator swap_n for the star graph S_n has a special role to play in this paper; so, for an arbitrary node $x = x_1, x_2, \ldots, x_n$ in S_n, we denote the node $\text{swap}_n(x) = x_n, x_2, \ldots, x_1$ by $\hat{x}$ in this paper.

2.2. Hierarchical star graph

A hierarchical star graph network $\text{HS}_{(n,n)}$ of dimension n for any integer $n \geqslant 2$ consists of $n!$ modules (each module is a star graph of dimension n) interconnected by additional edges. Each node in $\text{HS}_{(n,n)}$ is denoted by a two-tuple address (x, y) where both x and y are arbitrary permutations of n distinct symbols. For each node (x, y), x identifies the module the node belongs to and y further identifies the node within the module; thus, for each node (x, y) in $\text{HS}_{(n,n)}$, x is the *module identifier* and y is the *local identifier*. There are two types of edges (links) in $\text{HS}_{(n,n)}$: *local links* that connect two nodes in the same module and *external links* that connect nodes from two different modules.

Definition 1. Consider two arbitrary nodes (x, y) and (x', y') in $\text{HS}_{(n,n)}$; there exists an edge between these two nodes iff one of the following three conditions is satisfied.

(1) $x = x'$ and $y' = \text{swap}_i(y)$ for some i, $2 \leqslant i \leqslant n$;

(2) $x \neq x' \wedge x \neq y$ and $x = y' \wedge x' = y$.
(3) $x \neq x' \wedge x = y$ and $x' = y' \wedge x = \hat{x}'$.

Remark 1

- The links derived from the first condition are called *local links* since they link the two nodes with the same module identifier, while the links derived from the last two conditions are the *external links* since they connect nodes from different modules.
- The external links are further divided into two categories: links derived from the condition (2) are called *non-diameter external links* while the diameter external links are those derived from condition (3).
- Two nodes on a *non-diameter external link* switches their respective module and local identifiers; two nodes on a *diameter external link* have identical module and local identifiers.

Example 1. Fig. 1 shows a $HS_{(3,3)}$. Here, $n = 3$; $HS_{(3,3)}$ consists of six modules each of which is a star graph S_3 of dimension 3. Each node has a two part address (x, y), where both x and y is an arbitrary permutation of three letters "a", "b" and "c".

For notational purposes in the sequel, we use the following symbols to denote different types of edges in $HS_{(n,n)}$:

- $\rightarrow$: Single local link.
- $\Rightarrow$: A group of local links.
- $\mapsto$: An external link. It may be diameter or non-diameter external link.
- $\hookrightarrow$: Non-diameter external link.
- $\rightsquigarrow$: Diameter external link.

Each node in $HS_{(n,n)}$ is assigned a label $((x_1x_2\cdots x_n), (y_1y_2\cdots y_n))$ where $(x_1x_2\cdots x_n)$ is a permutation of n distinct symbols and $(y_1y_2\cdots y_n)$ is also a permutation (not necessarily distinct from

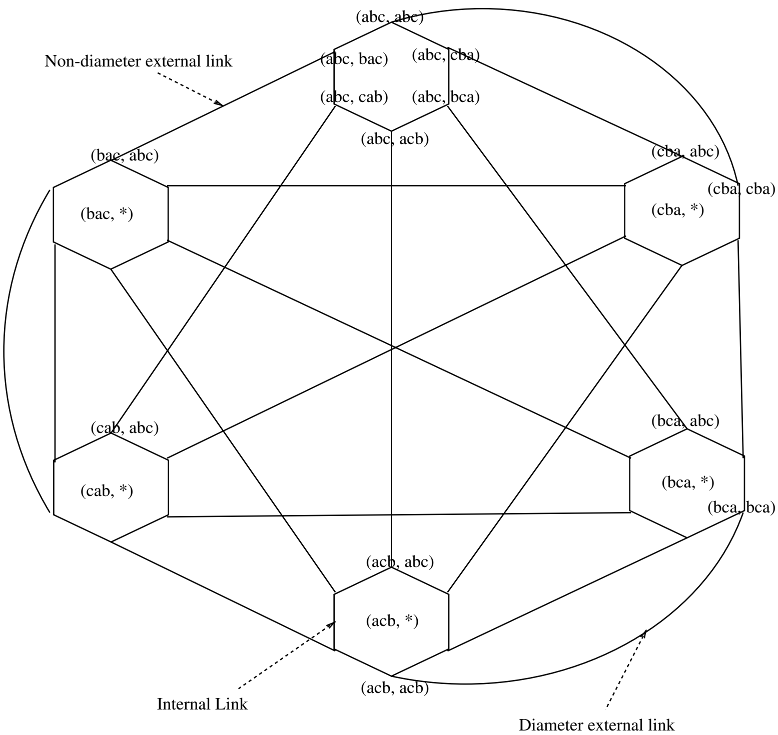

Fig. 1. Hierarchical star network $HS_{(3,3)}$.

$(x_1x_2\cdots x_n)$) of the same n distinct symbols. We refer to $(x_1x_2\cdots x_n)$ as the *module-id* and $(y_1y_2\cdots y_n)$ as the *local-id* of any node in $HS_{(n,n)}$. The edges of the $HS_{(n,n)}$ graph are defined by the following n generators:

$$h_1((x_1x_2\cdots x_n),(y_1y_2\cdots y_n)) = \begin{cases} ((x_nx_2\cdots x_{n-1}x_1),(y_ny_2\cdots y_{n-1}y_1)) \\ \quad \text{if } x_i = y_i, \forall i, 1 \leqslant i \leqslant n \\ ((y_1y_2\cdots y_n),(x_1x_2\cdots x_n)) \\ \quad \text{otherwise} \end{cases}$$

$$h_i((x_1x_2\cdots x_n),(y_1y_2\cdots y_n)) = (x_1x_2\cdots x_n),(y_iy_1y_2\cdots y_{i-1}y_{i+1}\cdots y_n),\ \forall i, 2 \leqslant i \leqslant n$$

Remark 2

- The set of n generators of the graph $HS_{(n,n)}$ $\Omega = \{h_i,\ 1 \leqslant i \leqslant n\}$ is closed under inverse; in particular h_i for all i is its own inverse; thus the edges in $HS_{(n,n)}$ are bidirectional.
- For an arbitrary n, $n \geqslant 2$, for any arbitrary node (u,v) of the graph $HS_{(n,n)}$ $\delta(u,v) \neq (u,v)$ where $\delta \in \Omega$; also, for any two $\delta_1, \delta_2 \in \Omega$, $\delta_1(u,v) \neq \delta_2(u,v)$.
- $HS_{(n,n)}$ has $n!$ distinct modules, each module is a star graph S_n; a module of $HS_{(n,n)}$ with module-id x is denoted by $[x,*]$.

Theorem 1. $HS_{(n,n)}$ *is a regular graph of degree n.*

Proof. Consider an arbitrary node (x,y) in $HS_{(n,n)}$. It has exactly $n-1$ local links incident to its $n-1$ local neighbors in the same basic module. It also has exactly one external link, either diameter external link for the node which has same module and local id ($x = y$), or non-diameter external link for the node which has different module and local id ($x \neq y$). Thus, each node has exactly n edges incident on it in a $HS_{(n,n)}$. □

Theorem 2. $HS_{(n,n)}$ *contains* $(n!)^2$ *nodes and* $\frac{n(n!)^2}{2}$ *edges.*

Proof. From definition, $HS_{(n,n)}$ consists of $n!$ basic modules, each of which is a star graph S_n of dimension n. A star graph S_n has $n!$ nodes and hence, $HS_{(n,n)}$ consists of $n! \times n!$ nodes. Using Theorem 1 the number of edges n $HS_{(n,n)}$ is given by $\frac{n(n!)^2}{2}$. □

Remark 3. Throughout the paper we have used n to denote the dimension (order) of the graph; it is also to be noted that each node in $HS_{(n,n)}$ has a node degree n.

2.3. Simple routing and diameter

Since $HS_{(n,n)}$ consists of $n!$ modules, each of which is a star graph S_n of dimension n, we can utilize the shortest routing scheme in a star graph [2] to develop a simple point to point routing scheme in $HS_{(n,n)}$. We start with the following two remarks.

Remark 4. (*Diameter and shortest routing in star graph* S_n) Let u and v be two arbitrary nodes (permutations of n distinct symbols) in S_n and $D(u,v)$ is the distance of the node u from the node v. It is known [2] that $D(u,v) \leqslant \lfloor 3(n-1)/2 \rfloor$, i.e., the diameter of the star graph $\mathscr{D}(S_n) = \lfloor 3(n-1)/2 \rfloor$. Given two arbitrary nodes u and v in S_n, the algorithm to compute the shortest path from u v is also given in [2]. Since the star graph is node-symmetric, in routing between two nodes, the destination node is commonly assumed to have the identity permutation I as its label. The routing between two nodes then is accomplished according to the following two rules [1]:

1. If "a" is the leftmost symbol, move it to any position not occupied by the correct symbol, and
2. If "x" (any symbol other than "a") is the leftmost symbol, move it to its correct position.

Remark 5. Consider two arbitrary nodes (u_s, v_s) and (u_d, v_d) in $HS_{(n,n)}$ where $u_s = u_d$, i.e., the source node belongs to the same module as the destination node. Then, a simple path from (u_s, v_s) to the destination node (u_d, v_d) is computed by the shortest routing scheme in a star graph S_n and the distance between the nodes is always $\leqslant \lfloor 3(n-1)/2 \rfloor$.

Now, consider two arbitrary nodes (u_s, v_s) and (u_d, v_d) in $HS_{(n,n)}$ where $u_s \neq u_d$. The following algorithm Simple_Route computes two simple paths from (u_s, v_s) to the destination node (u_d, v_d) in $HS_{(n,n)}$.

Algorithm Simple_Route
Path P1:

- Use "shortest routing scheme in star S_n" to go from node (u_s, v_s) to (u_s, u_d) in the module with module-id u_s.
- Follow the external link from node (u_s, u_d) to (u_d, u_s) (note that this external link is always a non-diameter link since $u_s \neq u_d$).
- Use "shortest routing scheme in star S_n" to go from node (u_d, u_s) to the destination node (u_d, v_d) in the module with module-id u_d.

Thus the path P1 generated can be expressed as

$$(u_s, v_s) \Rightarrow (u_s, u_d) \hookrightarrow (u_d, u_s) \Rightarrow (u_d, v_d)$$

Path P2:

- Use "shortest routing scheme in star S_n" to go from node (u_s, v_s) to (u_s, v_d) in the module with module-id u_s.
- Follow the external link from node (u_s, v_d) to (v_d, u_s) (assuming $u_s \neq v_d$ this external link is non-diameter; if $u_s = v_d$, this link is not needed).
- Use "shortest routing scheme in star S_n" to go from node (v_d, u_s) to the node (v_d, u_d) in the module with module-id v_d.
- Follow the external link from node (v_d, u_d) to the destination node (u_d, v_d) (assuming $u_d \neq v_d$ this external link is non-diameter; if $u_d = v_d$, this link is not needed).

Thus the path P2 generated can be expressed as

$$(u_s, v_s) \Rightarrow (u_s, v_d) \hookrightarrow (v_d, u_s) \Rightarrow (v_d, u_d) \hookrightarrow (u_d, v_d)$$

Example 2. Consider the source node (abc, abc) and the destination node (bca, cba) in $HS_{(3,3)}$. Algorithm Simple_Route computes the following paths.

$$\begin{aligned} \text{P1}: \quad & (abc, abc) \to (abc, cba) \to (abc, bca) \\ & \hookrightarrow (bca, abc) \to (bca, cba) \\ \text{P2}: \quad & (abc, abc) \to (abc, cba) \hookrightarrow (cba, abc) \\ & \to (cba, cba) \to (cba, bca) \hookrightarrow (bca, cba) \end{aligned}$$

The length of the path P1 is 4, while that of path P2 is 5.

Remark 6. Note that neither of the paths generated by algorithm Simple_Route in Example 2 is optimal since there exists a path of length 3 from node (abc, abc) to node (bca, cba) in $HS_{(3,3)}$ as follows:

$$(abc, abc) \rightsquigarrow (cba, cba) \to (cba, bca) \hookrightarrow (bca, cba)$$

Thus the routing algorithm, although a simple one, is not a shortest routing algorithm.

Theorem 3. *For two arbitrary nodes (u_s, v_s) and (u_d, v_d) in $HS_{(n,n)}$, the algorithm Simple_Route generates a path of length $\leqslant 3n-2$.*

Proof. Consider the two paths generated by the algorithm: path P1 consists of two shortest routes in S_n and an external link while the path P2 consists of two shortest routes in S_n and two external links. It immediately follows from Remark 4 that the length of the shorter path of P1 and P2 is at most $2 \times \lfloor 3(n-1)/2 \rfloor + 1 \leqslant 3n-2$. □

Theorem 4. *The diameter of the hierarchical star graph $HS_{(n,n)}$ is at most $3n-2$.*

Proof. It directly follows from Theorem 3. □

2.4. Comparison with other networks

$HS_{(n,n)}$ of dimension n is an n-regular (regular with node degree n) graph of diameter at most $3n-2$. In this section, we compare the proposed network with existing families of networks with respect to node degree, diameter and cost. The networks with smaller degrees have larger diameters than networks (of comparable number of nodes) with larger node degrees. In order to reflect this trade-off between node degree and diameter in network design, authors in [3,4,7] have traditionally used the concept of *cost* of a network. Cost of a network is defined to be the product of node degree and the diameter of the network and for networks of comparable number of nodes this concept of cost provides a good performance measure of the network design. We compare the various performance metrics of four different graphs, e.g., our proposed $HS_{(n,n)}$, star graphs S_n, folded hypercubes FH_n [7], and hierarchical folded hypercubes $HFN_{(n,n)}$ in Table 1. Detailed numerical comparisons for different sized networks are shown in Table 2 while Figs. 2–4 show the summary comparison of the different families in graphical form in terms of the parameters node

Table 1
Comparison of four different graph families

	$HS_{(n,n)}$	S_n	FH_n	$HFN_{(n,n)}$
Nodes	$(n!)^2$	$n!$	2^n	2^{2n}
Degree	n	$n-1$	$n+1$	$n+2$
Diameter	$2\left\lfloor\frac{3(n-1)}{2}\right\rfloor+1$	$\left\lfloor\frac{3(n-1)}{2}\right\rfloor$	$\left\lceil\frac{n+1}{2}\right\rceil$	$2\left\lceil\frac{n+1}{2}\right\rceil+1$
Cost	$n\times\left(2\left\lfloor\frac{3(n-1)}{2}\right\rfloor+1\right)$	$(n-1)\times\left\lfloor\frac{3(n-1)}{2}\right\rfloor$	$(n+1)\times\left\lceil\frac{n+1}{2}\right\rceil$	$(n+2)\times\left(2\left\lceil\frac{n+1}{2}\right\rceil+1\right)$

Table 2
Detailed numerical comparison

Hierarchical star $HS_{(n,n)}$										
Value of n	4	6	8	10	12	14	16	18	20	22
Nodes	576	5.18×10^{3}	1.63×10^{9}	1.32×10^{13}	2.29×10^{17}	7.6×10^{21}	4.38×10^{26}	4.1×10^{31}	5.92×10^{36}	1.26×10^{42}
Degree	4	6	8	10	12	14	16	18	20	22
Diameter	9	15	21	27	33	39	45	51	57	63
Cost	36	90	168	270	396	546	720	918	1140	1386
Star graph S_n										
Value of n	6	9	13	16	19	22	26	29	33	36
Nodes	720	3.62×10^{5}	6.23×10^{9}	2.09×10^{13}	1.22×10^{17}	1.12×10^{21}	4.03×10^{26}	8.84×10^{30}	8.68×10^{36}	3.72×10^{41}
Degree	5	8	12	15	18	21	25	28	32	35
Diameter	7	12	18	22	27	31	37	42	48	52
Cost	35	96	216	330	486	651	925	1176	1536	1820
Folded hypercube FH_n										
Value of n	9	19	30	43	57	73	88	105	122	140
Nodes	512	5.24×10^{5}	1.07×10^{9}	8.8×10^{12}	1.44×10^{17}	9.94×10^{21}	3.09×10^{26}	4.06×10^{31}	5.32×10^{36}	1.39×10^{42}
Degree	10	20	31	44	58	74	89	106	123	141
Diameter	5	10	16	22	29	37	45	53	62	71
Cost	50	200	496	968	1682	2738	4005	5618	7627	10,011
Hierarchical folded hypercube $HFN_{(n,n)}$										
Value of n	5	9	15	22	29	36	44	52	61	70
Nodes	1024	2.62×10^{5}	1.07×10^{9}	1.76×10^{13}	2.88×10^{17}	4.72×10^{21}	3.09×10^{26}	2.02×10^{31}	5.32×10^{36}	1.39×10^{42}
Degree	7	11	17	24	31	38	46	54	63	72
Diameter	7	11	17	25	31	39	47	55	63	73
Cost	49	121	289	600	961	1482	2162	2970	3969	5256

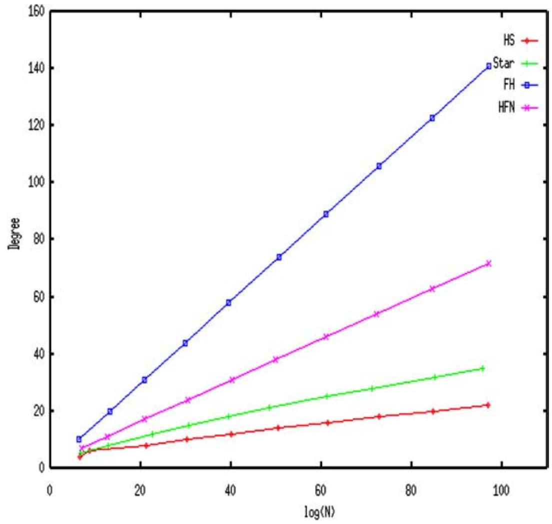

Fig. 2. Comparison of node degrees with size.

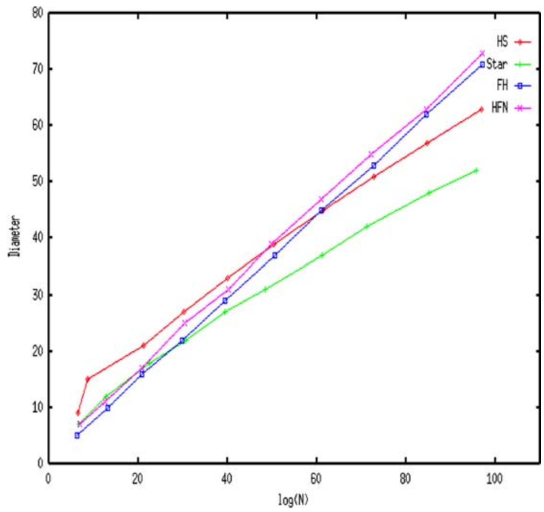

Fig. 3. Comparison of diameters with size.

degree, diameter and cost. We can readily make the following observations.

- For networks of any size, the node degree of the hierarchical star graphs $HS_{(n,n)}$ is always smaller than that of any of the other three networks under consideration and the difference becomes more prominent as the size of the networks grow larger.

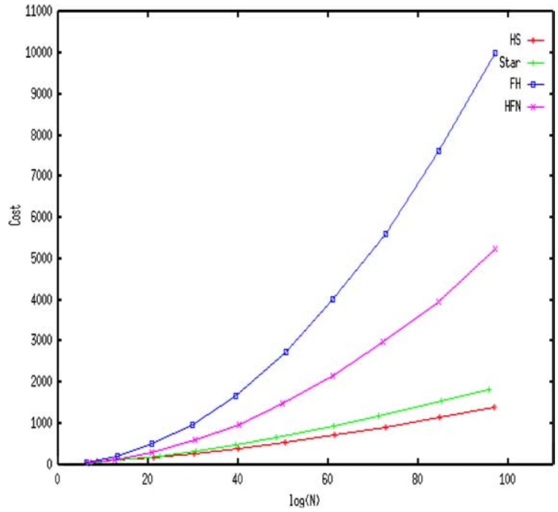

Fig. 4. Comparison of cost with size.

- Hierarchical star graphs $HS_{(n,n)}$ and the star graphs S_n have sub-logarithmic diameter while the folded hypercubes FH_n and the hierarchical folded hypercubes $HFN_{(n,n)}$ have logarithmic diameters. Note that the diameter of $HS_{(n,n)}$ is higher than that of other graphs when the size of the network is relatively small, but as the network size grows, diameter of $HS_{(n,n)}$ becomes smaller than that of FH_n and $HFN_{(n,n)}$ while the diameter of S_n remains always the smallest.
- Cost of $HS_{(n,n)}$ is always the lowest among that of all four networks for networks of all sizes.

3. Embedding in $HS_{(n,n)}$

3.1. Cycles in $HS_{(n,n)}$

First, we note that a star graph S_n of dimension n contains all cycles of even length ℓ, $6 \leqslant \ell \leqslant n!$ [12]. More specifically, we have the following lemma.

Lemma 1. *Given two arbitrary adjacent nodes u and v in S_n ((u,v) is an edge in S_n), we can construct a cycle of length ℓ in S_n containing the edge (u,v) for all even ℓ, $6 \leqslant \ell \leqslant n!$*

Proof. See [12]. □

Corollary 1. *Given two arbitrary adjacent nodes u and v in S_n ((u, v) is an edge in S_n), there exists a path of length $\ell - 1$ between nodes u and v for all even ℓ, $6 \leqslant \ell \leqslant n!$*

Theorem 5. *The hierarchical star graph $HS_{(n,n)}$ contains a Hamiltonian cycle.*

Proof. Consider two arbitrary nodes X_1 and X_2 in a star graph S_n such that $X_1 = \text{swap}_n(X_2)$. By Lemma 1, there exists a Hamiltonian cycle in S_n containing the edge (X_1, X_2) in S_n; number the nodes in this Hamiltonian cycle as $X_1, X_2, \ldots, X_{n!}$; we denote this Hamiltonian by $\Rightarrow_H$. Each module in $HS_{(n,n)}$ is a star graph S_n and hence contains the Hamiltonian $X_1, X_2, \ldots X_{n!}$; also, $HS_{(n,n)}$ has $n!$ modules $[X_i, *]$ each with module-id X_i, $1 \leqslant i \leqslant n!$ We construct the Hamiltonian cycle in $HS_{(n,n)}$ as follows:

$$\begin{aligned}&(X_1,X_3)\hookrightarrow(X_3,X_1)\Rightarrow_H(X_3,X_2)\hookrightarrow(X_2,X_3)\\ &\quad\rightarrow(X_2,X_4)\hookrightarrow(X_4,X_2)\Rightarrow_H(X_4,X_1)\hookrightarrow(X_1,X_4)\\ &\quad\rightarrow(X_1,X_5)\hookrightarrow\cdots,\cdots(X_{n!},X_2)\Rightarrow_H(X_{n!},X_1)\\ &\quad\hookrightarrow(X_1,X_{n!})\rightarrow(X_1,X_1)\rightsquigarrow(X_2,X_2)\rightarrow(X_2,X_1)\\ &\quad\hookrightarrow(X_1,X_2)\rightarrow(X_1,X_3)\end{aligned}$$

We start with the node (X_1, X_3) in module $(X_1, *)$, follow a non-diameter external link to module $(X_3, *)$, traverse all nodes in the module $(X_3, *)$ (by traversing the local Hamiltonian cycle), follow a non-diameter external link to the node (X_2, X_3) in module $(X_2, *)$ and continue the pattern. This path always comes back to the module $[X_1, *]$ or $[X_2, *]$ after traversing all nodes in the module X_i and then goes to traverse nodes in module $[X_{i+1}, *]$, $3 \leqslant i \leqslant n!$ The remaining four nodes (X_1, X_1), (X_1, X_2), (X_2, X_1), and (X_2, X_2), are visited after all the other nodes in $HSN_{(n,n)}$ are visited. Note that there are two external links between modules $[X_1, *]$ and $[X_2, *]$; one is a *non-diameter external link* between nodes (X_1, X_2) and (X_2, X_1), the other is a *diameter external link* between nodes (X_1, X_1) and (X_2, X_2). The above path starts from (X_1, X_3) and ends at (X_1, X_3) and travels all nodes in $HS_{(n,n)}$ exactly once. □

Example 3. Fig. 5 shows one Hamiltonian in $HS_{(3,3)}$ constructed along the line of the proof of the above theorem. In this example, $n = 3$, and $X_1 = abc$ and $X_2 = cba = \text{swap}_3(abc)$. Also note that the Hamiltonian $\Rightarrow_H$ in a basic module is $(abc, cba, bca, acb, cab, bac)$ as well as there are six modules, each a star graph of dimension 3.

Lemma 2. *For any even ℓ, $6 \leqslant \ell \leqslant n!$, the hierarchical star graph $HS_{(n,n)}$, $n \geqslant 3$, contains $n!$ mutually pairwise disjoint cycles of length ℓ.*

Proof. The graph $HS_{(n,n)}$ contains $n!$ mutually disjoint modules, each of which is a star graph S_n of dimension n. This, coupled with Lemma 1, yields the desired result. □

Lemma 3. *For any even ℓ, $12 \leqslant \ell \leqslant 4 * n!$, there exists a cycle of length ℓ in $HS_{(n,n)}$, for $n \geqslant 3$ (i.e., $HS_{(n,n)}$ can embed the ring of length ℓ with dilation 1 and link congestion 1).*

Proof. Consider four arbitrary nodes X_1, X_2, X_3, and X_4 in a star graph S_n such that X_1 and X_3 are neighbor nodes and X_2 and X_4 are neighbor nodes. We can immediately get a cycle of length 8 in $HS_{(n,n)}$ as follows.

$$\begin{aligned}(X_1,X_4)&\rightarrow(X_1,X_2)\mapsto(X_2,X_1)\\ &\rightarrow(X_2,X_3)\mapsto(X_3,X_2)\\ &\rightarrow(X_3,X_4)\mapsto(X_4,X_3)\\ &\rightarrow(X_4,X_1)\mapsto(X_1,X_4)\end{aligned}$$

This cycle involves nodes in four modules $[X_1, *]$, $[X_2, *]$, $[X_3, *]$, $[X_4, *]$ and it consists of four local links and four external links. From Corollary 1, each local link in the above cycle can be independently substituted by a simple path of length $\ell - 1$ for any even ℓ, $6 \leqslant \ell \leqslant n!$; Thus, for any even ℓ, if we can solve the following equation

$$x_1 + x_2 + x_3 + x_4 = \ell, \quad x_i = 2, 6, 8, \cdots, n! \qquad (1)$$

then we get a cycle of length ℓ. It can be easily shown that the above equation has at least one set of solution when ℓ is even and $12 \leqslant \ell \leqslant 4 * n!$, and $n \geqslant 3$. □

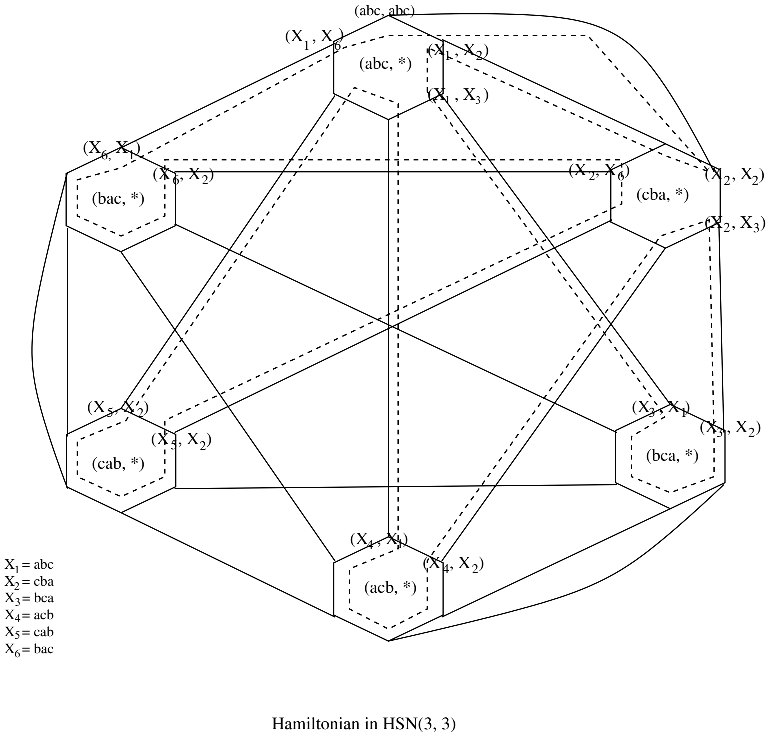

Fig. 5. Hamiltonian in $HS_{(3,3)}$.

Example 4. Consider a $HS_{(3,3)}$ in which we try to find a cycle of length 12. From Eq. (1), we can choose $x_1 = 6$ and $x_2 = x_3 = x_4 = 2$. So, the cycle could be $(abc, acb) \rightarrow (abc, cab) \rightarrow (abc, bac) \rightarrow (abc, abc) \rightarrow (abc, cba) \rightarrow (abc, bca) \mapsto (bca, abc) \rightarrow (bca, cba) \mapsto (cba, bca) \rightarrow (cba, acb) \mapsto (acb, cba) \rightarrow (acb, abc) \mapsto (abc, acb)$.

If we are trying to find a cycle of length 20, we can choose $x_1 = x_2 = x_3 = 6$ and $x_4 = 2$. The cycle can be constructed accordingly.

Lemma 4. For any even ℓ, $4 * n! \leqslant \ell \leqslant (n!)^2$, there exists a cycle of length ℓ in $HS_{(n,n)}$, for $n \geqslant 3$.

Proof. Consider the Hamiltonian cycle in $HS_{(n,n)}$ we constructed in the proof of Theorem 5:

$$(X_1, X_3) \hookrightarrow (X_3, X_1) \Rightarrow_H (X_3, X_2) \hookrightarrow (X_2, X_3) \rightarrow (X_2, X_4) \hookrightarrow (X_4, X_2) \Rightarrow_H (X_4, X_1) \hookrightarrow (X_1, X_4) \rightarrow (X_1, X_5) \hookrightarrow \cdots, \cdots (X_{n!}, X_2) \Rightarrow_H (X_{n!}, X_1) \hookrightarrow (X_1, X_{n!}) \rightarrow (X_1, X_1) \rightsquigarrow (X_2, X_2) \rightarrow (X_2, X_1) \hookrightarrow (X_1, X_2) \rightarrow (X_1, X_3)$$

This cycle visits all nodes in module $[X_i, *]$, $3 \leqslant i \leqslant n!$ along the Hamiltonian path of a star graph S_n. Similarly, as we did in proving Lemma 3, we can substitute each of these Hamiltonian paths independently by a path of even length 2 or 6 to $n!$ In addition, incorporating all the $2 \times n!$ nodes in modules $[X_1, *]$ and $[X_2, *]$, we get all even cycles of length between $4 \times n!$ and $n! \times n!$ □

Theorem 6. *For any even ℓ, $6 \leqslant \ell \leqslant (n!)^2$, there exists a cycle of length ℓ in $HS_{(n,n)}$, where $n \geqslant 3$.*

Proof. Combining Lemmas 2–4, the proof readily follows. □

3.2. 2D Mesh in $HS_{(n,n)}$

We prove that $HS_{(n,n)}$ embeds the largest possible meshes with dilation 3 and link congestion 4. An embedding of networks is an one-to-one mapping φ from node set of source network to the node set of destination network. Thus, a link in the source network is mapped to one or a group of links in destination network. The dilation of a mapping φ is defined as the maximal distance between $\varphi(\mu)$ and $\varphi(v)$ for any two nodes μ, v in the source network. For each link e in destination network, we use $c(e)$ to denote the number of links in source network whose corresponding path in target network contains e. The *link congestion* of a mapping φ is defined as the maximal value of $c(e)$ for all edges e in the target network.

Theorem 7. *A $n! \times n!$ 2D mesh can be embedded in $HS_{(n,n)}$ with dilation 3 and link congestion 4.*

Proof. Consider a Hamiltonian path in a star graph S_n; number the $n!$ nodes on this path as $X_1, X_2, \ldots, X_{n!}$; we view the nodes of $HS_{(n,n)}$ as (X_i, X_j), where $1 \leqslant i,j \leqslant n!$ (note that nodes (X_i, X_j) and (X_i, X_{j+1}) in $HS_{(n,n)}$, $1 \leqslant i \leqslant n!$, $1 \leqslant j < n!$, is connected by a local link). Now consider a 2D mesh with $n!$ rows and $n!$ columns and denote by $M(i,j)$ the node on the i-th row and j-th column of this mesh. We map the node $M(i,j)$ of the mesh onto the node (X_i, X_j) of an $HS_{(n,n)}$ for $1 \leqslant i,j \leqslant n!$

There are two kinds of edges in a mesh. For each edge $(M(i,j), M(i,j+1))$ in the 2D mesh, there exists a direct edge between the nodes (X_i, X_j) and (X_i, X_{j+1}) in $HS_{(n,n)}$, $1 \leqslant i \leqslant n!$, $1 \leqslant j < n!$ To simulate the edge between $M(i,j)$ and $M(i+1,j)$ in the mesh, $1 \leqslant i < n!$, $1 \leqslant j \leqslant n!$, the path between nodes (X_i, X_j) and (X_i, X_{j+1}) in $HS_{(n,n)}$ can be computed as follows.

- *Case* $[i = j]$: $(X_i, X_j) \rightarrow (X_i, X_{j+1}) \hookrightarrow (X_{j+1}, X_i)$.
- *Case* $[i + 1 = j]$: $(X_i, X_j) \hookrightarrow (X_j, X_i) \rightarrow (X_j, X_{i+1})$.
- *Case* $[i \neq j \wedge j \neq i + 1]$: $(X_i, X_j) \hookrightarrow (X_j, X_i) \rightarrow (X_j, X_{i+1}) \hookrightarrow (X_{i+1}, X_j)$.

Therefore, $HS_{(n,n)}$ embeds a $n! \times n!$ mesh with dilation 3.

To compute the congestion we first note that only the internal links and the non-diameter external links in $HS_{(n,n)}$ are used in this embedding of meshes. (1) Consider a local link that connects (X_i, X_j) and (X_i, X_{j+1}). For any edge e of this type, there are exactly two edges in the guest network (e.g, the mesh) whose corresponding paths contains e: the edge between $(M(i,j)$ and $M(i,j+1))$ and the other between $M(j,i)$ and $M(j+1,i)$; (2) Consider a non-diameter external link. Any edge e of this type in $HS_{(n,n)}$ is used at most four times as the four links $(M(i,j), M(i,j+1))$, $(M(i,j), M(i+1,j))$, $(M(j,i), M(j,i+1))$, and $(M(j,i), M(j+1,i))$ in the mesh are mapped to the paths in $HS_{(n,n)}$. So, for any e in $HSN_{(n,n)}$, $c(e)$ does not exceed four. Hence, $HS_{(n,n)}$ embeds a $n! \times n!$ mesh with congestion 4. □

From Lemma 4 and Theorem 7, we can at once conclude the following theorem (similar to that in [4]):

Theorem 8. *Any algorithm that executes on a ring of even length ℓ, $4 * n! \leqslant \ell \leqslant (n!)^2$, or a 2D mesh of size $n! \times n!$ using $\mathscr{F}(n)$ time steps, will also execute on an $HSN_{(n,n)}$ in at most $c\mathscr{F}(n)$ time steps, where c is a constant; asymptotic complexity of the algorithm will remain the same.*

4. Fault tolerance of $HS_{(n,n)}$

The node fault tolerance of an undirected graph is measured by the vertex connectivity of the graph. A graph G is said to have a vertex connectivity ξ if the graph G remains connected when an arbitrary set of less than ξ nodes are faulty (i.e., in the fault free graph there are ξ many node disjoint paths between any two arbitrary nodes). Obviously, the vertex connectivity of a graph G cannot exceed the minimum degree of a node in G. A graph is called *maximally fault tolerant* if vertex connectivity of the graph equals the minimum degree of a node. We know that the vertex connectivity of a star graph S_n is $n - 1$ [2]; since S_n is $(n - 1)$-regular, the star graphs are maximally fault tolerant; authors in [20] establish that the fault diameter of a star graph S_n is $\lfloor 3(n-1)/2 \rfloor + 2$ and they provide algorithms to compute the $(n-1)$ node-disjoint paths in a star graph S_n given an arbitrary source node and $(n-1)$ distinct arbitrary destination nodes. Our purpose in this section is to show that the proposed graph $\mathrm{HS}_{(n,n)}$ has a vertex connectivity of n and hence these graphs are maximally fault tolerant.

Theorem 9. *Between any two arbitrary nodes* (X_s, Y_s) *and* (X_d, Y_d) *in* $\mathrm{HS}_{(n,n)}$ *there exist n node disjoint paths.*

Proof. Consider two arbitrary nodes (X_s, Y_s) and (X_d, Y_d) in $\mathrm{HS}_{(n,n)}$. We need to consider three cases:

Case 1: $[X_s = X_d]$ Both the source node and the destination node are in the same module $[X_s, *]$. The module $[X_s, *]$ is a star graph S_n of dimension n and hence there exist $n-1$ node disjoint paths between the source and the destination node—each node belonging to all of these paths are in the module $[X_s, *]$. To get the last (n-th) node disjoint path, we consider three sub-cases:

Sub-case (a): $[X_s = Y_s \wedge X_s \neq Y_d]$ The path is given by

$$(X_s, X_s) \rightsquigarrow (\widehat{X}_s, \widehat{X}_s) \Rightarrow (\widehat{X}_s, Y_d) \hookrightarrow (Y_d, \widehat{X}_s) \rightarrow (Y_d, X_s) \hookrightarrow (X_s, Y_d)$$

Sub-case (b): $[X_s = Y_d \wedge X_s \neq Y_s]$ The path is given by

$$(X_s, Y_s) \hookrightarrow (Y_s, X_s) \rightarrow (Y_s, \widehat{X}_s) \hookrightarrow (\widehat{X}_s, Y_s) \Rightarrow (\widehat{X}_s, \widehat{X}_s) \rightsquigarrow (X_s, X_s)$$

Note: In case $Y_s = \widehat{X}_s$, the second external link and the adjacent group of local links are not needed.

Sub-case (c): $[X_s \neq Y_d \wedge X_s \neq Y_s]$ The path is given by

$$(X_s, Y_s) \hookrightarrow (Y_s, X_s) \Rightarrow (Y_s, Y_d) \hookrightarrow (Y_d, Y_s) \Rightarrow (Y_d, X_s) \hookrightarrow (X_s, Y_d)$$

This last path, in either of the three sub-cases, does not contain any node from the module $[X_s, *]$ except the source and the destination nodes; so, this path is node-disjoint from the earlier $(n-1)$ node disjoint paths.

Case 2: $[X_s \neq X_d]$ In this case, the source node and the destination node belongs to different modules $[X_s, *]$ and $[X_d, *]$. Each module is a star graph S_n which has a node connectivity $n-1$. Choose an arbitrary set of $(n-1)$ nodes $\{X_j, 1 \leqslant j \leqslant n-1\}$ in an S_n such that $X_i \notin \{X_s, Y_s, X_d, \hat{X}_s, \hat{Y}_s\}$, $1 \leqslant j \leqslant n-1$—this is always possible for $n > 3$. In an S_n, we can construct node disjoint paths from node Y_s (or from node Y_s) to the $(n-1)$ nodes $\{X_j, 1 \leqslant j \leqslant n-1\}$. Consider the following $(n-1)$ paths from (X_s, Y_s) to (X_d, Y_d), each corresponding to one $\{X_j, 1 \leqslant j \leqslant n-1\}$.

$$(X_s, Y_s) \Rightarrow (X_s, X_j) \hookrightarrow (X_j, X_s) \Rightarrow (X_i, X_d) \hookrightarrow (X_d, X_j) \Rightarrow (X_d, Y_d)$$

These $(n-1)$ paths are node disjoint except the source and the destination nodes. Note that any intermediate node in any of these paths are from only the modules $[X_s, *]$, $[X_d, *]$ and $[X_j, *]$, $1 \leqslant j \leqslant n-1$. To get the last ($n$-th) node disjoint path, we consider four sub-cases:

Sub-case (a): $[X_s = Y_s \wedge X_d \neq Y_d]$ The path is given by

$$(X_s, X_s) \rightsquigarrow (\widehat{X}_s, \widehat{X}_s) \Rightarrow (\widehat{X}_s, Y_d) \hookrightarrow (Y_d, \widehat{X}_s) \Rightarrow (Y_d, X_d) \hookrightarrow (X_d, Y_d)$$

Sub-case (b): $[X_s = Y_s \wedge X_d = Y_d]$ The path is given by

$$(X_s, X_s) \rightsquigarrow (\widehat{X}_s, \widehat{X}_s) \Rightarrow (\widehat{X}_s, \hat{Y}_d) \hookrightarrow (\hat{Y}_d, \widehat{X}_s) \Rightarrow (\hat{Y}_d, \hat{Y}_d) \rightsquigarrow (Y_d, Y_d)$$

Sub-case (c): $[X_s \neq Y_s \wedge X_d \neq Y_d]$ The path is given by

$$(X_s, Y_s) \hookrightarrow (Y_s, X_s) \Rightarrow (Y_s, Y_d) \hookrightarrow (Y_d, Y_s) \Rightarrow (Y_d, X_d) \hookrightarrow (X_d, Y_d)$$

Sub-case (d): $[X_s \neq Y_s \wedge X_d = Y_d]$ The path is given by

$$(X_s, Y_s) \hookrightarrow (Y_s, X_s) \Rightarrow (Y_s, \hat{Y}_d) \hookrightarrow (\hat{Y}_d, Y_s) \Rightarrow (\hat{Y}_d, \hat{Y}_d) \hookrightarrow (Y_d, Y_d)$$

This last path, in either of the four sub-cases, does not contain any node from the modules $[X_s, *]$, $[X_d, *]$ or $[X_j, *]$, $1 \leqslant j \leqslant n-1$ except the source and the destination nodes; so, this path is node-disjoint from the earlier $(n-1)$ node disjoint paths. □

Corollary 2. *The hierarchical star graph* $HS_{(n,n)}$ *of dimension n has a vertex connectivity of n and hence it is maximally fault tolerant.*

Remark 7. The constructive proof for the theorem on vertex connectivity (Theorem 9) readily suggests an optimal routing scheme in the network in the presence of maximal number of allowable faults (such that the system is not disconnected).

5. One-to-all broadcast in $HS_{(n,n)}$

One-to-all broadcast is very important to algorithm design on any network [11]; this is a frequently used communication pattern in which a message (or a data set) is transmitted from a source node to all other nodes in the network. This one-to-all broadcast is often a necessary step in designing parallel and distributed algorithms on networks. In what follows, we assume that all links in the network are bidirectional so that the node on either side of the link can send messages to the node on the other side. In addition, we assume that at any time, one node can only communicate with at most one other node which is adjacent to it. That is, we use the one port model of communication as was used in [7] unlike the all-port model of communication [11].

Remark 8. In one-port communication model, any one-to-all broadcast algorithm for a network of N nodes has a lower bound of $\mathcal{O}(\log N)$ on the broadcast time steps; this easily follows from the fact that the number of informed nodes at the end of any step can increase by a factor of 2 over that in the previous step.

Thus, any broadcast algorithm which uses $\mathcal{O}(\log N)$ time for any network of N nodes, is *asymptotically optimal*. Our purpose is to design an optimal broadcast algorithm for our proposed network $HS_{(n,n)}$. We use the optimal broadcast algorithm of [15] for a star graph S_n of dimension n that uses $\mathcal{O}(\log(n!)) = \mathcal{O}(n \log n)$ time.

Optimal broadcast in star graph S_n: The algorithm can broadcast a message to $n!$ processors in S_n using $\mathcal{O}(\log(n!)) = \mathcal{O}(n\log(n))$ time. The algorithm is based on the hierarchical structure of a star graph. An S_n can be divided into $n-1$ many sub-stars S_{n-1} each of dimension $n-1$; the sub-stars can be further subdivided and so on. The broadcasting algorithm has $n-1$ recursive steps. The first step distributes the message from the source node to all $(n-1)$ sub-stars S_{n-1}. In the subsequent k-th step, $2 \leqslant k < n-1$, there are $\prod_{i=n-k+1}^{n} i$ many parallel instances of the broadcast algorithm running on $\prod_{i=n-k+1}^{n} i$ many disjoint sub-stars S_{n-k}. There are two logical phases in each step of the algorithm. In phase 1, the message is sent to a sequence of nodes in such a way that each node (permutation of symbols) in the sequence would have a distinct symbol in the first position. Since such a sequence can be found such that all nodes in were embedded in a binary tree, the time required in phase 1 is $\mathcal{O}(\log(n))$. In phase 2, each node which received the message in phase 1 sends the message to its neighboring node obtained by swapping the first and last symbols in the node permutation. See [15] for details of the algorithm and its correctness proof. Call this algorithm *Broadcast_Star*.

We design an optimal broadcast algorithm for our proposed hierarchical star graph $HS_{(n,n)}$ by using the above algorithm Broadcast_Star as a subroutine. Consider an arbitrary node (x, y) in $HS_{(n,n)}$; the following algorithm *Broadcast_HS* broadcasts a message to all other nodes in $HS_{(n,n)}$.

Algorithm Broadcast_HS (x, y)

Step 1: Consider the module $[x, *]$ given the source node (x, y); use algorithm Broadcast_Star to transmit the message to all nodes in this basic module. This step takes $\mathcal{O}(n \log n)$ time.

Step 2: Each node (x, v) in the module $[x, *]$ transmits the message to node (v, x) via an external link in one unit time. Since each node has one external link and all these external links lead to distinct modules, each module in $HS_{(n,n)}$ has at least one informed node after this step.
Step 3: The informed node in each module $[v, *]$, $v \neq x$, broadcast the message in the module $[v, *]$ using the algorithm Broadcast_Star. For different modules, this broadcast is done concurrently. Thus, this step takes $\mathcal{O}(n \log n)$ time. After this step, all nodes in $HS_{(n,n)}$ have received the message.

Theorem 10. *The algorithm Broadcast_HS is an optimal one-to-all broadcast procedure for the graph* $HS_{(n,n)}$.

Proof. The three steps in the above algorithm take $\mathcal{O}(n \log n)$, $\mathcal{O}(1)$ and $\mathcal{O}(n \log(n))$ time respectively. Thus, the entire algorithm takes $\mathcal{O}(n \log(n))$ time. The number of nodes in $HS_{(n,n)}$ is $N = (n!)^2$ and hence the algorithm is optimal taking $\mathcal{O}(N)$ time. □

6. Conclusion

We have proposed a new two level hierarchical network using the well known star graphs as building blocks and compared its topological properties with the networks in the same category. We have shown that the proposed hierarchical star graphs $HS_{(n,n)}$ are superior to star graphs, folded hypercubes and the hierarchical folded hypercubes in terms of node degree, diameter and cost of the network. Specifically, we showed the following.

1. For networks of any size, the node degree of the hierarchical star graphs $HS_{(n,n)}$ is always smaller than that of any of the other three networks under consideration and the difference becomes more prominent as the size of the networks grow larger.
2. Cost of $HS_{(n,n)}$ is always the lowest among that of all four networks for networks of all sizes.
3. Hierarchical star graphs $HS_{(n,n)}$ and the star graphs S_n have sub-logarithmic diameter while the folded hypercubes FH_n and the hierarchical folded hypercubes $HFN_{(n,n)}$ have logarithmic diameters.

We have proposed simple routing in the network, showed the network is optimally fault tolerant as well as proposed an optimal broadcast algorithm. The proposed family of networks is interesting in its own terms and adds to the already established attractiveness of the star graphs as compared to hypercubes.

References

[1] S.B. Akers, B. Krishnamurthy, The star graph: an attractive alternative to *n*-cube, in: Proceedings of the International Conference on Parallel Processing (ICPP-87), St. Charles, Illinois, August 1987, pp. 393–400.
[2] S.B. Akers, B. Krishnamurthy, A group-theoretic model for symmetric interconnection networks, IEEE Transactions on Computers 38 (4) (1989) 555–566.
[3] L. Bhuyan, D.P. Agrawal, Generalized hypercube and hyperbus structure for a computer network, IEEE Transactions on Computers 33 (3) (1984) 323–333.
[4] D.R. Duh, G.H. Chen, J.F. Fang, Algorithms and properties of a new two level network with folded hypercubes as basic modules, IEEE Transactions on Parallel and Distributed Systems 6 (7) (1995) 714–723.
[5] K. Day, A. Tripathi, A comparative study of topological properties of hypercubes and star graphs, Technical Report TR 91–10, Computer Science Department, University of Minnesota, Minneapolis, MN, May 1991.
[6] K. Day, A. Tripathi, A comparative study of topological properties of hypercubes and star graphs, IEEE Transactions on Parallel and Distributed Systems 5 (1) (1994) 31–38.
[7] A. El-Amawy, S. Latifi, Properties and performance of folded hypercubes, IEEE Transactions on Parallel and Distributed Systems 2 (2) (1991) 31–42.
[8] P. Fragopoulou, S.G. Akl, Parallel algorithm for computing Fourier transforms on the star graph, in: Proceedings of the International Conference on Parallel Processing, St. Charles, Illinois, vol. III, 1991, pp. 100–106.
[9] K. Ghose, K.R. Desai, The HCN: a versatile interconnection network based on cubes, in: Proceedings of ACM International Conference on Supercomputing, 1989, pp. 426–435.
[10] K. Ghose, K.R. Desai, The design and evaluation of the hierarchical cubic network, in: Proceedings of International Conference on Parallel Processing, vol. 1, 1990, pp. 355–362.

[11] S.L. Johnsson, C.T. Ho, Optimum broadcasting and personalized communication in hypercubes, IEEE Transactions on Computers 38 (9) (1989) 1249–1268.
[12] J.S. Jwo, S. Lakshmivarahan, S.K. Dhall, Embedding of cycles and grids in star graphs, Journal of Circuits, Systems and Computers 1 (1) (1991) 43–74.
[13] Y. Li, S. Peng, W. Chu, Metacube: a new interconnection network for large scale parallel systems, in: Proceedings of the 7th Asia-Pacific Conference on Computer Systems Architecture, Melbourne, Victoria, Australia, 2002, pp. 29–36.
[14] A. Menn, A.K. Somani, An efficient sorting algorithm for the star graph interconnection network, in: Proceedings of the International Conference on Parallel Processing, St. Charles, Illinois, 1990, vol. III, pp. 1–8.
[15] V.E. Mendia, D. Sarkar, Optimal broadcasting on the star graph, IEEE Transactions on Parallel and Distributed Systems 3 (4) (1992) 389–396.
[16] M. Nigam, S. Sahni, B. Krishnamurthy, Embedding Hamiltonians and hypercubes in star interconnection graphs, in: Proceedings of the International Conference on Parallel Processing, August 1990, pp. 340–343.
[17] K. Qiu, S.G. Akl, H. Meijer, On some properties and algorithms for the star and pancake interconnection networks, Journal of Parallel and Distributed Computing 22 (1) (1994).
[18] K. Qiu, H. Meijer, S.G. Akl, Decomposing a star graph into disjoint cycles, Information Processing Letters 39 (3) (1991) 125–129.
[19] K. Qiu, H. Meijer, S.G. Akl, On the cycle structure of star graphs, Technical Report 92-341, Department of Computer Science, Queen's University, Ontario, Canada, November 1992.
[20] Y. Rouskov, P.K. Srimani, Fault diameter of star graphs, Information Processing Letters 48 (5) (1993).

Wei Shi graduated with a MS in computer science from Colorado State university in 1998; currently he lives at 1544 Ivy Place, Superior, Colorado.

Pradip K. Srimani is a Professor and Chair of Computer Science at Clemson University. He has previously served the faculty of India Statistical Institute, Calcutta, Gesselschaft fuer Mathematik und Datenverarbeitung, Bonn, West Germany, Indian Institute of Management, Calcutta, India and Southern Illinois University, Carbondale, Illinois, Colorado State University, Ft. Collins, Colorado, and Technical University of Compiegne, France. He was the Editor-in-Chief of IEEE Computer Society Press during 1996–2000 and an associate Editor of IEEE Transaction on Data and Knowledge Engineering and Journal of Parallel Computing. His research interests include mobile computing, distributed computing, parallel algorithms, networks and graph theory applications. He is a Fellow of IEEE and a member of ACM.