A Statistical Analysis and Comparative Study of Embedded Hypercube

Mohd.Kalamuddin Ahmad
Research Scholar
Department of CS
Mewar University
Chhittorghara, India

Mohd. Husain, PhD
Prof.Department of CSE, Uttar PradeshTech.University
Lucknow, India

A.A. Zilli
Prof.Department of CSE, Integral University,
Lucknow, India

ABSTRACT

Consider the K-arry n-cube network is the most significant network structure in parallel computer architecture. Therefore the generic structure of K-arry n-cube used to design the various networks with static network topology of different parameters, and its makes different types hypercube networks ($N=2^n$) from static networks known as embedded hypercube scalable inter-PE connection network suitable for parallel computing systems. Show how to design the good interconnection network in parallel Architecture for less density as well as diameter to route the data to different path. Also it has been proved with the computational results of entire system that the embedded hypercube interconnection networks built is highly scale up in terms of communication. A complete design analysis and comparison of network with various other networks is given using different network parameters, and statistical analysis comparative optimal of torus architecture rather than mesh architecture.

Keywords

Topology, Data routing path, Embedded network, Hypercube network, Network parameters and Scalability, Network Metrics, Statistics.

1. INTRODUCTION

Parallel processing is an integral part of everyday life. The concept is so inbuilt in our existence that we benefit from it without realizing. When faced with a tough problem, we involve others to solve it more easily. This co-operation of more than one worker to facilitate the solution of a particular typical problem may be equivalent as parallel processing. The goal of parallel processing is thus to

Solve a given problem more rapidly, or to enable the solution of a problem that would otherwise be impracticable by a single worker. The way the nodes are connected to one another various among machines. In direct network architecture, each node has a point to point, or direct, connection to some number of nodes, called the neighboring nodes. Direct network have become popular architecture for constructing massively parallel computers because they scale well, that is the number of nodes in the system increases as well as communication bandwidth and processing capability of the system also increase [1] [2][3].

In direct network architecture, each node has a *point-to-point*, or direct, connection to some number of other nodes, called neighboring nodes. Neighboring nodes may send packets to one another directly, while nodes that are not directly connected must rely on other nodes in the network to transfer packets from source to destination. Although a router's function could be performed by the corresponding local processor, dedicated routers are used to allow overlapped. Computation as well as communication within each node, router supports some number of input and output channels. Internal channels connect the local processor memory to the router. External channels are used for communication between routers, and, therefore nodes. By connecting the input channels of one node to the output channels of other nodes, the *topology* of the direct network will be defined. For topologies in which packets may have to traverse some intermediate nodes, the *routing algorithm* determines the path selected by a packet to reach its destination. At each node, the routing algorithm indicates the next channel to be used. Efficient routing is critical to the performance of interconnection network [1],[6], [7],[9] and [10].

In this paper the proposed work how to route the data in massive parallel computer architecture with respect to reduce the diameter and comparatively statistical analysis.

2. ARCHITECTURAL PROPERTIES OF EMBEDDED

The interconnection network is a vital role in a parallel processing. A good interconnection network is expected to have least number of links, topological network cost and more reliable. The interconnection network must be able to built scale up .The data routing functions in embedded hypercube network could be analyzed [3]-[4,[5] and [9].

2.1 Mesh Embedded

A Single stage recirculating represent network. In network, each PE_i is allowed to send to any one of PE_{i+1}, PE_{i-1}, PE_{i+r} and PE_{i-r}, Where r= N in one circulating steps through entire network [1] and [8].

The following routing function from (1) - (5) apply on simple mesh network [8] –

$R_{+1}(i) = (i+1) \bmod N$ (1)

$R_{-1}(i) = (i-1) \bmod N$ (2)

$R_{+r}(i) = (i+r) \bmod N$ (3)

$R_{-r}(i) = (i-r) \bmod N$ (4)

$R_C(k_{n-1} \ldots k_{d+1}\, k_d\, k_{d-1} \ldots k_0) = (k_{n-1} \ldots k_{d+1}\ \ k'_d\ \ k_{d-1} \ldots k_0)$ (5)

Where $0 \leq i \leq N-1$, Commonly N know as perfect Square.

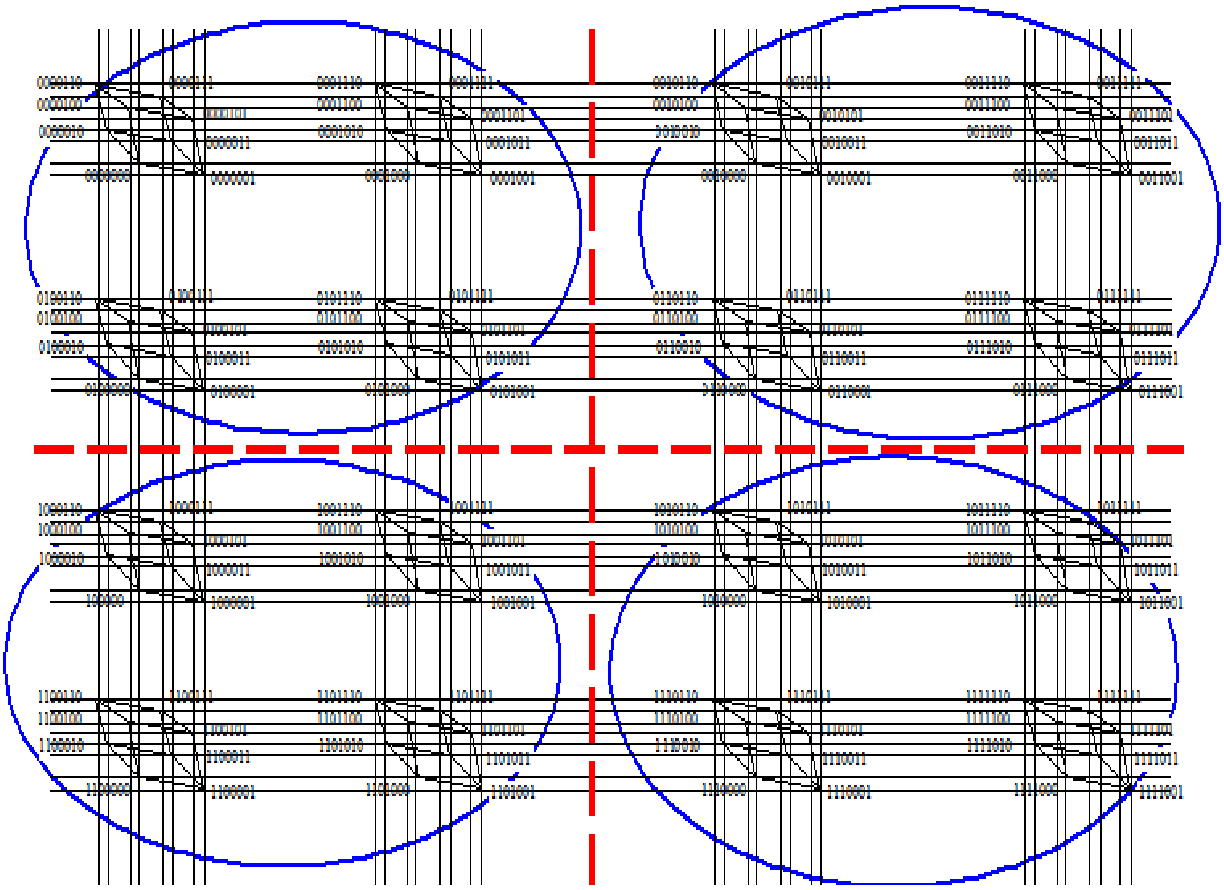

Figure: 1. (4, 4, 8)   Mesh Embedded Architecture

2.2 Torus Embedded

Now suppose consider $a \times b$ be the size of some concurrent torus networks with a number of rows and b number of columns and N being the number of nodes connected in the hypercube, the torus embedded hypercube network can be designed with the size of (a, b, N). Nodes with identical positions in the torus networks will be a group of N number of nodes connected in the hypercube configuration and can be addressed with three parameters such as row number i, column number j of torus and address of node k in hypercube where the addressed node is residing. Hence, a (a, b, N)–torus embedded hypercube network will have a × b × N number of nodes and a node with address as (i, j, k) where $0 \leq i <$ a, $0 \leq j <$ b and $0 \leq k < N$. The data routing functions of torus embedded hypercube as follow-

$$R_1\ (i,\ j,\ k)\ = (\ i,\ (\ j+1)\ \text{mod}\ b,\ k\ ) \qquad (6)$$

$$R_2\ (i,\ j,\ k)\ = (\ i,\ (\ b+j-1)\ \text{mod}\ b,\ k\ ) \qquad (7)$$

$$R_3\ (i,\ j,\ k)\ = ((\ i+1)\ \text{mod}\ a,\ j,\ k\ ) \qquad (8)$$

$$R_4\ (i,\ j,\ k)\ = ((a+i-1)\ \text{mod}\ a,\ j,\ k\ ) \qquad (9)$$

And also used hypercube routing function equation (5) as above mentioned in the mesh embedded.

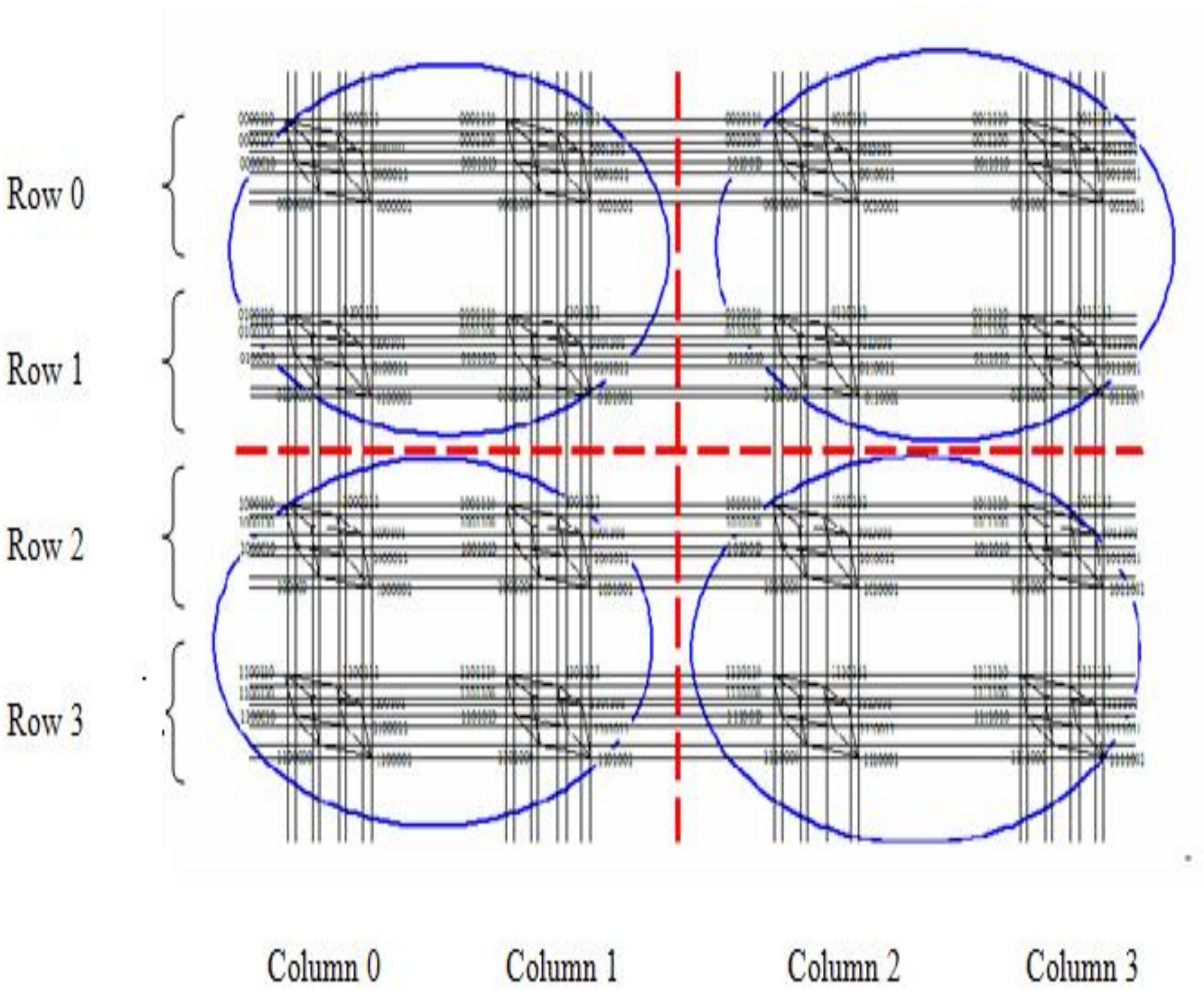

Figure: 2. (4, 4, 8) Torus Embedded Hypercube Network

The end to end connections of row and column of each connected in torus but are not represent in Figure 2. . A wraparound connection is making along each row or column if they have equal label as a complete of torus embedded hypercube network (4, 4, 8).

3. RESULTS AND DISCUSSION

3.1 Node degree

Table-1 and figure 3, Gives the comparison of node degree as a function of the number of processors of various other popular networks for parallel architecture.

Table 1: Comparative Results of Node Degree, For Networks Topology

Types of Network	Number of Processors					
	128	256	512	1024	2048	4096
2D-Torus	4	4	4	4	4	4
2D-Mesh	4	4	4	4	4	4
Linear Array	2	2	2	2	2	2

Ring	2	2	2	2	2	2
Hypercube	7	8	9	10	11	12
K-arry-n Cube	14, k=2	16,k=2	18, k=2	20,k=2	22,k=2	24, k=2

The comparison shows that the n-cube hypercube and K-arry n-cube interconnection networks are expensive and not suitable for parallel architecture. For these two networks, the node degree increases dramatically as the system expansion takes place. Obviously the cost per node also increases tremendously for these two networks as the system is scaled up shown in figure 3.

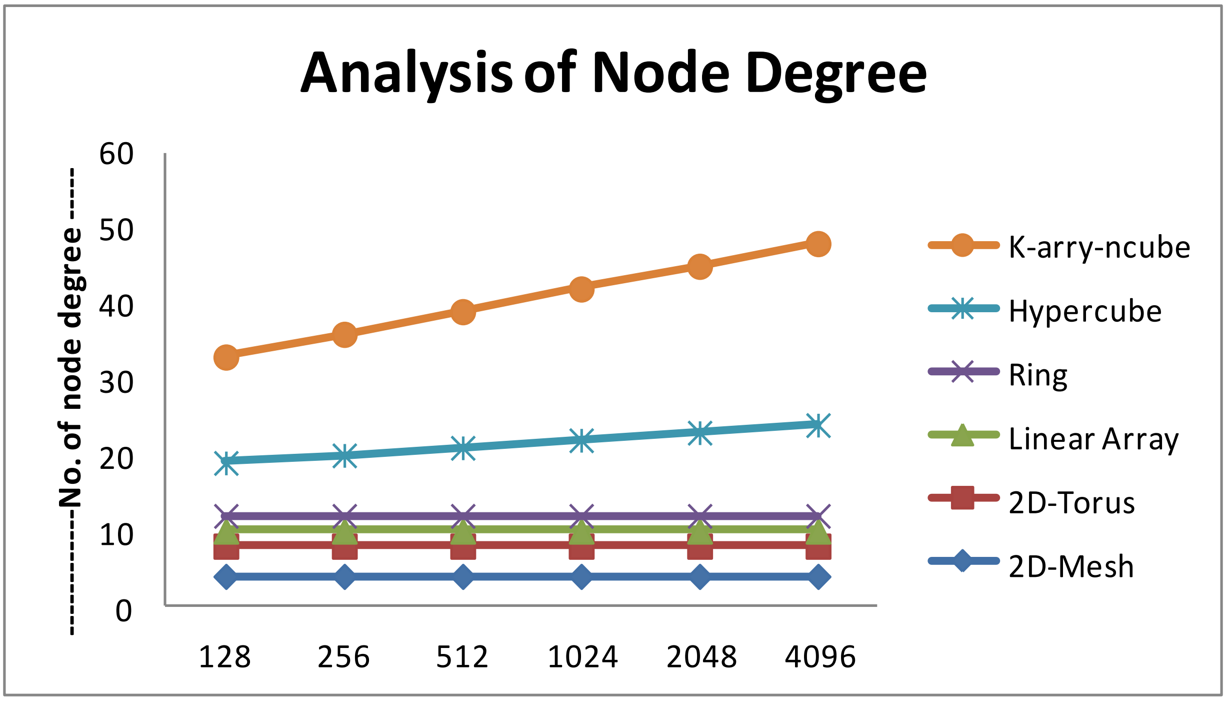

Figure: 3. Node degrees Analysis

3.2 Total Number of Links

Table 2: Comparative Results of Number of Links for Network Topology

Types of Network	**Number of Processors**					
	128	256	512	1024	2048	4096
2D-Torus	256	512	1012	2048	4096	8192
2D-Mesh	234	480	980	1984	4006	8064
Linear Array	127	255	511	1023	2047	4095
Ring	128	256	512	1024	2048	4096
Hypercube	64	128	256	512	1024	2048
K-arry-n Cube	896,k=2	2048,k=2	4608,k=2	10240,k=2	22528,k=2	28672,k=3

Table-2 and Figure 4 gives the comparison of Total number of wires or links for a PE-Interconnection network is expected to be in above table –

In table to reflects link complexity and directly related to the cost of the networks.

Table 2 shows the number of links with respect to the scaling design of simple network in the parallel computing architecture. It is observed that the total number of connection for two Dimensional static topology like Mesh, Linear array, and ring lies in between n-cube hypercube and two dimensional Torus network. It is to be noted that the torus offers larger number of links network as shown in figure 4.

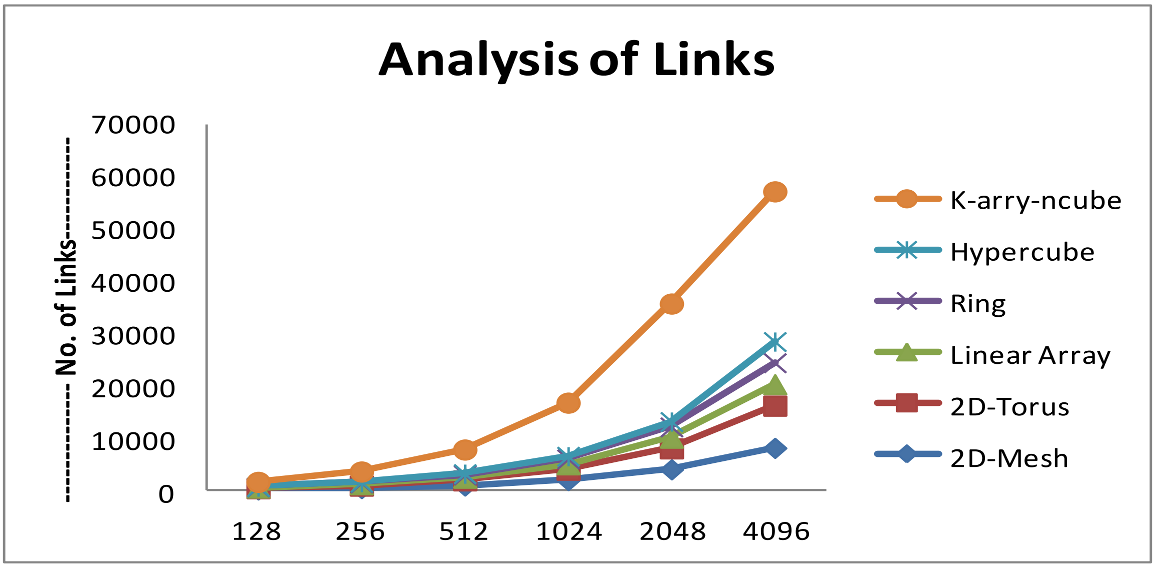

Figure: 4. Analysis of links

3.3 Network Diameter Analysis

Table-3 and Figure 5 gives the comparison of network diameter as a function of the number of processors of various other popular networks along with the embedded hypercube network worth to mention that due to their bottleneck performance as per the last comparisons, the mesh torus and mesh networks have been dropped from further comparisons.

Table 3: Comparative Results of Diameter for Embedded Network

Types of Network	Number of Processors					
	512	1024	2048	4096	8192	16386
(8,8, N) Mesh Embedded Hypercube	17,N=8	18,N=16	19,N=32	20,N=64	21,N=128	22,N=256
(16,16, N) Mesh Embedded Hypercube	31,N=2	32,N=4	33,N=8	34,N=16	35,N=32	36,N=64
(32,32, N) Mesh Embedded Hypercube	---	64,N=1	65,N=2	66,N=4	67,N=8	68,N=16
(8,8, N) Torus Embedded Hypercube	11,N=8	12,N=16	13,N=32	14,N=64	15,N=128	16,N=256
(16,16, N) Torus Embedded Hypercube	17,N=2	18,N=4	19,N=8	20,N=16	21,N=32	22,N=64
(32,32,N)Torus Embedded Hypercube	---	32,N=1	33,N=2	34,N=4	35,N=8	36,N=16

In the results given in Table 3 and Figure 5 as far as the network diameter is concerned, the torus embedded hypercube network required needs minimum network diameter to get interconnected between a source node and a destination node.

We mentioned the required the total diameter from source to destination is-

Mesh **D** $_{\mathbf{s\text{-}d}}$ = Diameter of two dimensional mesh + Diameter of hypercube

$$= \mathbf{2(\sqrt{p} - 1) + \log p'} \qquad (10)$$

Where p is no. of processor in two dimensional mesh, and p' is no. of processor in hypercube.

We are also mentioned here Torus embedded hypercube diameter from source to destination-

Torus **D** $_{\mathbf{s\text{-}d}}$ = Diameter of two dimensional torus + Diameter of hypercube

$$= \mathbf{\sqrt{p} + \log p'} \qquad (11)$$

Where p is no. of processor in two dimensional Torus, p' is no. of processor in hypercube. (Here we mentioned the total number of processors in two dimensional mesh or torus embedded hypercube = p + p')

In comparatively analysis –diameter of (8, 8, N) mesh embedded hypercube system are same as (16, 16, N) torus embedded hypercube system, and also diameter of (16, 16, N) mesh embedded hypercube system are same as (32, 32, N) torus embedded hypercube system, shown in figure 5.

Hence the torus embedded hypercube network is much superior than mesh embedded hypercube network as far as performance metrics and speed up than mesh embedded hypercube.

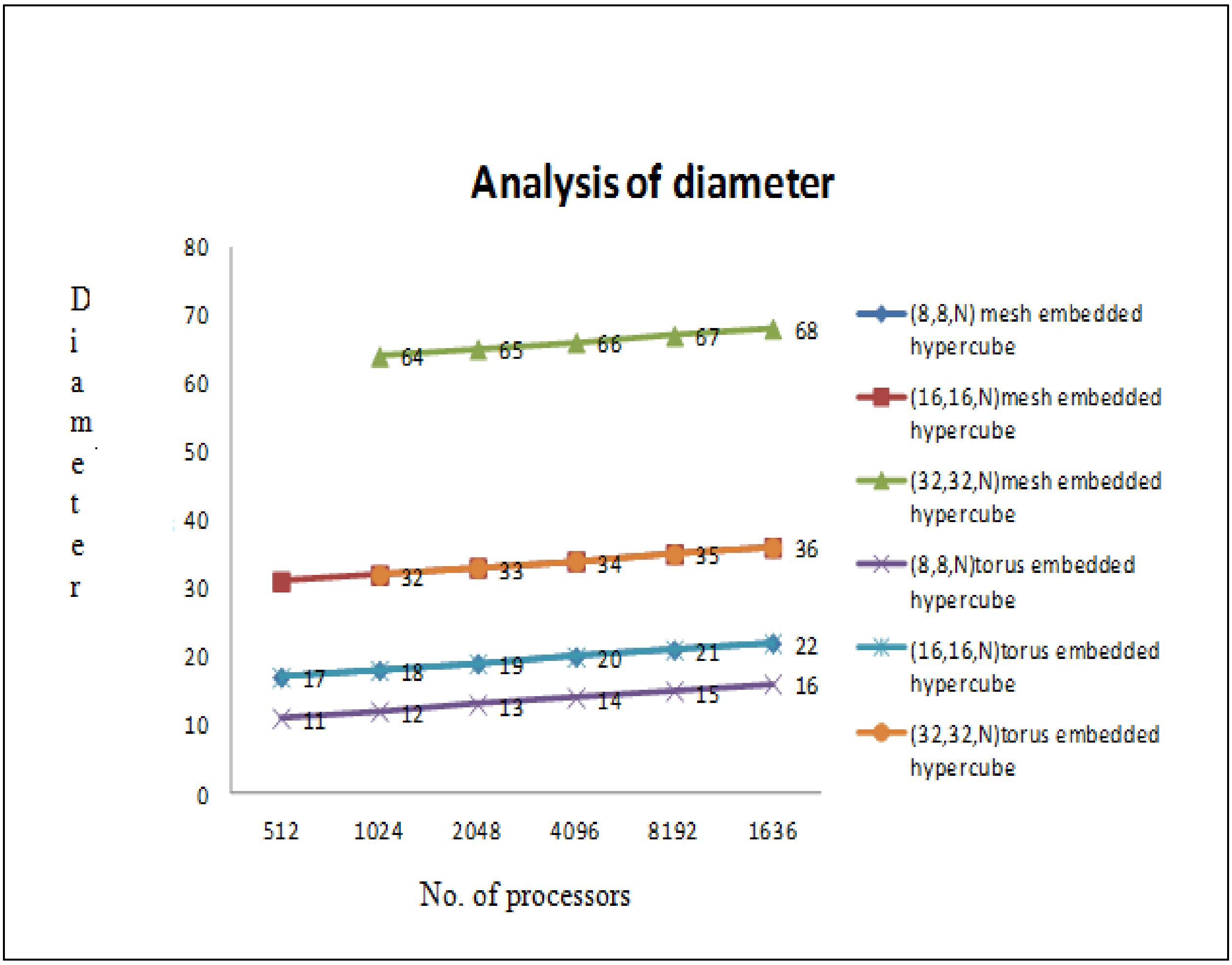

Figure: 5. Analysis of diameter

4. STATISTICAL ANALYSIS

Table-4 gives the descriptive statistical comparison of different network topology with respect to diameter as mean ,std deviation ,std error confidence level of mean and minimum and maximum boundary of diameter.

Table 4 shows the mean, std deviation, error, confident interval of mean and minimum and maximum density with respect to number of processors, comparatively optimal of torus architecture rather than mesh architecture.

Table 4: Descriptive Results of Diameter for Embedded Networks

Topology	Mean	Std. Deviation	Std. Error	95% Confidence Interval for Mean		Minimum	Maximum
				Lower Bound	Upper Bound		
(8,8, N) Mesh Embedded Hypercube	19.5000	1.87083	.76376	17.5367	21.4633	17.00	22.00
(16,16, N) Mesh Embedded Hypercube	33.5000	1.87083	.76376	31.5367	35.4633	31.00	36.00
(32,32, N) Mesh Embedded Hypercube	55.0000	26.98148	11.0151	26.6847	83.3153	.00	68.00
(8,8, N) Torus Embedded Hypercube	13.5000	1.87083	.76376	11.5367	15.4633	11.00	16.00
16,16, N) Torus Embedded Hypercube	19.5000	1.87083	.76376	17.5367	21.4633	17.00	22.00
(32,32, N) Torus Embedded Hypercube	28.3333	13.95230	5.69600	13.6913	42.9754	.00	36.00
Total	28.2222	18.02080	3.00347	22.1249	34.3196	.00	68.00

5. CONCLUSION

The proposed network is a combination of hypercube and torus network topologies. The analysis results show that torus embedded hypercube interconnection network is highly scalable, this network structure provides a great architectural support for parallel processing. The growth of the network is more efficient in terms of communication and speed up. The comparison shows that the n-cube hypercube and K-arry n cube interconnection networks are expensive and not suitable for parallel architecture. For these two networks, the node degree increases dramatically as the system expansion takes place. The number of links with respect to the scaling design of simple network in the parallel computing architecture. It is observed that the total number of connection for two Dimensional static network like Mesh, Linear array, and ring lies in between n-cube hypercube and two dimensional Torus network. It is to be noted that the torus offers larger number of links network. It is preferred to have a network with a network diameter of minimum value. The result of comparison shows that Torus and mesh network has the large network diameter. Further, for this network, the network diameter increases tremendously as the system is scaled up. The comparative statistical analysis shows that on some parameter torus architecture better than mesh architecture.

6. REFERENCES

[1] K. Hwang, *Advanced Computer Architecture: Parallelism, Scalability, Programmability***.** New York McGraw-Hill,1993.

[2] Kim Jong-Seok, Lee Hyeong-Ok and Heo Yeong-Nam**,** “Embedding among HCN(n,n), HFN(n,n) and hypercube” **,** *Proceedings of Eighth International conference on parallel and distributed systems (ICPADS)-2001* pp 533 – 540, 26-29 June 2001.

[3] A. Louri and H. Sung, “An Optical Multi-Mesh Hypercube: A Scalable Optical Interconnection Network for Massively Parallel Computing,” *IEEE J. Lightwave Technology*, vol. 12, pp. 704–716, Apr. 1994.

[4] N. Gopalakrishna Kini, M. Sathish Kumar and Mruthyunjaya H.S., “A Torus Embedded Hypercube Scalable Interconnection Network for Parallel Architecture,” IEEE explore conference publications, Mar.2009, pp.858-861.

[5] Ahmed Louri and Hongki Sung, “An Optical Multi-Mesh Hypercube: A Scalable Optical interconnection Network for Massively Parallel Computing,” Journal of Light wave Technology, Vol.12, No.4, Apr.1994, pp.704-716.

[6] N. Gopalakrishna Kini, M. Sathish Kumar and Mruthyunjaya H.S,” Performance Metrics Analysis of Torus Embedded Hypercube Interconnection Network”, International Journal on Computer Science and Engineering Vol.1(2), 2009, 78-80.

[7] L.M. Ni, and P.K. McKinley, “A Survey of Wormhole Routing Techniques in Direct Networks”, IEEE Computer, vol.26, no.2, pp. 62-76, Feb. 1993.

[8] Kai Hwang, F.A.Briggs,”Computer Architecture and Parallel Processing “, McGraw-Hill International edition.

[9] M. K. Ahamad, M. Husain.” Required Delay of Packet Transfer Model for Embedded Interconnection Network” International Journal of Engineering Research & Technology (IJERT) Vol. 2 Issue 1, January- 2013 ISSN: 2278-0181.

[10] J.F.Fang, J.Y.Hsiao and C.Y.Tang, “Embedding Meshes and TORUS Networks onto degree-four chordal rings,” IEE Proc.-Comput. Digit. Tech., Vol.145, No.2, Mar.1998.