On Graph Theoretical Properties of Extended Double Star Interconnection Network Topology

[1]Sadashiba Pati, [2] Vinay Singh, [3] Nibedita Adhikari,
[1,2]Usha Martin University, Ranchi, India
[3]Utkal University, Vani Vihar, Bhubaneswar, Odisha, India
sadashibapati1@gmail.com
nibedita.cs@utkaluniversity.ac.in

Abstract

The Extended Double star (EDS) parallel interconnection network with a network controller (NC) is a two level hybrid structure. It is a large scale network with the Double star as its basic building block. EDS network has degree *(n!+n+1)* and diameter $\lfloor 3/2(n-1) \rfloor + 2k$ where n and k are the two parameters denoting network dimension and the level of the network respectively. The extended double star network preserves all of the topological characteristics of the base or parent network, unlike the other derived networks. This large scale network is suitable for voluminous computing and communications. The bipanconnectivity and hamiltonian properties of EDS are investigated. The Embedding of a guest graph into another host graph is a very important criteria and establishes the robustness of the later. For the current study of the topological relationship of EDS network with the two main classes of interconnections namely ring and mesh networks is attempted via embedding. Also, the EDS network satisfies the Hamiltonian properties. The upper limit of ring and mesh embedding is also estimated.

Keywords: Embedding, star, ring, mesh.

1. Introduction

A parallel computing system is made up of several processing units that are linked together by an interconnection network and the software that enables the processing units to cooperate. Any parallel computing system's foundation is its connectivity network. Nowadays, big data is just one of many scenarios where massively parallel computing is taking place in the background. The big data concept emphasizes on solving the large problems by dividing them into smaller parts, which are solved simultaneously. It makes the process of finishing the work quicker than serial processing. The most well-known idea in this area is a computer architecture that supports multiple instructions and multiple data (MIMD). Many researchers are experimenting with the basic structure because the Parallel Interconnection Networks (PIN) are in the background of every kind of parallel computing systems. Major PINs are Hypercube[1], Crossed Cube[6], a Star graph[8], Extended Hypercube[3], Extended Star[2], and Double star[11] and Extended Double star[] to name a few.The EDS topology is suggested as an alternative to the star graph with higher number of computing nodes. For the past two decades, techniques for parallel processing have been researched as a means of enhancing computer system performance. Parallelism's popularity and recent breakthroughs in computing have speed up the creation of numerous appealing interconnection networks [4]. An interconnection network (IN) with more processors has more processing power. Multiple programs must leverage parallelism to do this form of a computational system known as parallel computing carries out calculations and message transfer among processors simultaneously. Nowadays, every scenario, including big data, has vast parallel computing running in the background. Networks of connections are used to implement MIMD computers. A machine can transport data from one node to a destination node with a minimum amount of latency in any PIN.The faster message passing or communication process can easily amplify the capability of a distributed shared memory model. Each message sent /received within a communicative entity must be an independent activity. The senders and receivers at the end nodes are where the messages are stored rather than on the communications channel itself. But in the case of shared memory communication, a memory block utilized as a communication device can be thought of as a storage space for all the data. Applications those are readily accepted as candidates for message passing communication are those which allow for relatively independent operation of processing units.Since many decades the message passing in massively parallel computing systems has been a potential area of research. The big data concept has also emphasized communication among the processing units. The parallel interconnection network is the backbone of big data systems.Thus, more attention is always given to faster access of data while storing or/and retrieving data from the data store during processing. In parallel systems where routing and broadcasting are faster and is accomplished at a low cost, then those will be a better candidate for implementation purposes[17]. Any parallel computing system is built on a parallel connectivity network. There should be as little lag as feasible when completing this activity. It should enable numerous such transfers to occur simultaneously. In addition, it should be cheap in comparison to the cost of the remainder of the equipment in the system. A network's topology, routing algorithm, switching scheme, and flow control mechanism are all defining factors. There are many kinds of PIN existing like hypercube and star graphs would be the most favored. Now a day's Big data is an extremely well-liked buzzword and also drawing in massive attention. Interconnection network continues to be mostly accepted to become the best reasonable type of parallel computing. This particular present effort is inspired form Extended Double Star (EDS) network [13]. A Extended double star system includes two star graphs together with a single Network Controller(NC), each one of $(2n!)^k + \frac{(2n!)^k-1}{2n!-1}$ nodes. Every node in an EDS has the same degree (n!+n+1). The embedding characteristics of interconnection networks play an important role, as they facilitate the transmission of messages in the event of faulty nodes. Acquiring a spanning broadcasting tree in the network is an integral part of the broadcasting process. The level of this particular spanning tree is proportional to the diameter of the system, therefore it reveals the entire message transmission time from one node to the rest. Additionally, the robustness of the host system is also exposed through the embedding of many additional networks such as rings and meshes [10]. Using this embedding, our goal is to create a more robust and efficient message-passing system for processing of large amounts of data in parallel [5].

The remaining sections of the paper are organized as follows: The second section provides the definition and topological properties of EDS network. The third section describes the bipanconnectivity and Hamiltonian property of the EDS network. The fourth section explains the embedding characteristics of an Extended Double star network. It explains the ring and mesh embedding of current EDS network. Finally, Section 5 brings the conclusion of the current paper.

2. Definition and Topological Properties of EDS

The EDS is a hierarchical topology consisting with one NC and a basic module which is a DS graph that enables the processing element to focus only on computational activity [11]. There are two parameters namely n and k forcharacterizing the layred structure of the EDS topology, where n is the dimension of DS and k represents the level of the NC. The movement from the outer ring to the inner ring and vice versa takes place through the leaf edge. The star notation is utilized here to represent the node addressing for the basic module. The star notation is the permutation of n bitstrings along with a binary bit for designating the ring structure.

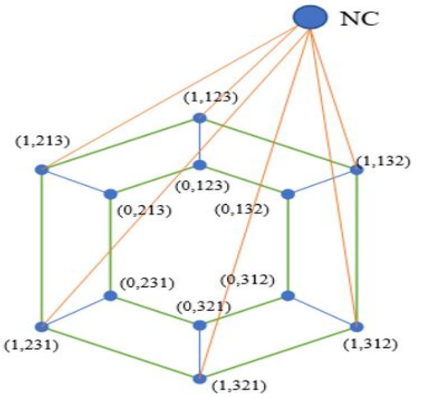

Fig. 1. Extended Double Star of Dimension Three, EDS$_{(3,1)}$

The different topological parameters of the EDS graph are discussed in detail using graph theoretical notations in [13].The total number of computing nodes of EDS is $P = (2n!)^k + \frac{(2n!)^k-1}{2(n!)-1}$. The degree of EDS network is $(n!+n+1)$.The total number of edges in EDS network is E = $2n(n-1) + n!\,\frac{(2n!)^k-1}{2(n!)-1}$.The diameter of the EDS network is $\lfloor \frac{3}{2}(n-1) \rfloor + 2k$. The cost of computation in the EDS(n, k) is given by $\left[\lfloor \frac{3}{2}(n-1) \rfloor + 2k \right] \times (n! + n + 1)$.

3. amiltonian Property of EDS Network

Bipartite Property

In classical graph theory, the vertex V of a bipartite graph can be split into two disjoint and independent subsets $V_1$and V_2. Then there is an edge set E' subset of the edge set Econnecting a vertex in V_1 to one in V_2. The vertex sets $V_1$and $V_2$are resulted from the vertex set V of the graph by removing E'. Equivalently, a bipartite graph is a graph that does not contain any odd-length cycles.

***Theorem1*: EDS Network is bipartite.**

Proof: In EDS apart from NC the PEs connected to both the inner and outer rings can be split into two disjoint and independent sets of vertices V_1 andV_2 respectively. The sets are made distinct by removing the links from outer ring to inner ring.Here $V_1 \cap V_2 = \emptyset$. Also,in EDS network there are no odd-length cycles as each ring contains *n!*number of processing nodes. As V_1 and V_2 node sets belong to the outer and inner rings respectively, hence all removed edges are from V_1 to V_2 satisfying the bipartitecondition.This implies that EDS is bipartite. (***Proved***)Following is a simple algorithm to find out EDS graph is Bipartite

Algorithm for Bipartite EDS (n, k)

In this Bipartite EDS(n, k) algorithm there are two parameter *n* and *k*, where *n* represent the network dimension and *k* represent the level of NC.Theaddressof the computing node is denoted as*V(x, y)*, where the*x*designates level of NC and position of NC connected to the outer ring and*y*typify the cluster address and node position inside the cluster.

Algorithm *BPEDS(V,E, n, k)*

Notations

V: Set of vertices
E: Set of edges
n: Dimension of DS
k: Level of NC
X: Denotes the level of NC and position of NC in outer ring
Y: Cluster address and nodeposition in the cluster
E': Set of edges removed from original topology
V_1: Set of vertices in the outer ring connected to NC
V_2: Set of vertices in the inner ring

```
1.      Start
2.      While mod(V)>0
{
3.      Scan v<x,y>for x address bits
4.      (add NC  to V1)
5.      If the node address bits is starting with 1
then  Put into set V1
6.      Else if node address bits starting with 0
Put into set V2.
7.      Add the edge in E'
8.      Scan the neighbor PE
9.      ||V ||= ||V ||-1
}
10.     If V1 ∩ V2 = Ø and  mod(V1) =k+ mod (V2)
then Return "EDS is Bipartite"
Else Return "EDS is not Bipartite"
11.     Stop
```

Illustration

In the Fig. 2 the removed edges are marked with red ticks depicting that EDS(3,1) is bipartite. The nodes of inner ring (node address bits starting with 0) are not connected to the NC and hence belong to node set V_2. The set of nodes with starting address bit 1 in outer ring and the NC belong to set V_1. As the base module is bipartite, hence the entire recursive structure EDS(n,k) will behave in the same manner.

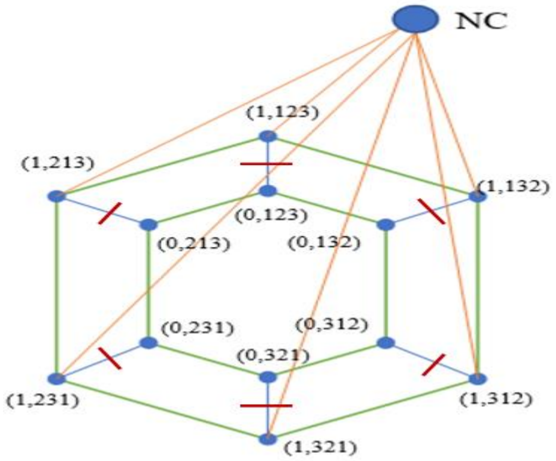

Figure 2: Bipartite Condition of Base Module EDS(3,1)

Hamiltonian Laceablility

A connected bipartite graph is called Hamilton-laceable, if it hasV_1- V_2 Hamiltonian path for all pairs of vertices $v_1$and v_2, where $v_1$belongs to one set of the bipartition, and $v_2$to the other.

***Theorem 2:* EDS is Hamiltonian-laceable.**

Proof: From Th.1, it's is clear that the EDS is a connected bipartite graph. It has even number of nodes. Also, EDS has V_1– V_2Hamiltonian path for all pairs of vertices $v_1$and v_2, where v_1 belongs to one set of the bipartition, and v_2 to the other.In the EDS network with the help of NC ($V_1 \cup V_2$, E') is possible where E' contains those edges that are removed to construct $V_1$and V_2. Hence EDS is Hamiltonian-laceable. (***Proved***)

From Theorem 1 and 2 it can be verywell observed that a Hamiltonian path will exist in the EDS topology. A Hamiltonian path is a path between two vertices of a graph where each vertex is visited exactly once. It is also known as a Hamilton path. In a graph, a Hamiltonian cycle can be viewed as a closed loop where the beginning and endpoints of the path are adjacent.

***Theorem3*: Extended Double Star EDS$_{(n,k)}$ contains a Hamiltonian cycle.**

Proof: According to the improved degree-based condition for Hamiltonicityin a graph is given by Mehedy, Kamrul and Kaykobad in 2007 as, for p number of nodes, the graph must have at least $\frac{p}{4}$ edges.

In EDS(n,k) topology, the count of edges (E) = $2n(n-1)+n!\left(\frac{(2n!)^k-1}{2n!-1}\right)$

In EDS(n,k) topology, the count of nodes (p) = $(2n!)^k+\frac{(2n!)^k-1}{2(n!)-1}$

If the total edges of EDS(n,k) are more than $\frac{p}{4}$ for (p > 1) then the Hamiltonian cycle must be existing.

Now, by using mathematical induction method we have to proof that:

$S_{n,k}$: $2n(n-1)+n!\left(\frac{(2n!)^k-1}{2n!-1}\right)>\frac{(2n!)^k+\frac{(2n!)^k-1}{2(n!)-1}}{4}$ is true. (for n ≥ 3, k = 1)

Base Case (show that $S_{n,k}$ is true):

For, $S_{3,1}$: $2\times 3\,(3-1)+3!\left(\frac{(2\times 3!)^1-1}{2\times 3!-1}\right)>\frac{(2\times 3!)^1+\frac{(2\times 3!)^1-1}{2(3!)-1}}{4}$

$\Rightarrow S_{3,1}:\ (12+1)\ >\frac{13}{4}$

$\Rightarrow S_{3,1}:\ 13\ >\ 4$

So, the base case $S_{3,1}$ is true.

For inductive hypothesis, assume that $S_{t,r}$ is true for t ≥ 3 & r = 1 .

$S_{t,r}$: $2t(t-1)+t!\left(\frac{(2t!)^r-1}{2t!-1}\right)>\frac{(2t!)^r+\frac{(2t!)^r-1}{2(t!)-1}}{4}$ is true. (for t ≥ 3, r = 1)

$\Rightarrow S_{t,r}$: $2t^2$ - $2t$ + $t!$ $>\frac{2t!+1}{4}$ is true. (for, r = 1)

We have to show that, $S_{t,r}$ is true follows that $S_{t+1,r+1}$ is true.

Consider, $S_{t+1,r+1}$ = $2(t+1)(t+1-1)+(t+1)!\left(\frac{\{2(t+1)!\}^{r+1}-1}{2(t+1)!-1}\right)$

= $2t(t+1)+(t+1)!\left(\frac{\{2(t+1)!\}^{2}-1}{2(t+1)!-1}\right)$ (for, r = 1)

= $2t^2+2t+(t+1)!\,\{2(t+1)!\}^2+1$

= $2t^2+2t+4\{(t+1)!\}^3+1$

= $2t^2+2t+4\{(t+1)!\}^3+1+2t-2t+t!-t!$

= $2t^2$ - $2t$ + $t!$ $+2t+4\{(t+1)!\}^3+1+2t-t!$

$>\frac{2t!+1}{4}+4t-t!+4\{(t+1)!\}^3+1$

$>\frac{2t!+1}{4}+\ 4[t+\{(t+1)!\}^3]\ -\ t!+1$

$>\frac{2(t+1)!^2+2(t+1)!+1}{4}$

$>\frac{2(t+1)!\{(t+1)!+1\}+1}{4}$ (R.H.S)

Therefore, $S_{t,r}$ is true follows that $S_{t+1,r+1}$ is true.

So by mathematical induction method, it can be assumed that

$S_{n,k}$: $2n(n-1)+n!\left(\frac{(2n!)^k-1}{2n!-1}\right)>\frac{(2n!)^k+\frac{(2n!)^k-1}{2(n!)-1}}{4}$ is true. (for n ≥ 3 , k = 1)

In this way we can prove that,

$S_{n,k}$: $2n(n-1)+n!\left(\frac{(2n!)^k-1}{2n!-1}\right)>\frac{(2n!)^k+\frac{(2n!)^k-1}{2(n!)-1}}{4}$ is true. (for n ≥ 3 , k = 2)

$S_{n,k}$: $2n(n-1)+n!\left(\frac{(2n!)^k-1}{2n!-1}\right)>\frac{(2n!)^k+\frac{(2n!)^k-1}{2(n!)-1}}{4}$ is true. (for n ≥ 3 , k = 3)

.

. . .. & so on

Therefore, $S_{n,k}$: $2n(n-1)+n!\left(\frac{(2n!)^k-1}{2n!-1}\right)>\frac{(2n!)^k+\frac{(2n!)^k-1}{2(n!)-1}}{4}$ is true. (for n ≥ 3 , k ≥ 2)

This implied that, Extended Double star EDS(n,k) contains a Hamiltonian cycle. ***(Proved)***

4. Embedding

Graph embedding has been increasingly essential in a range of computer architecture and machine learning techniques in recent years. We perform multiple tasks, including clustering, principal component analysis (PCA), classification, etc., in a very robust manner using the nodes, edges and other components of the graph embedding. Embedding is a most commonly used machine learning technique that involves with the representation of a complex object such as texts, images, as well as graphs into a vector with a reduced number of features as compared to the dimension of the dataset of billions number of nodes in a graph, while still sustaining the most important information about them.

Let, G is a guest and H is a host finite graphs of n vertices, where $V(G)$and $V(H)$ denote the vertex set of $G\ andH$ and $E(G)$ and $E(H)$ denote the edge set of $G\ andH$ respectively. Then an embedding function$f$ of G into H is defined as:

1. f is a one-to-one mapping from $V(G)\rightarrow V(H)$.
2. P_f is a one-to-one mapping from $E(G)\ to\ P_f\big(f(u),f(v)\big) : P_f(f(u),f(v))$ is a path in H between $f(u)$ and $f(v)$ for $\{(u,v)\in E(G)\}$.

From the definition it is clear that the graph embedding assigns a fixed-length vector representation to each entity (typically nodes) in the graph. These embedding preserve the graph's topology, which are lower dimensional representation of the graph. Graph embedding allows for the efficient simulation of one network architecture through another. Embedding is very essential part of computer science because of we can easily modified an algorithm for graph H, which is made for graph G. Dilation, congestion, and expansion are among the parameters related to graph embedding. The dilation of an embedding is defined as the maximum length of such paths that can be taken over all source edges. Link congestion provides the most of the inter-process communication source channels that can be mapped to a single physical host link. To reduce contention on links or router buffers, link congestion should be minimized. Node congestion defines the maximum number of inter-process channels that can pass through a single host router. Expansion is the measure of processor utilization. It is defined as the ratio of the vertex size of host to guest graph.

4.1 Ring Embedding in EDS

The EDS is an interconnection network that consists oftwo-star networks linked to a single network controller (NC). Previously, studies are made how different topologies can be embedded as binary trees, meshes, rings of stars, in CQs, and SCQs. Through this method, we are able to acquire ring properties in EDS analogous to a star and double star network. A ring is usually used to describe a path where the starting and ending nodes are the same and same node cannot be travelled more than once. The Hamiltonian cycle has already been derived from an EDS network. In below, after passing through each node of the EDS network at least once, the route is finished when it reaches the node from which it initially set off.

Lemma 1:The embedding of ring is possible in $EDS_{(3,1)}$ and the upper bound on the size of the ring is 13.

Proof: By the use of Gray code, we must encode numbers so that the only difference between them is a single digit. Frequently, the term Gray code refers to a "reflected" code, or more precisely, the binary reflected Gray code. Developing an n-bit Gray code in EDS:

$EDS_{(3,1)}$ network consist of two star graphs together with single Network Controller. A ring with 12 nodes (R1 through R12) and one NC is shown in Fig.3. The addresses of the nodes in the EDS graph are given below.

R1= (1,123) R2= (1,132) R3= (1,312)

R4= (1,321) R5= (1,231) R6= (1,213)

R7= (0,213) R8= (0,231) R9= (0,321)

R10= (0,312) R11= (0,132) R12= (0,123)

R13= NC (Network Controller)

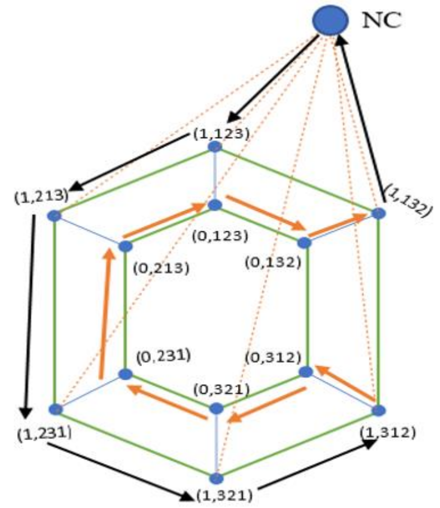

Fig. 3. Embedding of Ring Topology in $EDS_{(3,1)}$

In $EDS_{(3,1)}$ network 12 edges and 1 NC are to be considered. Therefore, 12+1= 13 number of edges needed to embed a ring in $EDS_{(3,1)}$ network shown in the Fig.3. ***(Proved)***

Lemma 2: The upperbound of the ring sizeembedded in $EDS_{(3,2)}$ network is 169.

Proof: In order to embed a ring in $EDS_{(3,2)}$, all nodes will be covered just once and the beginning and ending points being the same. Right after travelling through a cluster, depending upon the y part of the node address, next star graph neighbor will be chosen as the next subsequent related clusteras shown in Fig.4 below. The $EDS_{(3,2)}$ network consists of twelve numbers of clusters connected in star manner to the main basic block DS network in inner and outer ring as shown in Fig.4 and a single Network Controller at level 0. At level one, there are 2n! numbers of clusters and each cluster has 13 nodes (12 computing nodes and one NC). To embed a ring in each network, 13 edges are needed. Therefore, total 13 × 12 = 156 number of edges are needed to embed a ring. At level zero, the $EDS_{(3,2)}$ network consists of a DS network with a single NC, which has 13 nodes and it required total 13 edges to embed a ring. So, maximum 156 + 13 = 169 edges are needed to embed a ring in the$EDS_{(3,2)}$ network. ***(Proved)***

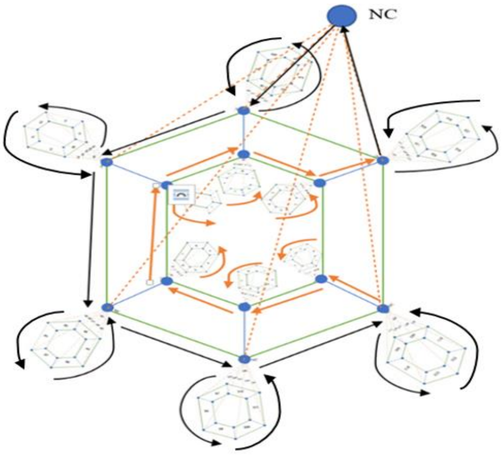

Fig. 4. Embedding of Ring topology in EDS$_{(3,2)}$

Lemma 3: The upper bound on the size of the ring possible in EDS$_{(4,1)}$ network topology is 49.

Proof:EDS$_{(4,1)}$ network consists of four number of double star networks together with single Network Controller. Each double star network needed 11 number of edges to embed a ring in it as shown in the Fig.5. So, there are total 11 × 4 = 44 number of edges are required to embed a ring in four different DS networks. Furthermore, each DS networks are connected with each other in star manner. Therefore, again five number of edges are needed to embed a complete ring shown in the Fig.5. So, maximum 44 + 5 = 49 edges are needed to embed a ring in a EDS$_{(4,1)}$ network. ***(Proved)***

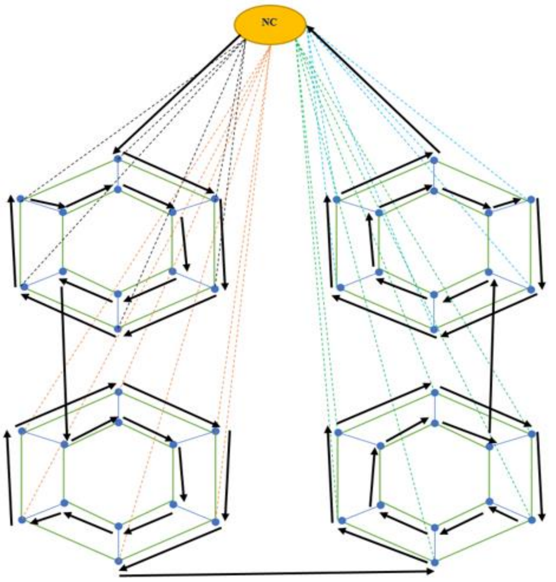

Fig. 5. Embedding of Ring topology in EDS$_{(4,1)}$

Lemma 4: The maximum size of the ring in EDS$_{(4,2)}$ network is 673 .

Proof:EDS$_{(4,2)}$ network consists of four number of double star networks with a single Network Controller. Each double star networks consists of twelve numbers of clusters connecting in star manner shown in the Fig.6. At level one, each clusters have 13 nodes, which required total 13 number of edges to embed a ring and there are total four number of independent DS networks shown in Fig.6. So, there are total (13 × 12) × 4 = 624 number of edges are needed to embed ring. At level zero, each double star network required 11 number of edges to embed a ring in it as shown in the Fig.6. So, there are total 11 × 4 = 44 number of edges are required. Furthermore, each DS networks are connected with each other in star manner. So, again five number of edges are required to embed a complete ring shown in the Fig.6. Therefore, at level zero the network required total 44 + 5 = 49 edges to embed a ring. So, maximum 624 + 49 = 673 edges are needed to embed a ring in EDS$_{(4,2)}$ network. ***(Proved)***

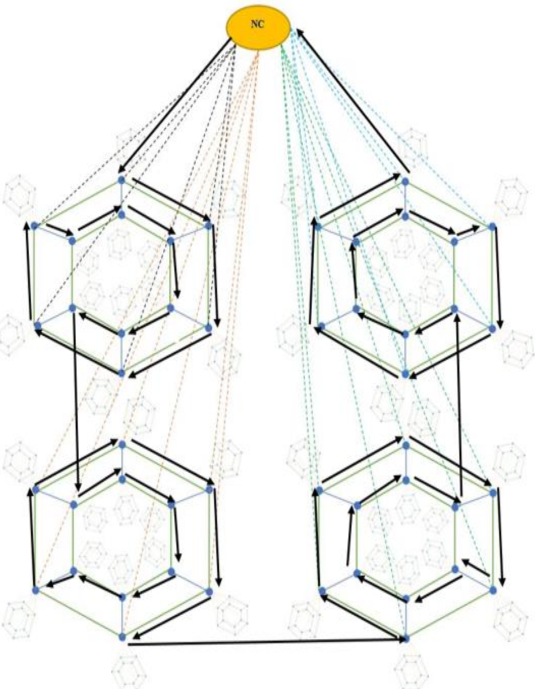

Fig. 6. Embedding of Ring topology in EDS$_{(4,2)}$

***Theorem-1:* The upper bound on the ring size embedded in EDS$_{(n,k)}$ is (2n! + 1) + 52 (k-1) (n) (n-1) (n-2)...3.**

Proof:The Extended Double Star structurecontains $(2n!)^k + \frac{(2n!)^k-1}{(2n)!-1}$ number of computing nodes. To calculate the total sum of connecting edges in EDS$_{(n,k)}$ to embed a ring, a comparison is being made from the above stated four ring embeddings. From *Lemma-1*: In EDS$_{(3,1)}$ network, the maximum size of the ring is 13. At level zero, the EDS$_{(3,1)}$ network has 13 edges, which can be written as (2n! + 1) edges. Therefore, we can say that in general the EDS$_{(3,1)}$ network has (2n! + 1) + (k-1) edges. Where, (k-1) is the highest level of NC in EDS$_{(n,k)}$network. From *Lemma-2*: In EDS$_{(3,2)}$ network, the maximum size of the ring is 169. At level zero, the EDS$_{(3,2)}$ network has 13 edges, which can be written as (2n! + 1) edges. At level one, the EDS$_{(3,2)}$ network has (13 × 12) edges, which can be written as 52 × n edges. Therefore, we can say that in general the network has (2n! + 1) + (k-1) × 52 × n edges. Where, (k-1) is the highest level of NC in EDS$_{(n,k)}$ network. From *Lemma-3*: In EDS$_{(4,1)}$ network, the maximum size of the ring is 49. At level zero, the EDS$_{(4,1)}$ network has 49 edges, which can be written as (2n! + 1) edges. Therefore, we can say in general the network has (2n! + 1) + (k-1) edges. Where, (k-1) is the highest level of NC in EDS$_{(n,k)}$ network. From *Lemma-4*: In EDS$_{(4,2)}$ network, the maximum size of the ring is 673. At level zero, the EDS$_{(4,2)}$ network has 49 edges, which can be written as (2n! + 1) edges. At level one, the EDS$_{(4,2)}$ network has [(13 × 12) × 4] edges, which can be written as 52 × n × (n-1) edges.

Therefore, we can say in general the network has (2n! + 1) + (k-1) × 52 × n × (n-1) edges, where, (k-1) is the highest level of NC in the EDS$_{(n,k)}$ network. After considering all of the above four cases, to get a generalized equation for EDS$_{(n,k)}$, the upper bound on the size of the ring contained in EDS$_{(n,k)}$ structure will be (2n! + 1) + 52 (k-1) (n) (n-1) (n-2)….3. ***(Proved)***

From the Theorem 1 it is clear that the dilation and expansion of ring embedding in EDS topology is one. Also, the ring embedding has unit link and node congestion.

4.2 Mesh Embedding in EDS

Mesh network is nothing but more than connect one processor to four other processors. In mesh network, there is a connection between the processors in the last column and the first processor of the next row. There should be a connection between the processors in the bottom right and top left corners.

Lemma 5: The upper bound on the number of 2D disjoint mesh embedding in EDS$_{(3,1)}$is 6.

Proof:The EDS$_{(3,1)}$topology consists of two sets of star graphs organized into two rings with single Network Controller connected to the outer rings. The links connectingthe inner ring with the outer ring encompass six numbers of two dimensional meshes as shown in Fig.7 (a). Here the meshes shown in figure are numbered from 1 to 6 as shown in Fig. 7 (b). Hence, maximum 6 numbers of 2D meshes are contained in EDS$_{(3,1)}$. ***(Proved)***

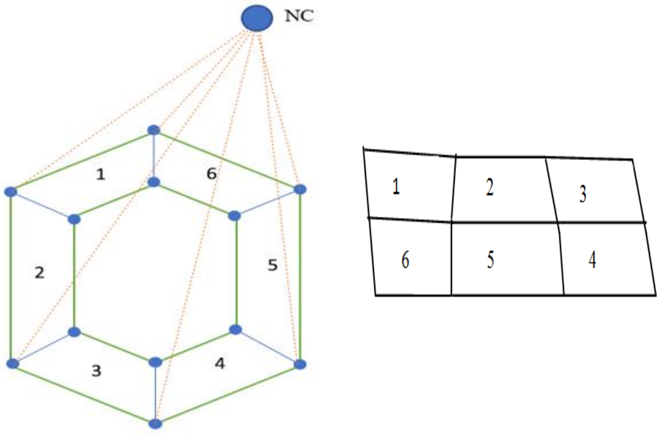

Fig. 7.(a) Mesh Embedding in EDS$_{(3,1)}$, (b) Six number of 2D Meshes in EDS$_{(3,1)}$

Lemma 6: The highest number of meshes contained in theEDS$_{(3,2)}$topology is determined to be 78.

Proof:TheEDS$_{(3,2)}$ network consists of a double star network with a single Network Controller and the double star network has twelve numbers of clusters connecting in star manner shown in Fig.8. At level one, the network has twelve number of clusters. Each clusters creates 6 number of meshes. Therefore, total (12 × 6) = 72 number of meshes are created by the network. At level zero, the EDS$_{(3,2)}$ network consists of single DS network with a NC, which embed with six number of meshes. So, maximum 72 + 6 = 78 meshes are possible in EDS$_{(3,2)}$.
(proved)

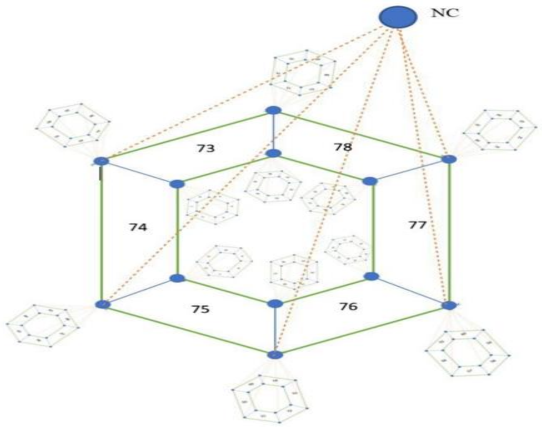

Fig. 8. Embedding of Mesh in EDS$_{(3,2)}$

Lemma 7: The highest number of meshes contained in the EDS$_{(4,1)}$ topologyis determined to be 36.

Proof:$EDS_{(4,1)}$ network consists of four double star networks linked to a single Network Controller and each double star networks embedded with six number of meshes. So, total 6 × 4 = 24 number of meshes are created by the DS networks. Again all of the four double star networks creates two sets of meshes with each other shown in the Fig.9. Therefore, the EDS network creates twelve number of meshes. So, maximum 24 + 12 = 36 meshes number of meshes are possible in $EDS_{(4,1)}$. ***(Proved)***

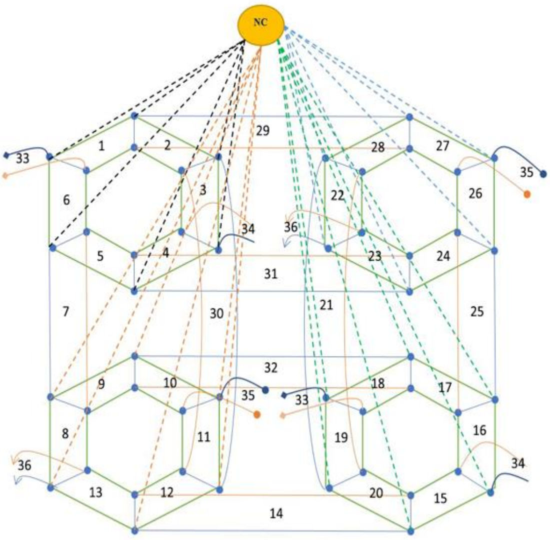

Fig. 9. Embedding of Mesh in $EDS_{(4,1)}$

Lemma 8: The upper bound on the meshembedding into the $EDS_{(4,2)}$ topology is determined to be 324.

Proof:$EDS_{(4,2)}$ network consists of four number of double star networks with a single Network Controller and each double star networks have twelve numbers of clusters connecting in star manner shown in the Fig.10. At level one, each clusters creates six meshes and there are total (12 × 4) number of clusters. So, the maximum 12 × 4 × 6 = 288 number of meshes possible at level one. At level zero, $EDS_{(4,2)}$ network consists of four number of double star networks which are connected to a single NC. Each double star network embedded with six number of meshes. So, total 6 × 4 = 24 number of meshes are created by the DS networks. Then, all of the four double star networks creates two sets of meshes with each other shown in the Fig.11 and again are created 12 number of meshes. Therefore, at level zero maximum 24 + 12 = 36 meshes number of meshes are possible. So, maximum 288 + 36 = 324 number of meshes possible in $EDS_{(4,2)}$ network . ***(Proved)***

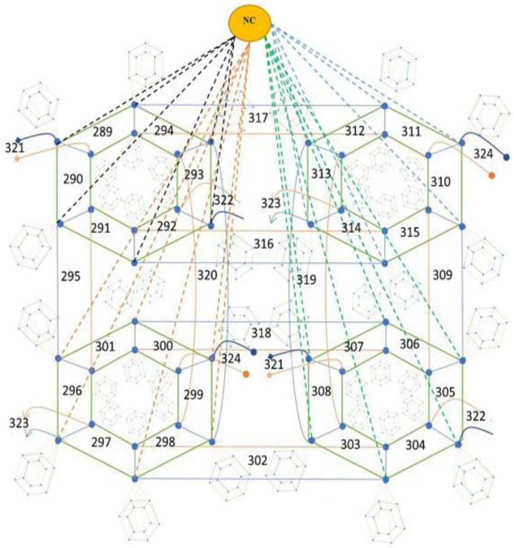

Fig. 10. Mesh Embedding in $EDS_{(4,2)}$

Theorem-2: The total number of meshes that can be embedded in $EDS_{(n,k)}$ is $\frac{n!(n-1)}{2}$ + 24 (k-1) (n) (n- 1) (n- 2)….3 .

Proof: The $EDS_{(n,k)}$ is a n dimensional network of k levels. The EDS network has $EDS_{(3,1)}$ as the basic module and the network has six number of meshes. To calculate the total number of meshes in $EDS_{(n,k)}$, we have to compare the above four mesh embedding of extended double star network. From lemma-5: In $EDS_{(3,1)}$ network, the maximum size of the meshes are six. At level zero, the $EDS_{(3,1)}$ network has six meshes, which can be written as $\frac{n!(n-1)}{2}$ meshes. Therefore, we can say that in general the network has $\frac{n!(n-1)}{2}$+ (k-1) meshes. Where, (k-1) is the highest level of NC in $EDS_{(n,k)}$ network. From lemma-6: In $EDS_{(3,2)}$ network, the maximum size of the meshes are 78. At level zero, the $EDS_{(3,2)}$ network has six meshes, which can be written as $\frac{n!(n-1)}{2}$ meshes. At level one, the $EDS_{(3,2)}$ network has (12 × 6) meshes, which can be written as 24 × n meshes. Therefore, we can say that in general the network has $\frac{n!(n-1)}{2}$ + (k-1) × 24 × n meshes. From lemma-7: In $EDS_{(4,1)}$ network, the maximum size of the meshes are 36. At level zero, the $EDS_{(4,1)}$ network has 36 meshes, which can be written as $\frac{n!(n-1)}{2}$ meshes. Therefore, we can say that in the network has $\frac{n!(n-1)}{2}$ + (k-1) meshes. From lemma-8: In $EDS_{(4,2)}$ network, the maximum size of the meshes is 324. At level zero, the $EDS_{(4,2)}$ network has 36 meshes, which can be written as $\frac{n!(n-1)}{2}$ meshes. At level one, the $EDS_{(4,2)}$ network has [(12 × 6) × 4] meshes, which can be written as 24 × n × (n-1) meshes.

Therefore, we can say that in general the network has $\frac{n!(n-1)}{2}$ + (k-1) × 24 × n × (n-1) meshes. After considering all of the above four cases, to get a generalized equation for EDS$_{(n,k)}$, The total number of meshes in EDS$_{(n,k)}$ network will be, $\frac{n!(n-1)}{2}$ +24(k-1)(n)(n-1)(n-2)…3 . ***(Proved)***

By inspecting the Lemmas and Theorem 2 it is clear that the dilation, congestion and expansion of mesh embedding in the EDS topology is unity.

5. Conclusion

In this current study, the Extended Double Star network topology is anunique interconnection system that can be suitable for implementing large-scale parallel computing. As compared to previous networks, this particular EDS network has significantly better qualities such as node degree, diameter, cost, traffic density, and robustness. The architecture is bipartite and Hamiltonian laceable. Embedding of both the Ring and Mesh topologies can effortlesslydone into the Extended Double star network. The longest ring that can be realized in the EDS is estimated. Thus, it is concluded that the EDS contains cycles greater than 6 as subgraphs. This research also finds an upper bound on the number of disjoint meshes that can be embedded into EDS topology. The extended double star network conserves all of the topological features of the original star network. The inclusion of the controller nodes result in faster and economical message passing feature and it will be beneficial for the parallel computing systems. The inner rings computing nodes can also be treated as back up nodes can make the topology more fault tolerant. In addition to that the inner and outer ring design can be a candidate for implementing distributed file structure with map and reduce framework of parallel systems which strongly focuses on scope for some future research. The graph theoretical results of EDS topology obtained here claim that this hierarchical architecture can be very influential from communication and computation point of view in the Big data scenario.

References

1. Y. Saad, M.H. Schultz, “Topological properties of hypercubes”, IEEE Transactions on Computers, Vol. 37, pp.867–872, 1988.

2. Nibedita Adhikari, C.R. Tripathy, Binod Nag “On the Extension of a Star Based Parallel Interconnection Network Topology”, Proc. of The Eighth Intl. Conf. On Advances in Computing, Control and Networking - ACCN 2018, IRED, USA, Jun2018, Paris, France, pp1-5, 2018.

3. C.R. Tripathy, “Star-cube: a new fault tolerant interconnection topology for massively parallel systems”, Journal of The Institution of Engineers (India), ETE Division, Vol. 84, pp. 83–92, 2004.

4. N. Adhikari, C.R. Tripathy, “Extended crossed cube: an improved fault tolerant interconnection network”, IEEE International Conference on Networked Computing, pp. 86–91, 2009.

5. Abuelrub, Emadeddin Mohamed, "Interconnection Networks Embeddings and Efficient Parallel Computations." (1993). LSU Historical Dissertations and Theses. 5554.

6. N. Adhikari and C R Tripathy, “Star crossed cube: an alternative to star graph”, Turkish Journal of Electrical Engineering and Computer Sciences, Vol.22, pp. 719-734, 2014.

7. Rahman, M. S., M. Kaykobad, J. S. Firoz. “New Sufficient Conditions for Hamiltonian Paths”. – The Scientific World Journal, Vol. 2014, 2014, pp. 1-7.

8. N. Adhikari and C R Tripathy, “n-star: A New Two Level Interconnection Network”, In: Proc. of 8th International Conference on Distributed Computing and Internet Technology (ICDCIT’12), pp. 50-61,2012.

9. N.K. Swain, C.R. Padhan, N. Adhikari. (2022). “On Embedding Properties of Double-Star Interconnection Network Topology”. In: Udgata, S.K., Sethi, S., Gao, XZ. (eds) Intelligent Systems. Lecture Notes in Networks and Systems, vol 431. Springer, Singapore.

10. S. Ranka, J. Wang, N. Yeh, “Embedding meshes on the star graph”, Proceedings of the IEEE Conference on Supercomputing, pp. 476–485, 1990.

11. H. Barik, N. Swain, L. Rout, and N. Adhikari, “Double Star: A high performance network for big data”, SSRN eLibrary CTFC 2019, (January 10, 2020).

12. Mehedy, L., H. M. Kamrul, M. Kaykobad. “An Improved Degree Based Condition for Hamiltonian Cycles”. – Information Processing Letters, Vol. 102, 2007, pp. 108-112.

13. Sadashiba Pati, Nibedita Adhikari and Vinay Singh, “Extended Double Star: A Massive Parallel Big Data Network”,2023 ECB, Special Issue 2023, Vol 12, No.3, pp. 3898-3912.

14. N. Adhikari and B. Nag, “On Topological Properties of A Star based Large Scale Parallel System”, Proceedings of ETNCC2011, International Conference on Emerging Trends in Networks and Computer Communications, IEI Udaipur Section April 22-24, (2011).

15. K. Day and A. Tripathy, “A Comparative Study of Topological properties of Hypercubes and Star Graphs”, IEEE Trans. Parallel & Distributed Systems, Vol.5.

16. S.B. Akers, D. Harel, B. Krishnamurthy, “The star graph: an attractive alternative to the n-cube”, International Conference on Parallel Processing, pp. 1249–1268, 1987.

17. N. Adhikari, C.R. Tripathy, “On a new interconnection network for large scale parallel systems”, International Journal of Computer Applications, Vol. 23, pp.39–46, 2011.