BULGARIAN ACADEMY OF SCIENCES

CYBERNETICS AND INFORMATION TECHNOLOGIES • Volume **21**, No 1
Sofia • 2021 Print ISSN: 1311-9702; Online ISSN: 1314-4081
DOI: 10.2478/cait-2021-0003

Hierarchical Hexagon: A New Fault-Tolerant Interconnection Network for Parallel Systems

Laxminath Tripathy[1], *Chita Ranjan Tripathy*[2]

[1]*Department of Computer Science and Engineering, ITER, Siksha 'O' Anusandhan University, Bhubaneswar, India*
[2]*Department of Computer Science and Engineering, VSS University of Technology, Burla 768018, India*
E-mails: laxmintripathy@gmail.com crt.vssut@yahoo.com

Abstract**:** *A new interconnection network topology called Hierarchical Hexagon HH*(n) *is proposed for massively parallel systems. The new network uses a hexagon as the primary building block and grows hierarchically. Our proposed network is shown to be superior to the star based and the hypercube networks, with respect to node degree, diameter, network cost, and fault tolerance. We thoroughly analyze different topological parameters of the proposed topology including fault tolerance routing and embedding Hamiltonian cycle.*

Keywords**:** *Interconnection network, cost, bisection width, fault-tolerant routing, packing density, Hamiltonian cycle.*

1. Introduction

An interconnection network topology is denoted by a graph where the processing elements are represented by vertices and the bidirectional communication channels are represented by edges. It is designed with low diameter, small degree, minimum cost, high packing density, and high fault tolerance capability. Most of the parallel systems use Hypercube as the basic building block for interconnection which has been studied extensively. A better alternative to the Hypercube is the Star graph [1]. The Star graphs possess most of the desirable properties of Hypercube [2-4]. Many other interconnection topologies have also been proposed in the past which use some known networks as the building blocks [5-7]. Adhikari and Tripathy [8] in 2012 proposed a network called Mstar which is based upon Metacubes [9]. It has been shown that the performance of Mstar is better than Star graph, Starcube [10], Metacube in terms of size, degree, cost, diameter, and average distance. Shi and Srimani [11] in 2005 proposed Hierarchical Star and claimed that it is superior to Star, Folded Hypercube, and Hierarchical Folded Hypercube [12].

In recent years, the Hierarchical Interconnection Networks (HIN) have attracted increasing attention of the researchers. This is because they provide a framework to design networks with reduced link cost. The said networks employ multiple levels in which the lower-level networks are used to provide local communication and the higher-level networks are used to facilitate remote communication. A number of hierarchical interconnection networks have been proposed in the literature [13-17]. However, for getting better fault-tolerance in hierarchical interconnection network the hexagon can be considered as the basic building block.

In this paper, a new fault-tolerant interconnection network called the Hierarchical Hexagon HH(n) with degree n is proposed. We thoroughly investigate the topological properties, routing, fault-tolerance, and the performance of the proposed network HH(n). It is shown that the proposed network is better than Star, Hierarchical Star, Mstar, Folded Hypercubes, Hierarchical Folded Hypercube HFN(n, n), Dragonfly [18], 3-D Torus, Tofu [19] (Ajima, Hypercube [20], and Fat tree [21].

The rest part of the present paper is organized as follows: In Section 2, the brief description of the existing topologies is presented. The newly-proposed network HH(n) is discussed in Section 3. Its topological properties are discussed in Section 4. The detail routing technique in both fault-free and faulty situations are elaborated and illustrated in Section 5. The performance comparison is carried out in Section 6. The presence of a Hamiltonian cycle in the proposed system is proved in Section 7. The concluding remarks are presented in Section 8.

2. Related works

Trobec Roman et al. [22] in 2017 surveyed that most of the supercomputers use Fat Tree, Dragonfly, K Computer [23], Tofu (5D Torus), and 3-D Torus as interconnection networks. The Dragonfly topology is a highly scalable and high-performance network. It also provides fully interconnection patterns. The Dragonfly is used as the interconnection topology in many supercomputers. Fuentes et al. [24] in 2012 have proved that the concentrated torus has a better option than the dragonfly network for a limited number of nodes. In the same work, the authors also have shown the cost of the concentrated torus to be lower with better fault tolerance than the dragonfly network.

Another highly scalable interconnection topology is the Tofu which has some similarity with Dragonfly topology. But, the diameter of Dragonfly is less than Tofu, 3-D torus, and Hypercube. For scaling up the network to a large extent, the length of the cable becomes a promising challenge. This concern is resolved by introducing optical cable. Since the uses of long optical cables dictate the cost. However, this packaging can be resolved by making a balance between copper for short links and optics for longer links [18].

In many topologies, high radix (degree) routers are used to reduce the network diameter, as it becomes the major bottleneck of the communication. In dragonfly topology and concentrated torus, approximately 75 and 80 percent of the total links are connected to a router, respectively [24]. In the case of any router fault, the most part of links and computing nodes become disabled. The performance of Dragonfly may drop significantly due to the negative effects derived from congestion situations. The worst of them is the Head-of-Line (HoL) blocking effect, which appears when a packet at the head of a buffer blocks the rest of the packets, even if they request access to the available ports.

A Hierarchical interconnection network using only two-levels was originally proposed by Dandamudi and Eager [14] in 1990. Duh, Chen and Fang [12] in 1995 proposed a two-level topology called a Hierarchical Folded Hypercube which is the variation of a Folded Hypercube. Ghose and Desai [7] 1989 proposed a Hierarchical Cube Network consisting of a $2n$ number of basic modules in which each one is a Hypercube of dimension n. It has been proved that Hierarchical Cube is superior to Hypercube. The Hierarchical Interconnection Networks (HIN's) become a better choice when a large number of processors are to be connected in a parallel system. There are two main motivations behind the use of hierarchical topology. First, for very large systems, the number of links needed with conventional networks may become prohibitively large. Hence, the future systems need to minimize the number of links to reduce the hardware complexity and cost. The Hierarchical Interconnection Networks exploit the locality that exists in communication patterns to allow a reduction in the required number of links. Second, the communication overhead introduced by the interconnection networks for large systems can be reduced by matching the structure of the problem to the communication structure of the system (i.e., network topology). This matching is achieved effectively in a Hierarchical Interconnection Network.

3. The proposed interconnection network: The Hierarchical Hexagon (HH(n))

In this section, we propose a new interconnection network called the Hierarchical Hexagon (HH). The proposed network is built upon Hexagon as the basic building block. It consists of a number of hexagons depending upon the degree of the node. The minimum degree is assumed to be 2. So, the total number of nodes in the proposed topology is a power of 6. A HH with the degree n=3 (HH(3)) is shown in Fig. 1.

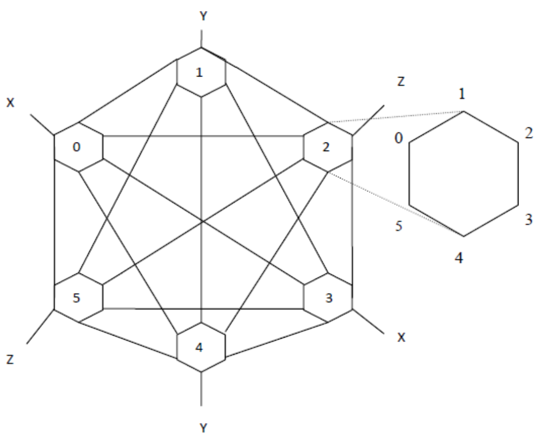

Fig. 1. Hierarchical Hexagon HH(3)

The modular layout diagram of the HH(4) without connection is shown in Fig. 2. Here, each module has degree 3 and it consists of a total of 36 such modules. Each module can be connected to the rest of the modules in order to make its degree 4. A Hierarchical Hexagon Network HH(n) of dimension n consists of a number of modules (each module is a hexagonal graph of dimension n) interconnected by edges. Each node of HH(n) is represented by two parameters (m, n), where m is the module number and n represents the node number within the module.

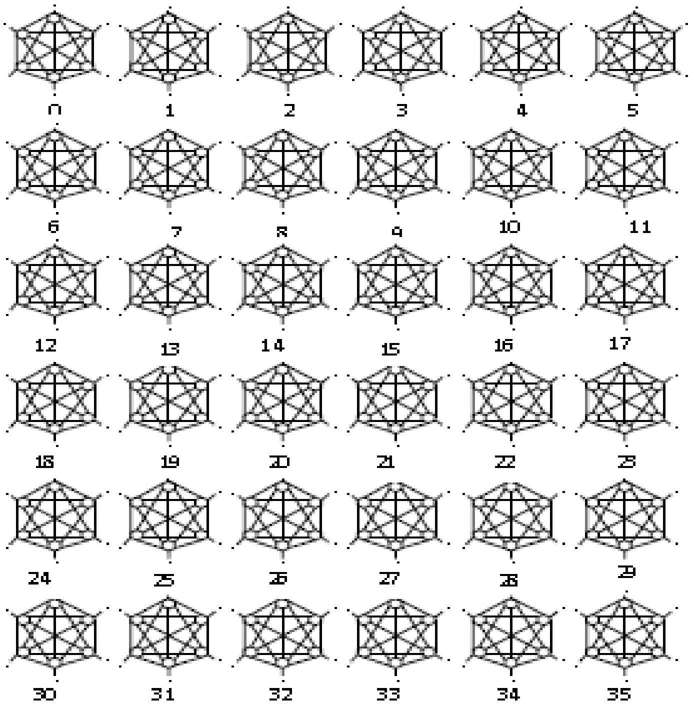

Fig. 2. Module layout of an HH(4)

4. Topological properties of the HH(n)

Definition 1. A HH(n) is represented by the interconnection of hexagons with degree n where $n \geq 2$ and two nodes are connected by an edge if any one of the following conditions is satisfied.

(a) The node1(m, n) for each m, there exists an edge from node position n to $n + 1$ mod 6 and $n - 1$ mod 6.

(b) If $m \neq n$, there exists an edge between node1(m, n) to nod2(n, m), where $n <$ total number of nodes in one module and $m<$ total number of modules in the network.

(c) If m=n, there exists an external edge /link between node1(m, m) to node2(m', m') where $|m - m'|$ = total module/2.

Theorem 1. A HH(n) consists of $6^{2^{n-2}}$ number of nodes, where $n \geq 2$.

Proof: As one hexagon consists of 6 nodes, which is similar to a Hierarchical

Star with dimension 2, so the minimum dimension is 2. The number of nodes grows from dimension 2 onwards. Again, as the size grows in quadratic, so the exponent must be multiples of 2, i.e., 2^{n-2}. As the total number of nodes in a hexagon is 6. So, the total number of nodes in HH(n) will be $6^{2^{n-2}}$. Hence the proof.

Theorem 2. HH(n) consists of $6^{2^{n-2}-1}$ number of hexagons.

Proof: From Theorem 1, it is clear that the total number of nodes present in HH(n) is $6^{2^{n-2}}$. So, the total numbers of hexagons in an HH(n) are $\frac{6^{2^{n-2}}}{6} = 6^{2^{n-2-1}}$. Hence the proof.

Theorem 3. The total number of modules of an HH(n) is $6^{2^{n-3}}$ for $n \geq 3$.

Proof: The size of HH(n) increases in quadratic (exponential power of 2). So, the total number of modules can be obtained from the square root of the total number of nodes present, i.e., $\sqrt{6^{2^{n-2}}}$=$6^{2^{n-2-1}}$= $6^{2^{n-3}}$. Hence the proof.

Lemma 1. Each module of an HH(n) consists of a number of sub-modules. From the total number of nodes, the sub-module number can be derived by dividing the node number by the total number of modules.

Illustration 1. Here, we illustrate the applicability of the Theorem 3 with an example.

An HH(n) consists of $6^{2^{n-3}}$ number of modules and the same number of nodes in each module. Again, each module consists of $6^{2^{n-2}/4}$ a number of sub-modules and this division continue until it obtains one hexagon. Let n = 4, then the HH(4) consist of $6^{2^{4-2}/2}$=36 modules and 36 nodes in each module. Again, each module consists of 6 hexagonal sub-modules. The total number of nodes = 6^4 = 1296 nodes. Suppose, the node number is 1293, then, the module number and node number within a module can be obtained by |1293|mod36 = node33 of the 35th module. Again from node number 33, the sub-module number is obtained by |33|mod6= node3 of 5th sub-module.

Theorem 4. The total number of edges of an n-dimensional hexagon HH(n) is $6^{2^{n-2}} \times n/2$.

Proof: The total number of edges present in a regular topology is the product of the total number of nodes and half of a degree. In HH(n), the degree is n and the total number of nodes is $6^{2^{n-2}}$. So, the total number of edges = $6^{2^{n-2}} \times n/2$. Hence the Proof.

Illustration 2. Here, we illustrate the applicability of the Theorem 4 with an example.

Let n = 2, then for the HH(2) total number of edges = $6^{2^{2-2}} \times \frac{2}{2} = 6$. It is also shown in Fig. 1 that HH(2) consists of 6 nodes. Similarly, the HH(3) as shown in Fig. 1 consists of $6^{2^{3-2}} \times \frac{3}{2} = \frac{36\times3}{2} = 54$ number of edges.

Theorem 5. The diameter $D(n)$ of HH(n) of degree n can be found, recursively by the expression

$$D(n)=\begin{Bmatrix} 2 \times D\ (n-1)-\ 1 & \text{if} & n > 2, \\ 3 & \text{if} & n = 2, \end{Bmatrix}$$

where $D(n-1)$ is the diameter of HH($n-1$).

Proof: The diameter of a network is the maximum of the shortest distance between any two nodes. From Definition 1, it is clear that all modules are interconnected and within one module each sub-modules are also interconnected. So, there is at least one edge in between any two modules and also in between any two sub-modules within a module. The total number of external edges is half of the total number of modules. As the hexagon is the basic building block which consists of 6 edges, so the diameter of a hexagon is 6/2=3. The number of hexagons increases with the increase in dimension. The total distance between any two nodes can be reduced by two edges if an external link exists in between two modules, otherwise, the path will be reduced by 1 either through an external link of another module. Hence the proof.

Illustration 3. The Theorem 5 is illustrated here through an example.

Let us consider n = 3, the distance between node1 (0, 1) to node 2 (4, 2) of an HH(3) can be computed as follows.

As there is no external link between node1 and node2, so a path can be selected from another module, i.e, module2. The diameter is shown in the black shaded link in Fig. 3.

Theorem 6. The cost of HH(n) is C(n) which is recursively found by

$$C(n) = \begin{cases} n \times (2 \times D(n-1) - 1) & \text{if } n > 2, \\ 3 \times n & \text{if } n = 2. \end{cases}$$

Proof: The cost of the network is the product of the node degree and diameter. It directly follows from Theorem 5.

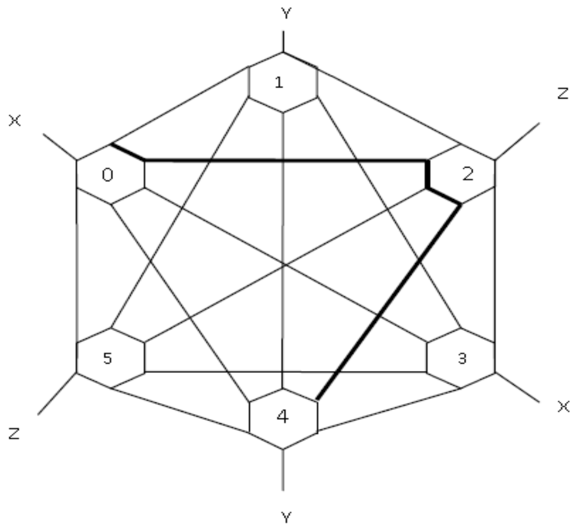

Fig. 3. Illustration of the diameter of HH(3)

Theorem 7. The bisection width of HH(n) is

$$\begin{cases} \left(\dfrac{6^{2^{n-3}}}{2} + 1\right) \times \dfrac{6^{2^{n-3}}}{2} & \text{if } n \geq 3, \\ 3 & \text{if } n = 2. \end{cases}$$

P r o o f: The bisection width of a network is the minimum number of links required to be removed in order to divide the network into two equal halves. The bisection width plays a vital role in VLSI layouts of the network topology. The minimum dimension of HH(n) is 2 which is a hexagon. A Hexagon consists of 6 edges, so, the bisection width for a hexagon is 3. In HH(n), each module is interconnected by two links, i.e., one internal link to each module and one external link to a specific module. So, the bisection width will be half of the total modules in addition to the total number of an external link. Hence the proof.

Illustration 4. The Theorem 7 is illustrated through the following example.

As shown in Fig. 3, the total number of modules is 6. So, we can take any three modules. Each module is connected to the other three modules by a single link and one external link. So, the total edge=$3\times3+3$(for external link) = 12. By removing these 12 edges, the topology structure will be divided into two equal halves. So, the bisection width of HH(3)=12.

Lemma 2. The total number of disjoint paths between any two nodes in HH(n) is n.

P r o o f: As the degree of HH(n) is n so, n number of edges are incident on any node, hich provides n number of disjoint paths to this node. Hence the proof.

5. Routing in HH

In this Section, routings in two different situations are considered for the proposed network HH(n). Those two are fault-free and faulty situation. The routing can be static in a fault-free situation. In the case of the occurrence of fault or congestion, a path can be selected dynamically. The routing in HH(n) is done by forwarding the messages directly to the module where the destination node is present. Instead of broadcasting, selective forwarding is done in HH(n) in order to reduce the network congestion. The source and destination nodes can be represented by two parameters as mentioned as per the Definition 1.

Let, Source = node1(X, Y), and Destination = node2 (X', Y'). The path between the source and destination can be obtained by using the following routing techniques.

5.1. Routing in Fault-free situation

Case 1. The destination is in the same module as the source node ($X=X'$).

All the sub-modules are interconnected within a module. For routing, the message can be sent to the sub-module directly where the destination node is present. The computation of finding a sub-module is done in a recursive manner until the smallest sub-module (Hexagon) is obtained. Within a hexagon, the message is forwarded in the shortest path.

Case 2. The source node and the destination node are in different modules ($X\neq X'$).

At first the message is directly sent to the destination module. Then, the destination sub-module can be computed by following the procedure outlined for Case 1.

i. Illustration of Fault-free routing

Here, we illustrate the details of routing in fault-free situation. We illustrate the fault-free routing in two cases, i.e., source and destination are present in the same module and in different modules. Let us consider node1 (0, 4), and node2 (0, 1) as source node and destination node respectively. As the module numbers of both nodes are same so, it indicates that both nodes are in the same module. The module cannot be further divided to obtain sub-module number as it is the smallest sub-module (Hexagon). Hence, the source and destination nodes are in the same Hexagon and message can be forwarded directly through the shortest path. The path from source1 and destination1 is illustrated in shaded colour in Fig. 4. Similarly, the Case 2 of the fault-free situation is illustrated in the same figure. Consider node3 (4, 5) as source node and node4 (2, 5) as the destination node. As the module numbers are different so, they are in different modules and further sub-module is the smallest one. So, the message can be directly forwarded to the destination module and within sub-module (Hexagon) the path can be selected as per Case 1 of Fault-free routing.

The two paths for the above considered cases are explained as follows.

The Path1 represents the path between the source1 and the destination1 and the Path2 represents the path between the source2 and the destination2.

Source1= node1 (0, 4), Destination1= node2 (0, 1),

Path1= (0, 4) – (0, 5) – (0, 0) – (0, 1),

Source2 = node3 (4, 5), Detination2 = node4 (2, 5),

Path2 = (4, 5) – (4, 0) – (4, 1) – (4, 2) – (2, 4) – (2, 5).

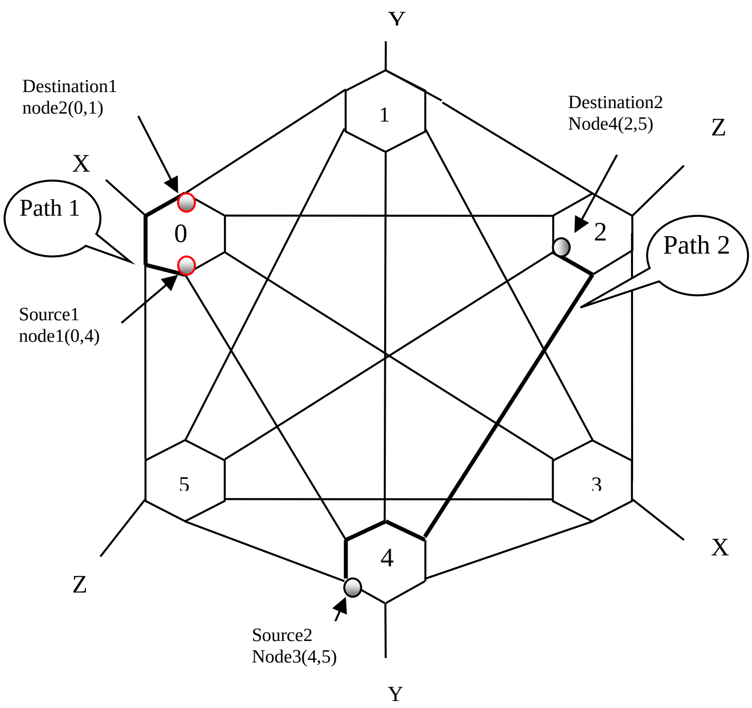

Fig. 4. Path in Fault-free situation

5.2. Routing in a faulty situation (Fault-tolerant routing)

Here, we consider the following cases of a faulty situation.

Case 1. The source and destination are in same smallest sub-module (Hexagon).

In the case of link/node becoming faulty, the message can be sent to the destination by forwarding the message in reverse direction.

Case 2. The source node and the destination node are in different modules ($X \neq X'$).

In the case of link/node becoming faulty, the message can be sent to the module connected to the node just before the faulty node/link. Then, that module will send the message directly to the destination module. The routing in HH(n) is selective forwarding instead of broadcasting.

ii. Illustration of routing

The fault-tolerant routing in the said cases is illustrated by considering different positions of the source and destination nodes. Let us consider two nodes, i.e., source1 node1 (3, 5) and destination1 node2 (3, 1). As both nodes are in the same module so, the path is selected according to Case 1 of fault-free routing procedure. It is shown in Fig. 5. The dotted lines show the faulty links. The faulty link is tolerated by forwarding packets in reverse direction as per Case 1 of fault-tolerant routing method. As the source and destination module are different, so, the path is established through a direct link to the destination module as per Case 2 of routing. The number on the link shows the following type of path.

Static path/ Fault-free path=1

Fault tolerant path=2.

In a link fault situation, the fault can be tolerated by forwarding packets to the other module which is directly connected to the node before/after the faulty one. The dotted line shown in Fig. 5 represents the faulty path. This fault is tolerated by forwarding packets to module 2 and then to the destination module through a direct link. So,

Path1= (4, 2) – (4, 1) – (4, 0) – (0, 4) – (0, 5) – (0, 0),

Path2= (4, 2) – (2, 4) – (2, 5) – (2, 0) – (0, 2) – (0, 1) – (0, 0).

According to Lemma 2, the total number of disjoint paths between any two nodes of HH(3) is 3. In Fig. 5 the disjoint paths between node1 (4, 2) and node2 (0, 0) are as follow:

Path1 = (4, 2) – (4, 1) – (4, 0) – (0, 4) – (0, 5) – (0, 0),

Path2 = (4, 2) – (2, 4) – (2, 5) – (2, 0) – (0, 2) – (0, 1) – (0, 0),

Path3 = (4, 2) – (4, 3) – (3, 4) – (3, 3) – (0, 0).

According to the proposed principle of the routing of HH(n), the Path1 is to be selected, because the source module is directly connected to the destination module. But in Path2 and Path3, the source and destination modules are connected through other modules. In static routing, the Path1 will be selected and rest paths can be used in case of a fault situation.

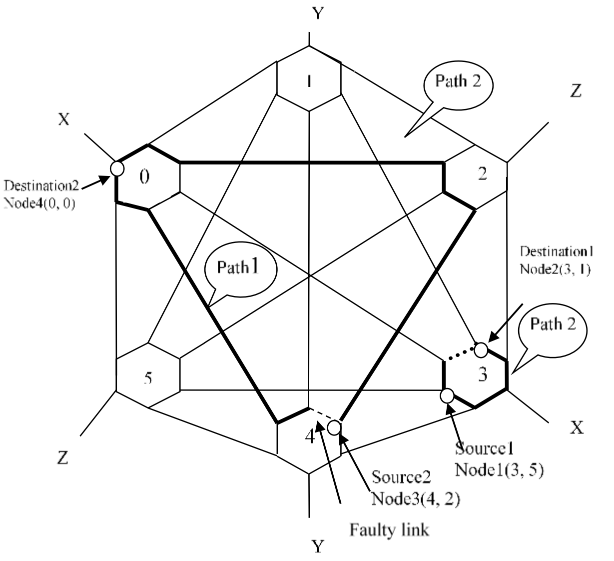

Fig. 5. Path between source node1 (4, 2) and destination node2 (0, 0)

6. Performance comparison

Table 1. Analytical comparisons of HH(n)

Type of Network	Degree	Number of nodes	Diameter	Cost factor
Hierarchical Hexagon HH(n) (proposed)	n	$6^{2^{n-2}}$	$\begin{cases} 2\times D\ (n-1)-\ 1 & \text{if } n>2 \\ 3 & \text{if } \ n=2 \end{cases}$	$\begin{cases} n\times(2\times D\ (n-1)-\ 1) & \text{if } n>2 \\ 3\times n & \text{if } \ n=2 \end{cases}$
Hierarchical Star HS$_{(n,\,n)}$	n	$(n!)^2$	$2\left\lfloor\frac{3(n-1)}{2}\right\rfloor+1$	$n\times(2\left\lfloor\frac{3(n-1)}{2}\right\rfloor+1)$
Star S_n	$n-1$	$n!$	$\left\lfloor\frac{3(n-1)}{2}\right\rfloor$	$(n-1)\times\left\lfloor\frac{3(n-1)}{2}\right\rfloor$
Folded Hypercube FH$_n$	$n+1$	2^n	$\left\lceil\frac{n+1}{2}\right\rceil$	$(n+1)\times\left\lceil\frac{n+1}{2}\right\rceil$
Hierarchical Folded Hypercube HFN$_{(n,\,n)}$	$n+2$	2^{2n}	$2\left\lceil\frac{n+1}{2}\right\rceil+1$	$(n+2)\times(2\left\lceil\frac{n+1}{2}\right\rceil+1)$
Metastar Mstar(k, m)	$k+m-1$	$2^k(m!)^2$	$(\left\lfloor\frac{3(m-1)}{2}\right\rfloor+1)2^k$	$(\left\lfloor\frac{3(m-1)}{2}\right\rfloor+1)2^k\times(k+m-1)$
Exchanged Crossed Cube ECQ(s, t)	$(s+1)/(t+1)$	2^{s+t+1}	$\left(\left\lceil\frac{s+1}{2}\right\rceil+\left\lceil\frac{t+1}{2}\right\rceil+2\right)$	$\left(\left\lceil\frac{s+1}{2}\right\rceil+\left\lceil\frac{t+1}{2}\right\rceil+2\right)\times\left(\frac{s+t+2}{2}\right)$
Exchange Folded Cross Cube EFCQ(s, t)	$(s+2)/(t+2)$	2^{s+t+1}	$\left(\left\lceil\frac{s+2}{2}\right\rceil+\left\lceil\frac{t+2}{2}\right\rceil-1\right)$	$\left(\frac{s+t+4}{2}\right)\times$ $\times\left(\left\lceil\frac{s+2}{2}\right\rceil+\left\lceil\frac{t+2}{2}\right\rceil-1\right)$

Table 2. The numerical comparison of HH(n)

Network type	Value of n	Number of nodes	Degree	Diameter	Cost factor
Hierarchical Hexagon HH(n) (proposed)	2	6	2	3	6
	4	1296	4	9	36
	6	2.82×10^{12}	6	33	198
	8	1.85×10^{49}	8	129	1032
	10	1.6×10^{199}	10	513	5130
	12	6.71×10^{796}	12	2049	24,588
Metastar Mstar(k, m)	2	8	2	4	8
	4	1152	4	10	40
	6	1.03×106	6	16	96
	8	3.26×109	8	22	176
	10	2.64×10^{13}	10	28	280
	12	4.58×10^{17}	12	34	408
Hierarchical Star HS(n, n)	2	6	2	3	6
	4	576	4	9	36
	6	5.18×10^{5}	6	15	90
	8	1.63×10^{9}	8	21	168
	10	1.32×10^{13}	10	27	270
	12	2.29×10^{17}	12	33	396
Star graph S_n	6	720	5	7	35
	9	3.62×10^{5}	8	12	96
	13	6.23×10^{9}	12	18	216
	16	2.09×10^{13}	15	22	330
	19	1.22×10^{17}	18	27	486
	22	1.12×10^{21}	21	31	651
Folded Hypercube FH_n	9	512	10	5	50
	19	5.24×10^{5}	20	10	200
	30	1.07×10^{9}	31	16	496
	43	8.8×10^{12}	44	22	968
	57	1.44×10^{17}	58	29	1682
	73	9.94×10^{21}	74	37	2378
Hierarchical Folded hypercube HFN(n, n)	5	1024	7	7	49
	9	2.62×10^{5}	11	11	121
	15	1.07×10^{9}	17	17	289
	22	1.76×10^{13}	24	25	600
	29	2.88×10^{17}	31	31	961
	36	4.72×10^{21}	38	39	1482
	4	38416	3.2	21	67.2
	6	7.5×10^{6}	3.2	33	105.6
	8	14×10^{8}	3.2	45	144
	9	20×10^{9}	3.2	51	163.2
	10	28.92×10^{10}	3.2	57	182.4
	11	4.049×10^{12}	3.2	63	201.6
	40	7×10^{45}	3.2	237	545.1
ECQ	3	8	2	4	8
	4	16	3	5	15
	5	32	3	5	15
	6	64	4	6	24
	7	128	4	6	24
	9	512	5	7	35
	10	1024	6	8	48
	12	4096	7	9	63
EFCQ	3	8	3	2	6
	4	16	4	3	12
	5	32	4	3	12
	6	64	5	4	20
	7	128	6	4	24
	9	512	7	5	35
	10	1024	7	6	42
	12	4096	8	7	56

In general, for a desirable interconnection structure, the degree, diameter, and cost should be as small as possible. A network with large node degree tends to increase the hardware cost. Smaller diameter means the lower communication overheads. As the diameter of HH(n) can be obtained recursively so, the cost factor is also computed, recursively.

In this section, the performance of the proposed Hierarchical Hexagon network is compared with the existing networks based upon the results of the comparison of various topological properties. The comparison is done in terms of degree, the number of nodes, diameter, and its cost. Table 1 presents the analytical comparison among the HH(n), HS(n, n), S_n, FH_n, HFN(n, n), Mstar(k, m), ECQ [25], and EFCQ [26].

The results in Table 1 show that these networks have their own advantages and disadvantages. The numerical comparisons of various performance metrics of different networks with the proposed network are shown in Table 2.

For a particular node degree, the total number of nodes in the proposed network HH(n) is observed to be more among all the topologies. As an example: let us consider node degree 6 in Table 2, the number of nodes present in HH(n) is the highest among Mstar, Star graph, Hierarchical Star, Folded Hypercube, Hierarchical Folded hypercube, ECQ, and EFCQ. Similarly, the cost is also the least among other topologies for any size of the network. The result of the comparison of our proposed network HH(n) with other topologies in terms of size, degree, cost, diameter, and packing density are shown in Fig. 3 to Fig. 6. The comparison of diameter and the degree of HH(n) with some prominent networks like Dragonfly, Fat tree, Tofu, 3-D Torus are also shown in Table 3.

Table 3. Comparison of the degree and diameter of the proposed network Hierarchical Hexagon (for size=4096).

Topology	Degree	Diameter
3-D Torus	6	24
Tofu	10	12
Hypercube	12	12
Dragonfly	48	5
Fat-tree	32	5
Hierarchical Hexagon	5	17

From the Table 3, it is observed that for a particular size of a network, the degree of our proposed network Hierarchical Hexagon is the lowest, though the diameter is more than others except for the 3-D Torus.

From Fig. 6, it is observed that for designing a higher size network, the required node degree is the least in case of the proposed network HH(n). The degree of HS(n) is found to be the lowest among the Star graph, Folded Hypercube, and Hierarchical Folded Hypercube. The comparison of the diameter with respect to the size of the network is shown in Fig. 7.

The diameter of HH(n) is observed to be smaller than HFcube [17]. As HFCube has the lowest diameter among dBCube(c, d), HHC, HCN(n, n), FolH, RFH, REH, FloH, and THIN [27]. So, the HH(n) has the lowest diameter among all these networks. The diameter of *n*-star is the lowest for all size of the network but the degree is more in comparison to HH(*n*).

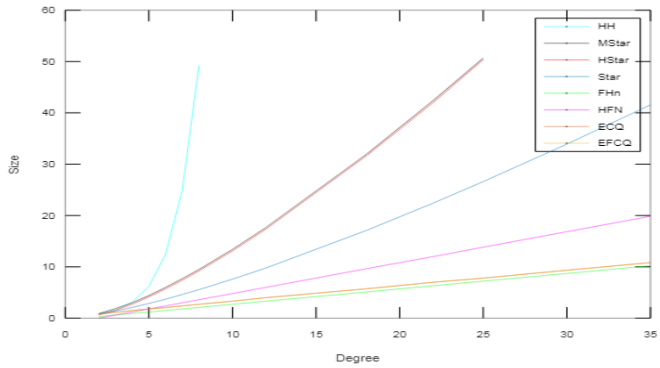

Fig. 6. Comparison of size with degree of HH(*n*)

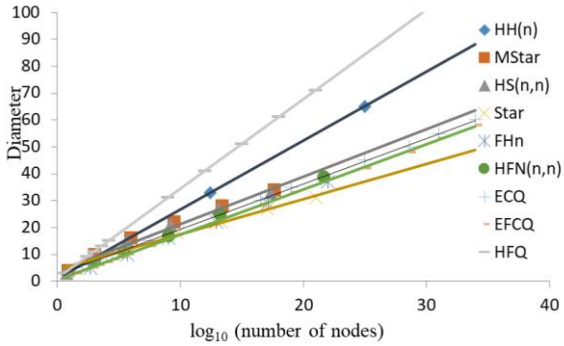

Fig. 7. Comparison of diameter versus size of HH(*n*)

The comparison of cost is shown in Fig. 8. The cost of HH(*n*) is the smallest of all the networks: Star graph, HStar, MStar, FHn, HFn, ECQ, and EFCQ for any size of the network.

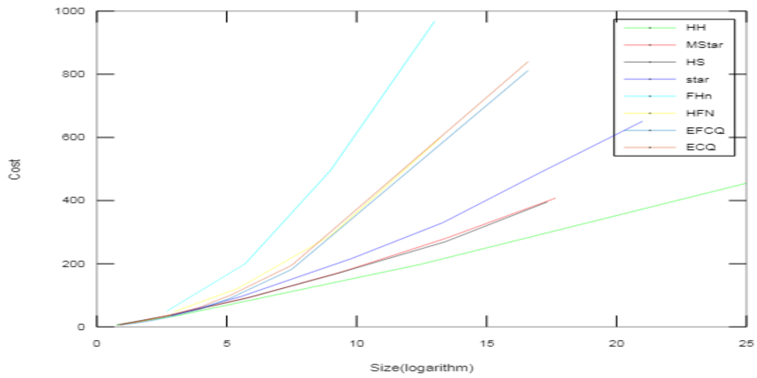

Fig. 8. Comparision of cost versus size of HH(*n*)

The packing density of a network is defined in terms of the size of the network per unit cost. It is an important factor in deciding the size of VLSI design layout. The comparison of packing density of HH(n) is compared and shown in Fig. 9. From the same figure, it is observed that the packing density of HH(n) is the highest among all topologies under consideration.

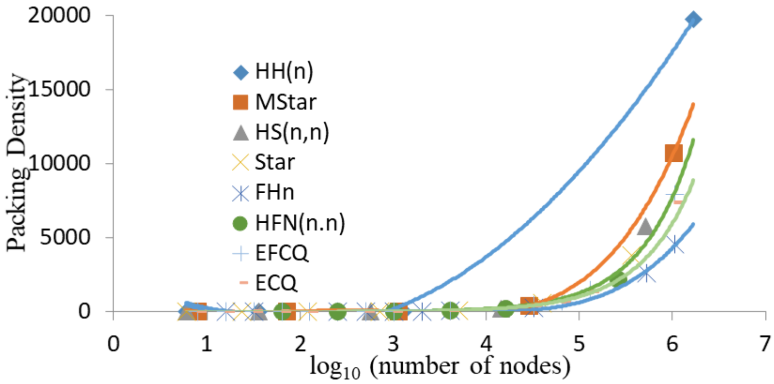

Fig. 9. Comparison of packing density versus size of HH(n)

7. Hamiltonian property of the proposed network HH(n)

The presence of the Hamiltonian cycle contributes significantly to the fault-tolerance of an interconnection. A Hamiltonian cycle is a simple cycle that visits every node exactly once. A Hamiltonian path is derived from a Hamiltonian cycle by removing any link from that cycle. Thus, each node of a network is assigned a label. The label is assigned depending upon the position of a node in a Hamiltonian path in the network. The first node in the path is assigned as label 0 and the last node in the path is assigned label $N-1$, where N is the total number of nodes in the network. It is also used to design adaptive fault-tolerant routing technique. The desired property of an interconnection topology is the presence of a Hamilton cycle, which increases fault tolerance and it plays a vital role in the design of parallel algorithms, such as broadcasting. As deadlock recovery can be achieved on any network with Hamiltonian cycle by using two deadlock buffers per node.

Theorem 8. Hierarchical Hexagon(n) contains a Hamiltonian cycle.

Proof: Garey and Johnson [28] in 1979 proved that to find the Hamiltonian cycle of a graph belongs to the NP-Complete problem of computer science. However, Rahman, Kaykobad and Firoz [29] in 2014 and Mehedy, Kamrul and Kaykobad [30] in 2007 impose conditions to prove the presence of the Hamiltonian cycle of a graph. According to the given conditions of Mehedy, Kamrul and Kaykobad [30] in 2007, if a graph consists of at least $n^2/4$ number of edges then it ensures the presence of the Hamiltonian cycle, where n represents the number of nodes. The total number of edges of an HH(n) is $6^{2^{n-2}} \times (n/2)$. In order to prove the presence of Hamiltonian cycle in HH(n), the total number of edges must be more than $n^2/4$. This can be proved by mathematical induction method.

Let $P(n) = 6^{2^{n-2}} \times n/2$.

For $n = 2$, $P(n)$ is 6.

So, $P(n) > n^2/4$ is true for $n=2$.

Similarly $P(n)$ is true for $n = k$. So, we have to prove that $P(n)$ is true for $k + 1$, i.e., $P(k + 1) > (k + 1)^2/4$,

(1) $$6^{2^{k+1-2}} = 6^{2^{k-1}} > (k+1)^2/4,$$

as $P(k)$ is true So, $6^{2^{k-2}} > k^2 / 4$

Since $6^{2^{k-2}} < 6^{2^{k-1}}$,

(2) $$6^{2^{k-2}} > k^2/4.$$

By finding square root in both side of Equation (2) we get

(3) $$6^{2^{k-3}} > k/2.$$

For all value of $k \geq 2$,

(4) $$6^{2^{k-2}} > 1/4.$$

By adding Equations (2), (3) and (4), we get

(5) $$6^{2^{k-2}}\left(2 + 6^{2^{-1}}\right) > (k^2+1+2k)/4,$$

As $2 + 6^{2^{-1}} < 6$.

So, Equation (5) can be written as $6^{2^{k-2}} \times 6 > (k^2+1+2k)/4 = 6^{2^{k-1}} > (k+1)^2/4$.

So, $P(k + 1)$ is true for any value of n.

Hence, the graph contains the Hamiltonian cycle.

A. Illustration of the Hamiltonian cycle in HH(n)

Starting from any module and the rest modules are covered in a cycle as follows: Module x – Module (x+1) – Module (x–1) – Module (x+2) – Module (x–2) – Module (x+3) –...+ Module (x+m) – Module x, where m is the number of modules.

When the message traverses to the next module (+ve sign), then the path will be in the anticlockwise direction within the module and the clockwise direction for before module (– ve sign).

Illustration 5. Here the applicability of Theorem 8 is illustrated with an example. The presence of the Hamiltonian cycle in the proposed HH(n) is illustrated. Consider the starting module for constructing a Hamiltonian path of HH(3) is 1. Hamiltonian cycle contains modules in the following order:

Module 1 – Module 2 – Module 0 – Module 3 – Module 5 – Module 4 – Module 1.

The Hamiltonian cycle is shown in Fig. 10, where the starting module is 1. The detail path is Module 1 (1-0-5-4-3-2) – Module 2 (1-2-3-4-5-0) – Module 0 (2-1-0-5-4-3) – Module 3 (0-1-2-3-4-5) – Module 4 (5-0-1-2-3-4) – Module 1 (1).

Similarly, the Hamiltonian cycle can also start from any other module. So, the total number of Hamiltonian cycles = the total number of modules/2. The total path length of the Hamiltonian cycle of HH(3) is 6×5+6=36. So HH(3) consists of three Hamiltonian cycles of length 36 each.

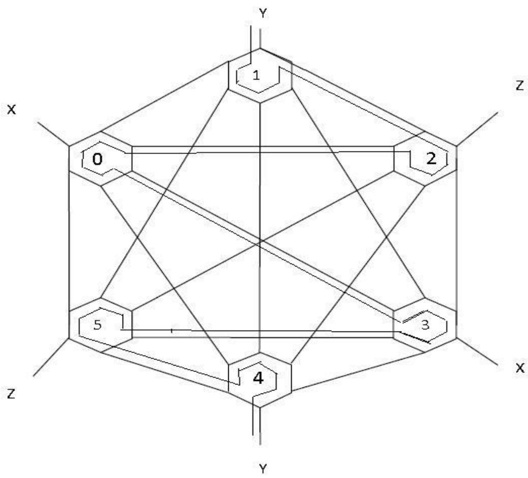

Fig. 10. Hamiltonian cycle embedded upon HH(3)

8. Conclusion

In this paper, a new hierarchical fault-tolerant interconnection network called as Hierarchical Hexagon (HH(*n*)) was proposed. Our proposed network can be used for massively parallel systems. It uses Hexagon as a basic building block. Its topological properties such as node degree, diameter, bisection width, network cost, and packing density were investigated and compared with other contemporary networks. It has been observed that HH(*n*) is best of the Star graphs, Hierarchical Star, Mstar, and other networks under consideration with regard to degree, diameter and network cost. The routing techniques for four different cases are presented along with illustrations:

(i) Fault-free routing in case of both source and destination are in same module;

(ii) Routing in fault-free situation where both source and destination nodes are in different module;

(iii) Fault-tolerant routing in the case of both source and destination nodes are in the same module;

(iv) Fault-tolerant routing in the case of both source and destination nodes are in different modules.

The HH(*n*) is found to offer a high degree of fault tolerance. The presence of a Hamiltonian cycle was illustrated through an example. The above comparisons are summarized below. For any size of a network, the node degree of HH(*n*) is always less as compared to the other networks under consideration.

(1) The diameter of HH(*n*) is smaller than HFCube, Fat tree, Dragonfly, 3-D Torus, Tofu, dBCube(*c*, *d*), HHC, HCN(*n*, *n*), FolH, RFH, REH, FloH, and THIN.

(2) The cost of HH(*n*) is observed to be the lowest among all other networks under consideration.

(3) The packing density of HH(*n*) is more than MStar, HStar, and HFcube. So, HH(*n*) is also suitable for design of VLSI chips.

References

1. A k e r s, S. B., D. H a r e l, B. K r i s h n a m u r t h y. The Star Graph: An Attractive Alternative to *n*-Cube. – In: Proc. of International Conference on Parallel Processing (ICPP'87), St. Charles, Illinois, 1987, pp. 393-400.
2. D a y, K., A. T r i p a t h i. A Comparative Study of Topological Properties of Hypercubes and Star Graphs. – Technical Report TR, 1991, Computer Science Department, University of Minnesota, Minneapolis, MN, pp. 91-10.
3. Q i u, K., H. M e i j e r, S. G. A k l. Decomposing a Star Graph into Disjoint Cycles. – Information Processing Letters, Vol. **39**, 1991, No 3, pp. 125-129.
4. N i g a m, M., S. S a h n i, B. K r i s h n a m u r t h y. Embedding Hamiltonians, and Hypercubes in Star Interconnection Graphs. – In: Proc. of International Conference on Parallel Processing, 1990, pp. 340-343.
5. E l-A m a w y, A., S. L a t i f i. Properties and performance of Folded Hypercube. – IEEE Transactions on Parallel and Distributed Systems, Vol. **2**, 1991, No 2, pp. 31-42.
6. G h o s e, K., K. R. D e s a i. The HCN: A Versatile Interconnection Network Based on Cubes. – In: Proc. of ACM International Conference on Supercomputing, 1989, pp. 426-435.
7. G h o s e, K., K. R. D e s a i. The Design and Evaluation of the Hierarchical Cubic Network. – In: Proc. of International Conference on Parallel Processing, Vol. **1**, 1990, pp. 355-362.
8. A d h i k a r i, N., C. R. T r i p a t h y. Mstar: A New Two Level Interconnection Network. – In: Proc. of 8th International Conference on Distributed Computing and Internet Technology (ICDCIT'12), 2012, pp. 50-61.
9. L i, Y., S. P e n g, W. C h u. Metacube: A New Interconnection Network for Large Parallel System. – In: Proc. of ACSAC02, Australian Computer Science Communication, Vol. **24**, 2001, No 4, pp. 29-36.
10. T r i p a t h y, C. R. Star-Cube: A New Fault-Tolerant Interconnection Topology for Massively Parallel Systems. – IE(I) Journal, ETE Div., Vol. **84**, 2004, No 2, pp. 83-92.
11. S h i, W., P. K. S r i m a n i. Hierarchical Star: A New Two-Level Interconnection Network. – Journal of Systems Architecture, Vol. **51**, 2005, pp. 1-14.
12. D u h, D. R., G. H. C h e n, J. F. F a n g. Algorithms and Properties of a New Two-Level Network with Folded Hypercubes as Basic Modules. – IEEE Transactions on Parallel and Distributed Systems, Vol. **6**, 1995, No 7, pp. 714-723.
13. M a l l u h i, Q. M., M. A. B a y o u m i. Hierarchical Hypercube: A New Interconnection Topology for Massively Parallel Systems – IEEE Transactions on Parallel and Distributed Systems, Vol. **5**, 1994, No 1, pp.17-30.
14. D a n d a m u d i, S. P., D. L. E a g e r. Hierarchical Interconnection Networks for Multicomputer Systems. – IEEE Transactions on Computers, Vol. **39**, 1990, No 6, pp. 786-797.
15. T a k a b a t a k e, T., K. K a n e k o, H. I t o. HCC: Generalized Hierarchical Completely-Connected Networks. – IEICE Transactions on Information & Systems, Vol. **E83-D**, 2000, No 6, pp. 1216-1224.
16. L i u, Y., C. L i, J. H a n. RTTM: A New Hierarchical Interconnection Network for Massively Parallel Computing. High Performance Computing and Applications. – Lecture Notes in Computer Science, Vol. **5938**, 2010, Springer, Berlin, Germany, pp. 264-271.
17. S h i, Y., Z. H o u, J. S o n g. Hierarchical Interconnection Networks with Folded Hypercubes as Basic Cluster. – In: Proc. of 4th International Conference/Exhibition on High Performance Computing in the Asia-Pacific Region, Vol. **1**, 2000, pp. 134-137.
18. K i m, J., J. D a l l y, W. S c o t t, S. A. D e n n i s. Technology-Driven, Highly Scalable Dragonfly Topology. – In: Proc. of ISCA '08. Washington, DC, USA IEEE Computer Society, 2008, pp. 77-88.
19. A j i m a, Y., I. T o m o h i r o, H. S i n y a, S. T o s h i y u k i. The Tofu Interconnect. – IEEE Micro, Vol. **32**, 2012, No 1, pp. 21-31.
20. H a y e s J. P., T. N. M u d g e. Hypercube Super-Computers. – In: Proc. of IEEE Society, Vol. **77**, No 12, 1989, pp. 1829-1841.
21. L e i s e r s o n, C. E. Fat-Trees: Universal Networks Hardware Efficient Supercomputing. – IEEE Transactions on Computers, Vol. **c-34**, 1985, No 10, pp. 892-901.

22. T r o b e c, R., R. V a s i l j e v i c, M. T o m a s e v i c, V. M i l u t i n o v i c, R. B e i v i d e, M. V a l e r o. Interconnection Networks in Petascale Computer Systems: A Survey. – Journal of ACM Computing Survey (CSUR), Vol. **49**, 2017, No 3, pp. 1-24.
23. A r i m i l l i, B., R. A r i m i l l i, V. C h u n g, S. C l a r k, W. D e n z e l, B. D r e r u p, T. H o e f l e r, J. J o y n e r, J. L e w i s, J. L i, N. N i, R. R a j a m o n y. The PERCS High-Performance Interconnect. – In: Proc. of 18th IEEE Annual Symposium on High-Performance Interconnects (HOTI), 2010, pp. 75-82.
24. F u e n t e s, P., E. V a l l e j o, C. M a r t´ı n e z, M. G a r c´ı a, R. B e i v i d e. A Comparison Study of Scalable and Cost-Effective Interconnection Networks for HPC. – In: Proc. of 41st International Conference on Parallel Processing Workshops (ICPPW), 2012, pp. 594-595.
25. L i, K., Y. M u, K. L i, G. M i n. Exchanged Crossed Cube: A Novel Interconnection Network for Parallel Computation. – IEEE Transactions on Parallel and Distributed Systems, Vol. **24**, No 11, 2013, pp. 2211-2219.
26. B h a v a n i, K., S. J e n a. Exchanged Folded Crossed Cube: A New Interconnection Network for Parallel Computation. – Information Processing Letter, Vol. **137**, 2018, pp. 40-46.
27. A b d-E l-B a r r, M., T. F. A l-S o m a n i. Topological Properties of Hierarchical Interconnection Networks. – A Review and Comparison. – Journal of Electrical and Computer Engineering, Vol. **2011**, 2011, pp. 1-12.
28. G a r e y, M. R., D. S. J o h n s o n. Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman, New York, NY, USA, 1979.
29. R a h m a n, M. S., M. K a y k o b a d, J. S. F i r o z. New Sufficient Conditions for Hamiltonian Paths. –The Scientific World Journal, Vol. **2014**, 2014, pp. 1-7.
30. M e h e d y, L., H. M. K a m r u l, M. K a y k o b a d. An Improved Degree Based Condition for Hamiltonian Cycles. – Information Processing Letters, Vol. **102**, 2007, pp. 108-112,

Received: 15.07.2020; Second Version: 19.01.2021; Accepted: 29.01.2021