LINEARLY EXTENDIBLE ARM (LEA) – A CONSTANT DEGREE TOPOLOGY FOR DESIGNING SCALABLE AND COST EFFECTIVE INTERCONNECTION NETWORKS

Jahangir Alam[1], Rajesh Kumar[2] and Zubair Khan[3]

[1]University Women's Polytechnic, Faculty of Engineering & Technology, Aligarh Muslim University Aligarh, U.P. - 202002, India
jahangiralam@zhcet.ac.in

[2] School of Mathematics & Computer Applications, Thapar University
Patiala, P.B. - 147004, India
rakumar@tiet.ac.in

[3]Department of Computer Science & Engineering, IIET Bareilly India
zubair.762001@gmail.com

ABSTRACT

This paper introduces a new class of processor interconnection topology called LEA (Linearly Extendable ARM). The attractive properties of LEA topology include constant degree, moderate diameter, regularity, symmetry, scalability and simple routing. LEA, when used as one of the components of an interconnection topology based on product graphs, is capable of maintaining the constant degree over a large number of nodes. Moreover, product networks based on LEA are not only highly scalable but also outperform other topologies of interest when either the degree x diameter or diameter x number of links is used as a desirable quality measure of an interconnection network. To prove the superiority of LEA, a product network based on it, has been compared with a number of topologies of interest. A simple routing algorithm has also been presented for the proposed topology.

Keywords: Interconnection Networks, Twisted Hypercube, Degree, Diameter, Scalability, Linearly Extendable Arm (LEA), Hybrid Topology.

1. INTRODUCTION

The possibility of interconnecting a large number of processors together to solve complex problems in scientific computations has been extensively addressed in the past. The performance of a parallel system is strongly affected by the underlying intercommunication network and the matching of the algorithms with the network topology. Many interconnection network topologies have been proposed in the literature for the purpose of connecting hundreds or thousands of processing elements [1], [2], [3]. All the proposed topologies have attractive features, as well as some inherent limitations. For example, a completely connected network might offer high fault-tolerant capability but is costly due to a large number of communication paths.

The interesting properties of the hypercube topology [4] - a logarithmic diameter, a simple labeling scheme, good connectivity, recursive scalability and symmetry have attracted the attentions of a large number of researchers [5], [6], [7]. A large number of hypercube based multiprocessor systems have become commercially available. Some of them are NCUBE/10 (can have up to 1024 processors), Intel's iPSC series (can have up to 128 processors), FPS's T series (can have up to 4096 processors) and nCube.

Among important parameters of processor topology are its scalability and modularity [8], [9]**.** Scalable processor topologies have the property that the size of the system (e.g., the number of nodes) can be increased with minor change in the existing configuration. Also, the increase in system size is expected to result in an increase in performance to the extent of the increase in size. However, in hypercube, the number of communication paths (wires) and the number of ports per processor increase with the increase in the topology size. This is considered to be a major drawback of the hypercube architecture. With the intension of solving this problem several other topologies have been reported in the literature. Some of the prominent topologies among them are Cross Cube [10], dB-Cube [11] and the hierarchical hypercube (HHC)[12]. These topologies in one way or other are the modification or generalization of the hypercube and suffer from similar problems. For example, to make the node degree independent of the topology size an interesting topology called CCC (Cube Connected Cycles) has been reported in the literature [13]. A CCC network is obtained by replacing each node of hypercube by a ring (cycle) of size n, where n is the dimension of the hypercube. The most attractive feature of CCC is its constant node degree (three). But, it has some unfruitful disadvantages over the hypercube – the large diameter and complex routing.

A recently reported topology has been named as Hex-Cell [3]. It is obtained by cascading the sets of six processors arranged as hexagons. Though, not based on the hypercube, Hex-Cell provides some attractive features like low node degree (less than or equal to 3) and simple routing. The diameter of a Hex-Cell consisting of N nodes is given by $4\sqrt{(N/6)} - 1$, which is significantly large as compared to the logarithmic diameter of hypercube. Such a large diameter may introduce significant delays in the network so, Hex-Cell can't be considered as an eligible candidate for massively parallel systems.

Another major category of processor interconnection topologies which has been extensively reported in the literature [1], [2], [14] is the Hybrid Topologies. A hybrid topology is a topology which is derived from two or more existing topologies using a graph theoretic operation. In most of such topologies researchers have used the Cartesian Product of two topology graphs. The general framework for the Cartesian Product of topology graphs has been first reported by Youssef [22] and was later extended by Khaled et al. [23]

Based on these concepts several product or hybrid topologies have been reported – Arrangement-Star [15], Hyper-Star [16], [17] Star-Cube, Hyper-Mesh [18], Hyper-Petersen [20], Banyan-Hypercube [19].

This paper introduces a new constant degree and scalable processor interconnection topology, called Linearly Extendable Arm (LEA). Other attractive properties of LEA include moderate diameter, good connectivity, symmetry and simple routing. Product Networks based on LEA, inherits all of its appealing properties and can maintain a constant degree over a large number of network size. It has been observed that they outperform other product topologies of interest

The reminder of this paper is organized as follows.

In Section 2 we introduce various preliminary definitions and notations. Section 3 introduces the Linearly Extendable Arm (LEA) topology and its properties. Section 4 discusses routing and broadcasting on LEA networks. Section 5 presents the advantages of using LEA as a product topology component. Section 6 concludes the paper.

2. PRELIMINARIES AND DEFINITIONS

This paper uses the standard graph theoretic terminology [21].

Definition 1: An interconnection network topology is usually represented as an undirected graph G = (V, E), where V is the set of nodes (vertices) and E is the set of edges. Each vertex represents a processor and each edge a communication link between processors.

Definition 2: The degree, d(v), of a vertex v, v ϵ V, is defined as the number of channels incident on the vertex. The number of channels into the vertex is the in-degree, $d_{in}(v)$ The number of channels out of a vertex is the out-degree, $d_{out}(v)$. The total degree, d(v), is the sum, $d(v) = d_{in}(v) + d_{out}(v)$. For an undirected network, the number of edges incident on a vertex is called the degree of that vertex.

Definition 3: The diameter, D(G)**,** of a topology graph , G, having, N, nodes is defined as the longest path, p, of the shortest paths between any two nodes. $D(G) = \max(\min_{p \epsilon p(i,j)} \text{length}(p))$. In this equation, p(i, j), is the length of the path between nodes i and j and length(p) is a procedure that returns the length of the path, p. In simple words diameter of a network is the maximum shortest path between any two nodes. The network diameter indicates the maximum number of distinct hops between any two nodes.

Definition 4: The (κ-1) - fault diameter of a κ-connected topology graph G(V, E), D κ(G), is defined as:

$D_\kappa(G) = \max\{d(G - F) \mid F \subset V(G), |F| = \kappa - 1\}$, Where F is an induced subgraph of G.

Definition 5: An n-dimensional binary hypercube, Q_n, consists of 2^n nodes. Each node v, for $0 \leq v \leq 2^n - 1$, is labeled as n-bit binary string, L(v). There is an edge between two nodes, u and v, if, and only if, their labels differ in exactly one bit position.

Definition 6: A graph, G(V, E), is called regular if all of its vertices have the same degree.

Definition 7: A graph G(V, E) is vertex-symmetric, if for every pair of vertices u and v, u, v ϵ V, there exists an automorphism of the graph that maps u into v.

3. LEA AND ITS TOPOLOGICAL PROPERTIES

Level m LEA is defined as follows. Let m be a positive integer such that m≥2, then level m LEA, denoted as LEA(m), is an undirected graph consisting of N(= 6m) processors (vertices) labeled as P_0, P_1, P_2… … …$P_{(N-1)}$, for $0 \leq i < N$ and arranged in m arms (columns) each of which consists of exactly six processors. A link (edge) exists between processors P_i and P_j, if, and only if: (i+1) modulo N = j or (i+3) modulo N = j. Alternately, for m ≥2, LEA(m) = G(V, E), where,

- $V = \{P_i \mid i \in I^+$ and $0 \leq i < N\}$
- $E = \{(P_i, P_j) \mid j = ((i+1)$ modulo N) OR j = ((i+3) modulo N)}.

From the definition of LEA it is obvious that each processor on an LEA network provides two links to the two distinct processors and receives two links from two distinct processors. Figure 1 shows LEA(3) .

Proposition 1: *The degree of each node of LEA(m) is 4 (constant).*
Proof:

Form the connectivity formulae of LEA it is clear that each processor on an LEA network provides two links to two distinct processors on the network and also receives two links from two distinct processors. Hence total number of links on any processor is 4. Hence, the degree of each node of LEA(m) is 4.

Proposition 2: *The diameter of LEA(m) is m+1*
Proof:

It is easy to note that:

Diameter of 12 node LEA, i.e., D(LEA(2)) = 3.
Diameter of 18 node LEA, i.e., D(LEA(3)) = 4.
Diameter of 24 node LEA i.e., D(LEA(5)) = 5
In the same manner it can be proved that D(LEA(m)) = m+1.

Proposition 3: *LEA(m) is a 4-regular, vertex (node) symmetric graph.*
Proof:

Each node (vertex) in LEA has the same degree (4). So, is clear that LEA(m) is 4-regular graph.

Let LEA(m) network is represented by a graph G(V, E), and σ be a permutation of the vertex set V, of LEA(m). Then, for any edge e = (u, v), e ϵ E and u, v ϵ V, we have σ(e) = (σ(u), σ(v)), is also an edge. It shows that graph G (representing LEA(m)) is isomorphic to itself (automorphism). Figure 2 shows an alternate representation of LEA(2). Using Figure 1 and Figure 2, we can easily understand the automorphism of LEA(3).

Based on the above discussion it is obvious that LEA(m) is 4-regular, vertex-symmetric graph.

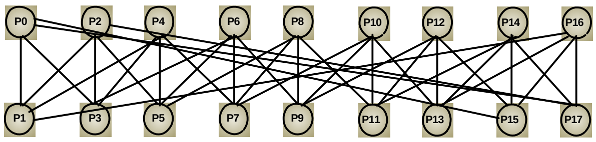

Figure 2: LEA(3) - Alternate Representation

Proposition 4: *The Bisection Width (β) of LEA(k) i.e. β(LEA(k)) is given by 8k or 1.34N, where N is the total number of nodes in LEA(k).*
Proof:

The Bisection Width of an interconnection network is defined as the minimum number of edges (wires) cut to split a network into two parts each having the same number of nodes.

Bisection Width of LEA(k) can be calculated as follows:

Table 1: Bisection Width Calculation for LEA(k)

m	N = 6*m	No. of edges cut to split LEA(m) into two Parts
2	12	16 = (8 x 2) = (4 x 12)/3 = 1.34 x 12
3	18	24 = (8 x 3) = (4 x 18)/3 = 1.34 x 18
4	24	32 = (8 x 4) = (4 x 24)/3 = 1.34 x 24
5	30	40 = (8 x 5) = (4 x 30)/3 = 1.34 x 30
…	…	…
…	…	…
k	6k	8k = (8 x k) = (4 x 6k)/3 = 1.34N

Clearly β(LEA(k)) = 8k = 1.34N

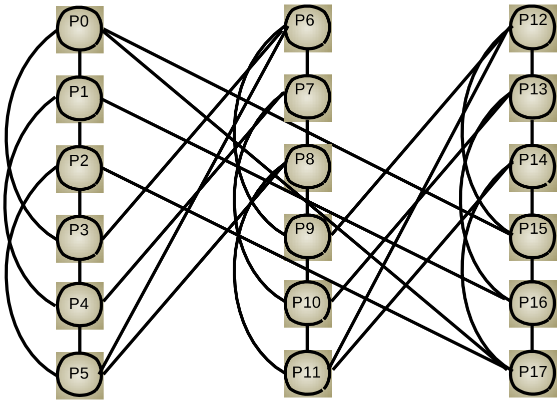

Figure 1: A 18-Node LEA i.e. LEA(3)

Proposition 5: *The average distance of LEA(m) is given by m(3m+4)/(6m-1)^2.*
Proof:

The average distance $\bar{d}(G)$ of a graph G, consisting of N nodes is given by the following equation:

$$\bar{d}(G) = \frac{\sum_{i=1}^{N}\sum_{j=1}^{N} d(i,j)}{N(N-1)}, \text{ where } i \neq j$$

The average distance $\bar{d}_v$ of a node v is obtained from the following equation:

$$\bar{d}_v = \frac{\sum_{j=1}^{N} d(i,j)}{N-1}, \text{ where } i \neq j$$

Average Distance of LEA can be calculated as follows. The average distances for node 1 in different level LEA networks are shown in the following table:

Table 2: LEA Average Distances for First Node

m	No. of Nodes	$\bar{d}_1$
2	12	(1+1+1+1+2+2+2+2+2+3+3)/11 = (4x1+5x2+2x3)/11 = 20/11
3	18	(1+1+1+1+2+2+2+2+2+2+3+3+3+3+3 +4+4)/17 = (4x1+6x2+5x3+2x4)/17=39/17
4	24	(4x1+6x2+6x3+5x4+2x5)/23
…	…	…
…	…	…
k	6k	(4x1+6x2+6x3+ … +6x(k-1)+5xk+2x(k+1)/(6k-1)

Clearly, the average distance of first node in LEA(m) is be given by:

$$\bar{d}_1 = \frac{(4x1 + 6x2 + 6x3 + ...6x(m-1) + 5xm + 2x(m+1)}{(6m-1)}$$

$$\bar{d}_1 = \frac{m(3m+4)}{(6m-1)}$$

Proceeding in the same manner we can prove that average distance of each node is given by the same formula. This is also evident from the fact that LEA is a 4-regular, node symmetric graph.

Now, the average distance of LEA graph i.e. $\bar{d}(LEA)$ could be calculated as follows.

$$\bar{d}(LEA) = \frac{\sum_{i=1}^{N}\sum_{j=1}^{N} d(i,j)}{N(N-1)} = \frac{m(3m+4)}{(6m-1)^2}$$

Proposition 6: *Given two nodes a, and b of an LEA network, then there exist exactly four node disjoint paths between a and b.*
Proof:

Let G(V, E) be a graph and u, v ϵ V, then a set of paths (denoted by η), between u and v, is said to be node disjoint if no node except u and v appears in more than one path. They are also referred to as parallel paths. Presence of node disjoint paths is essential between two nodes in an interconnection network to speed up transfers of large amounts of data and to provide alternate routes in cases of link or node failures.

To verify this proposition we have randomly chosen two distinct sets of source and destination nodes from different level LEA networks and listed the node disjoint between them. Our findings are summarized in the following table:

Table 3: Verifying Node Disjoint Paths in LEA

m	Source	Dest.	Set of node disjoint paths (η)
2	P_0	P_{11}	{$P_0 - P_{11}$, $P_0 - P_1 - P_{10} - P_{11}$, $P_0 - P_9 - P_8 - P_{11}$, $P_0 - P_3 - P_2 - P_{11}$}
	P_1	P_7	{$P_1 - P_{10} - P_7$, $P_1 - P_2 - P_3 - P_6 - P_7$, $P_1 - P_4 - P_7$, $P_1 - P_0 - P_9 - P_8 - P_7$}
3	P_0	P_{17}	{$P_0 - P_{17}$, $P_0 - P_{15} - P_{16} - P_{17}$, $P_0 - P_3 - P_2 - P_{17}$, $P_0 - P_1 - P_4 - P_5 - P_8 - P_{11} - P_{14} - P_{17}$ }
	P_2	P_{12}	{$P_2 - P_3 - P_6 - P_9 - P_{12}$, $P_2 - P_5 - P_8 - P_{11} - P_{12}$, $P_2 - P_1 - P_0 - P_{15} - P_{12}$, $P_2 - P_{17} - P_{14} - P_{13} - P_{12}$}
4	P_0	P_{23}	{$P_0 - P_{23}$, $P_0 - P_{21} - P_{22} - P_{23}$, $P_0 - P_3 - P_2 - P_{23}$, $P_0 - P_1 - P_4 - P_5 - P_5 - P_8 - P_{11} - P_{14} - P_{17} - P_{20} - P_{23}$ }
	P_3	P_{17}	{$P_3 - P_2 - P_{23} - P_{20} - P_{17}$, $P_3 - P_4 - P_7 - P_{10} - P_{13} - P_{16} - P_{17}$, $P_3 - P_0 - P_{21} - P_{18} - P_{17}$,$P_3 - P_6 - P_9 - P_{12} - P_{13} - P_{14} - P_{17}$}
…	…	…	…
…	…	…	…

In each observation we could find only four node disjoint paths.

Proposition 7: *LEA(m) is 4-connected i.e. the connectivity of LEA (κ(LEA)) is 4.*
Proof:

This proposition can be proved using the concepts of graph theory. From Whitney's Theorem, we know that a graph with at least (κ+1) vertices is κ-connected if, and only if, any two distinct vertices in the graph are connected by at least κ-node disjoint paths. In proposition 6, we have shown that 4 node disjoint paths exist between any two distinct nodes of LEA, for all levels. All levels of LEA consist of more than 5 nodes, so the connectivity of LEA(m) is 4.

Proposition 8: *The 3-fault diameter, $D_4(LEA)$ of LEA(m) is (m+2).*

Proof:

In an interconnection network, when some nodes become faulty the diameter of the faulty network may be different from that of the original network. A number of researchers have investigated the design of fault tolerant interconnection networks. A common notion of fault tolerance is based on the connectivity of the underlying graph G. This notion is denoted as $D_\kappa(G)$ and is referred to as the κ-1 fault diameter of G, where κ, is the connectivity of G.

The 3-fault diameter for LEA(m) network can be calculated as follows:

Table 4: 3 - fault diameter for various levels of LEA

m	Actual Diameter	3 – fault Diameter
2	3	4 = 3+1
3	4	5 = 4+1
4	5	6 = 5+1
…	…	…
…	…	…
m	m+1	m+2

Proposition 9: *LEA is highly scalable.*

Proof:

In LEA, moving from level k, to the next level, k+1, only an arm of six processors is added and the connections are adjusted according to the connectivity formulae. This feature provides a great deal of flexibility in node representation. For example using LEA it is very easy to design a network consisting of 24 (LEA(4)) nodes or 120 (LEA(20)) nodes, however networks with 24 nodes or 120 nodes can't be designed using hypercube.

4. ROUTING ON LEA NETWORKS

Routing involves the process of identification of a set of permissible paths that may be used by a message to reach its destination, and a function that selects one path from the set of permissible paths. Routing Algorithm is an important factor which affects the performance of an interconnection network. The following sub-sections describe the unicast and broadcast routing algorithms on LEA.

4.1 Unicast Routing Algorithm

For unicast routing on an LEA interconnection network, we present a routing algorithm based on deterministic (oblivious) routing technique [24]. The proposed algorithm makes good use of the topological properties of LEA architecture and is straightforward. Given a source processor P_s and a destination processor P_d, the algorithm attempts to find the shortest path along all the four nodes connected to P_s, using four different variables. When the destination is achieved along a particular path, the algorithm terminates declaring the path in question as shortest path and avoiding rest of the paths. The message is forwarded along the path returned by the algorithm as the shortest path. Figure 3 shows the **PSUDO code** for the proposed algorithm.

Algorithm: Finding Shortest Path and its length in LEA

N: Number of Nodes in underlying LEA network.
P_s: Source Processor
P_d: Destination Processor
X_i: Variable to record movements along node disjoint path i.
L_i: List of nodes traversed along node disjoint path i.
insert(L, p): A function that inserts a node p in the list L.
forward(L): A function that forwards a message from source to destination along the path contained in L.
sp: Shortest Path
spl: Length of Shortest Path
ℓ(L): Last node in the list L.

Require: N, s, d

```
procedure_rout(Ps, Pd)
begin
spl ← 1;
for i =1 to 4 do
insert(Li, Ps);
end for
X1 ← s-3; if (X1 < 0) then X1 ← N+X1;
X2 ← s+3; if (X2 >= N) then X2 ← |N-X2|;
X3 = s+1; if (X3 >= N) then X 3 ← |N-X3|;
X4 ← s-1; if (X4 < 0) then X4 ← N+X4;
for i =1 to 4 do
insert(Li, PXi );
end for
while true do
begin
if (X1= d OR X2 = d OR X3 = d OR X4 = d) then
    break;
diff ← (d-X1)
if (diff >= 3) then X1 ← X1-3;
else
{
if (diff > 0) then X1 ← X1+1;
else X1 ← X1-1;
}
if (X1 < 0) then X1 ← N+X1;
diff ← (d-X2)
if (diff >= 3) then X2 ← X2+3;
else
{
if (diff > 0) then X2 ← X2+1;
else X2 ← X2-1;
}
if (X2 >= N) then X2 ← |N-X2|;
X3 ← X3+1; if (X3 >= N) then X 3 ← |N-X3|;
X4 ← X4-1; if (X4 < 0) then X4 ← N+X4;
for i =1 to 4 do
insert(Li, PXi );
```

37. **end for**
38. spl ← spl + 1;
39. **end while**
40. **for** i=1 to 4 **do**
41. **if** ($\ell(L_i) = P_d$) **then** sp ← L_i;
42. **end for**
43. forward(sp)**;**
44. **end**

Figure 3: Shortest Path in LEA Network

Example: Let in a 18-node LEA i.e. LEA(3), processor P_2 needs to route a message to processor P_{15} then, we have the following trace of above algorithm:

Input:

N = 18, s = 2, d = 15

Steps (2 – 10):

$L_1 = L_2 = L_2 = L_3 = P_2$, $X_1 = 17$, $X_2 = 5$, $X_3 = 3$, $X_4 = 1$, spl = 1.

Steps (11 – 13):

$L_1 = P_2P_{17}$, $L_2 = P_2P_5$, $L_3 = P_2P_3$, $L_4 = P_2P_1$.

Steps (14 – 39):

X_1	X_2	X_3	X_4	L_1	L_2	L_3	L_4	Spl
16	8	4	0	$P_2P_{17}P_{16}$	$P_2P_5P_8$	$P_2P_3P_4$	$P_2P_1P_0$	2
15	11	5	17	**$P_2P_{17}P_{16}P_{15}$**	$P_2P_5P_8P_{11}$	$P_2P_3P_4P_5$	$P_2P_1P_0P_{17}$	**3**

Steps (40 – 44):

List L_1 is chosen as the list consisting of the shortest path and the message is routed from P_2 to P_{15} along the path contained in L_1.

It may be easily calculated that the above algorithm is a liner time i.e. o(N) algorithm.

4.2 Broadcasting on LEA

In some specific situations a particular node of an interconnection network needs to distribute its data to all other nodes on the network. This operation is known as broadcasting.

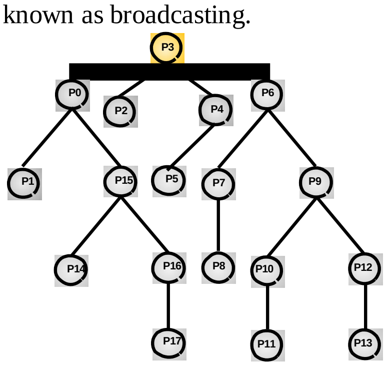

Figure 4: Minimum Broadcast Spanning Tree for LEA(3)

The minimum broadcast spanning tree of an interconnection network can be constructed by finding the shortest paths between source node and all the other nodes. Moreover, the height of a broadcast tree of an interconnection network is at the most equal to its diameter.

In previous sub-section we have presented an optimal routing algorithm for LEA network. Using the same algorithm, broadcasting algorithm can easily be developed for LEA interconnection network. Figure 4 shows the minimum broadcast spanning tree for an 18-node LEA, i.e. LEA(3), considering P_3 as the source node.

5. APPLICATIONS OF LEA

This section discusses how LEA can be used to design cost-effective product networks whether the degree x diameter or diameter x number of links is used as a desirable quality measure of a topology. We start with designing a new product topology called LEAH (LEA-Hypercube). LEAH is a product graph of LEA(m) and Q_n. The topological properties of LEAH along with other topologies of interest have been shown in table 5.

Table 5: LEAH and Other Topologies of Interest

Topological Property	DLH(m, n) [14]	Hyper-Mesh HM(n,r,c) [2]	Hex-cell HC(i) [3]	LEAH(m, n)
No. of Nodes (N)	$m.2^{n+2}$	$(r.c).2^{n}$	6(2i-1)	**$3m.2^{n+1}$**
Degree (d)	3+n	4+n	$4\sqrt{2i-1}-1$	**4+n**
Diameter (D)	m+n+1	n+r+c-2	3	**m+n+1**
No. of Links (L)	$m.2^{n+1}.(3+n)$	$2^{n-1}.(r.c).(4+n)$	18i-12	**$3m.2^{n}.(4+n)$**
Cost (D x d)	(3+n)(m+n+1)	(n+r+c-2).(n+4)	$3.(4\sqrt{2i-1}-1)$	**(4+n).(m+n+1)**
Extended Cost (L x D)	$(m+n+1).m.2^{n+1}.(3+n)$	$2^{n-1}.(r.c).(4+n).(n+r+c-2)$	$(4\sqrt{2i-1}-1).(18i-12)$	**$(m+n+1).3m.2^{n}.(4+n)$**

These properties of LEAH can be obtained using the results of the generalized framework on product networks [22], [23]. Then, we conduct a comparative study between five interconnection networks namely Hypercube, Double Loop Hypercube (DLH(m, n)), Hyper-Mesh (HM(n,r,c)), Hex-Cell and LEAH. We have used the most common parameters such as network size support (scalability), degree, diameter, cost (degree x diameter) and extended cost (diameter x number of links), which have been extensively used in the literature [1], [2], [3], [20] to evaluate the interconnection networks. Figures 5 - 8 show the comparison of the different topologies in terms of the parameters scalability, node degree, diameter, cost and extended cost. We can readily make the following observations:

Scalability (Network Size Support): To fairly compare the network size support capability (scalability) of each topology, we define the following criteria as the percentage of flexibility allowed in network size:

$$\%\text{Flexibility} = \frac{|\text{Desired Size} - \text{Available Size}|}{\text{Desired Size}} \times 100$$

with DLH and Hex-Cell. It is much better than the scalabilities of Hyper-Mesh and Hypercube. Almost all the network sizes (in the range 1 – 10000) can be represented with LEAH with a flexibility of 10% in the network size representation.

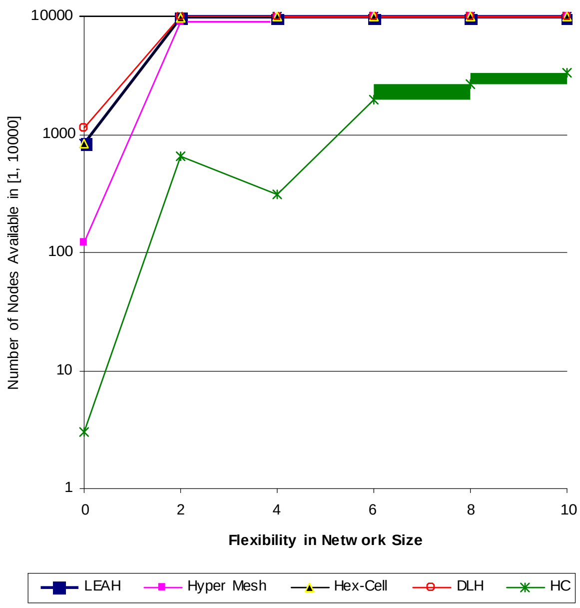

Figure 5: Network Size Support Comparison

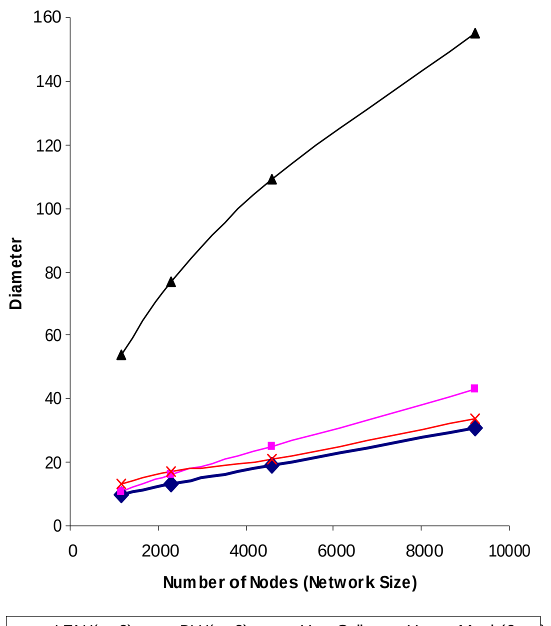

Figure 6: Network Diameter Comparison

Degree and Diameter: Except Hex-Cell, the degree of all the topologies shown in table 5 depends on the degree of the hypercube part used to design a particular topology. To make the comparison fair we have fixed the dimension of hypercube to 6. In this case DLH has the lowest degree (9) compared to the topologies).

Figure 6 shows the comparison of diameters for various topologies as a function of network size (number of processors). From figure 6 it is obvious

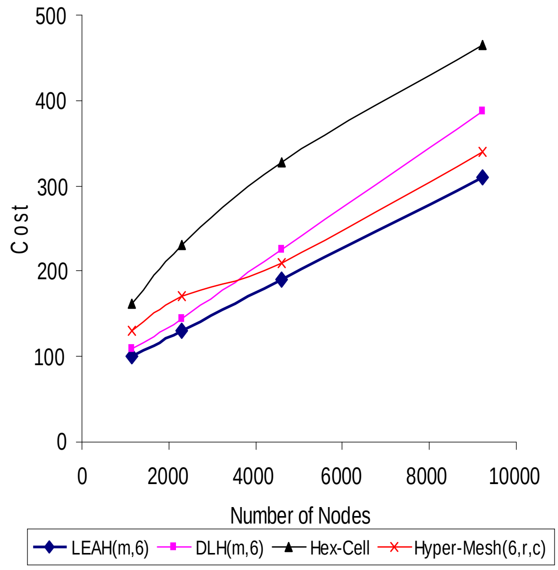

Figure 7: Network Cost Comparison

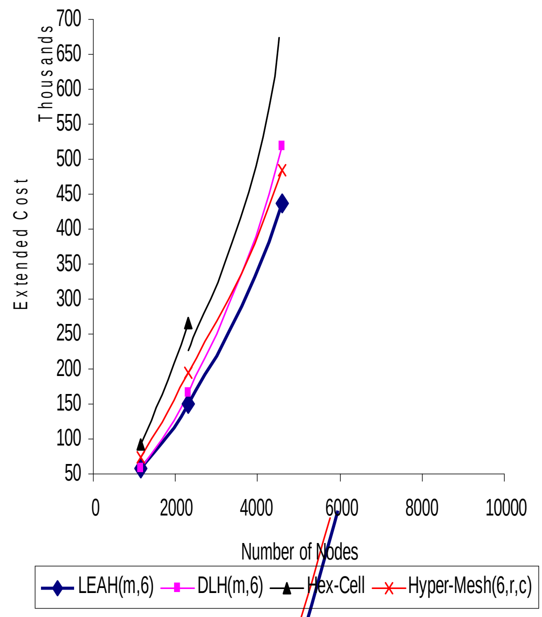

Figure 8: Extended Cost (Diameter x Number of Links) Comparison

that the diameter of LEAH for a given degree of hypercube is lowest in the family even up to 10000 nodes. The diameter may further be reduced by choosing m and n intelligently.

Cost and Extended Cost: Figure 7 and Figure 8 compare the topological network cost (degree x number of links) for various topologies of interest for a given degree of hypercube. The two figures indicate that topological network cost and the extended network cost is lowest for LEAH among all the topologies of interest.

For very large network sizes, it may be possible that the network cost or extended cost of Hex-Cell or Hyper-Mesh may supersede the cost of LEAH. In that case the cost may be reduced by decreasing the LEA component (m) and by increasing the hypercube component (n). As LEAH is highly scalable (comparable to Hex-Cell and better than Hyper-Mesh), all the network sizes that may be represented by Hex-Cell or Hyper-Mesh can also be represented by LEAH.

6. CONCLUSIONS

In this paper we have introduced and analyzed a new interconnection topology named as Linearly Extendable Arm (LEA). The proposed topology has some attractive features like constant degree, regularity, high scalability and symmetry. A simple routing algorithm which makes a good use of LEA architecture has also been proposed. We have also discussed the broadcasting procedure for LEA and have obtained a minimum broadcast spanning tree for it. Moreover, it has been shown that LEA is particularly useful for designing scalable and cost effective product networks. Product networks based on LEA can maintain a constant degree over a large number of nodes and they outperform other topologies of interest when either the degree x diameter or diameter x number of links is used as a desirable quality measure of an interconnection network.

References

[1]. A. Ayyoub and K. Day: The Hyperstar Interconnection Network, Journal of Parallel and Distributed Processing, Vol. 48, pp. 175-199 (1998).

[2]. E. Abuelrub: A Comparative Study on Topological Properties of the Hyper-Mesh Interconnection Network, Proceedings of the World Congress on Engineering, Vol. I, pp. 616-621 (2008).

[3]. Sharieh, M. Qatawneh, W. Almobaideen and A. Sleit: Hex-Cell – Modeling, Topological Properties and Routing Algorithm, European Journal of Scientific Research, Vol. 22 No.2, pp. 457-468 (2008)

[4]. Y. Saad and M. Schultz, Topological Properties of the Hypercube, IEEE Transactions on Computers, vol. C-37, No. 7, pp. 867-872 (1988).

Vectors: A New Fault-Tolerant Routing for Binary n-Cubes, Journal of Systems Architecture, vol. 47, No. 9, pp-783-793 (2002).

[6]. G. Chiu and K. Chon: Efficient Fault-Tolerant Multicast Scheme for Hypercube Multicomputers, IEEE Transactions on Parallel and Distributed Systems, vol. 9, No. 10, pp. 952-962 (1998).

[7]. R. Klasing: Improved Compressions of Cube-Connected Cycles Networks, IEEE Transactions on Parallel and Distributed Systems, vol. 19, No. 8, pp. 803-812 (1998).

[8]. A. Louri and H. Sung: Scalable Optical Hypercube- Based Interconnection Network for Massively Parallel Computing, Applied Optics, vol. 33, pp.7588-7598 (1994).

[9]. A. Louri, B. Weech, C. Neocleous: A spanning multichannel linked hypercube: a gradually scalable optical interconnection network for massively parallel computing, IEEE Transactions on Parallel and Distributed Systems, Vol. 9, Issue 5, pp.497-512 (1998).

[10]. K. Efe: The Crossed Cube Architecture for Parallel Computation, IEEE Transactions on Parallel and Distributed Systems, Vol 3, pp. 513 (1992).

[11]. C. Chen, D. P. Agrawal and J R Burke: dB-cube: a New Class of Hierarchical Multiprocessor Interconnection Network with Area-efficient Layout, IEEE Transactions on Parallel and Distributed Systems, Vol 4, pp. 1332 (1993).

[12]. Q. M. Malluhi and M A Bayoumi: The Hierarchical Hypercube: A New Interconnection Topology for Massively Parallel Systems, IEEE Transactions on Parallel and Distributed Systems, Vol. 5, pp. 17 (1994).

[13]. F. P. Preparata and J. Vuillemin: The cube-connected cycles: a versatile network for parallel computation, Communications of the ACM, Vol. 24(5), pp.300-309 (1981).

[14]. L. Youyao, H. Jungang, D. U. Huimin: A Hypercube Based Scalable Interconnection Network for Massively Parallel System, Journal of Computers, Vol. 3, No. 10, pp. 58-65 (2008)

[15]. A. Awwad, A. A. Ayyoub and M. Ould-Khaoua: On Topological Properties of Arrangement Star Network, Journal of Systems Architecture, No. 48, pp. 325-336 (2003).

[16]. K. Day and A. Tripathi: A Comparative Study of Topological Properties of Hypercubes and Star Graphs, IEEE Transactions on Parallel and Distributed Systems, Vol. 5 No. 1, pp. 31-38 (1994).

[17]. S. Zheng, B. Cong and S. Btttayeb: The Star-Hypercube Hybrid Interconnection Networks, Proceedings of the ISCA International

Conference on Computer Application in Design, Simulation and Analysis, pp. 98-101 (1993).

[18]. E. Abuelrub: On Hyper-Mesh Multicomputers, Journal of the Institute of Mathematics and Computer Sciences, Vol. 12 No. 2, pp. 81-83 (2002).

[19]. A. Youssef and B. Narahari: The Banyan-Hypercube Networks, IEEE Transactions on Parallel and Distributed Systems, Vol. 1 No. 2, pp. 160-169 (1990).

[20]. S. K. Das and A. K. Banerjee: Hyper Perersen Network: Yet Another Hypercube-Like Topology, Proceedings of Frontiers '92, pp. 270-277 (1992).

[21]. G. Chartrand and L. Lesniak: Graph and Diagraph (Second Edition), Wadsworth & Books, California, (1986).

[22]. Youssef: Design and Analysis of Product Networks, Proceedings of Frontiers '95. Fifth Symposium, Frontiers of Massively Parallel Systems (1995).

[23]. K. Day and A. A. Ayyoub: The Cross Product of Interconnection Networks, IEEE Transactions on Parallel and Distributed Systems, Vol. 8 No. 2, pp. 109-118 (1997).

[24]. H. E. Rewini and M. A. Barr: Advanced Computer Architecture and Parallel Processing (First Edition), Wiley-Interscience, Canada (2005).