International Journal of Applied Engineering & Technology

HYPER DOUBLE STAR INTERCONNECTION NETWORK FOR HIGH PERFORMANCE COMPUTING SYSTEMS

[1]Sadashiba Pati, [2] Vinay Singh, [3]Nibedita Adhikari and [4]Gyana Ranjan Nayak
[1,2]Usha Martin University, Ranchi, India
[3]Utkal University, Vani Vihar, Bhubaneswar, Odisha, India
[4]CUPGS, BPUT, Rourkela, Odisha, India

ABSTRACT.
With increasingly rapid economic globalization the need for high speed computing in all domains have become a critical problem. To alleviate data traffic in message passing, a large amount of money is spent on designing computing infrastructure and their maintenance. Everywhere there is a need for faster data access in the big data computing scenario. Hierarchical computing system is always a good solution in massive parallel computing. Data traffic management can provide better decision making in big data era. It is attracting great attention among people from all walks of life. Generally, the parallel algorithms / programs are termed as a set of message passing processes. It also needs high level of computation. Hence parallel interconnection networks play a vital role in high speed message passing and related computations. The Hypercube and Star graphs are two popular networks and many researchers have derived new structures from these two. The current paper attempts to propose a new star based hybrid topology and derive its topological properties to establish its strength.

Keywords: Cartesian product networks, Interconnection networks, Star graph, Hyper star graph

1 INTRODUCTION

The parallel Interconnection Network (PIN) design is currently getting a lot of attention for numerous major data applications. The Parallel processing architectures along with algorithms are examined over years for enhancing computer's performance. It's become a crucial topic because it is able to build an essential high end computing and message passing system. The recent electronic devices work with multiple sensors and always stay connected to the Internet. As a result huge amount of data is captured and need to be stored. This large-scale activity and voluminous data has resulted in Big data systems. The high volume and wide variety of data need to be processed accurately and at a faster rate. For these reasons the focal point of many researchers is on the parallel, multi-core computing. There has been a lot of research over many years into parallel processing architectures and methods to enhance computer's performance. Both computation and message transfer between processes occur simultaneously in parallel computing. Now, every scenario including Big data, uses massive parallel computing in the background. A large size problem can be broken down into many smaller problems with the same characteristics, each of which can be handled concurrently by some different processors in the system. As a result, the processing power of a system can be increased through the use of parallel computing. So, the most common way to do this kind of processing is with Multiple Instructions, Multiple Data (MIMD) computers. To use MIMD, you need interconnection networks. A machine can transmit data from its source node to its destination node in a parallel interconnection network, with the objective of achieving the lowest feasible latency in the process. PIN machines are less costly than other machines.

There are many different kinds of PIN, but the most common are star graphs and hypercube[1, 4]. The basic networks are updated throughout time to improve their topological properties. The interconnection network topology is a common foundation for parallel computing. The interconnection network is now recognized as the most accurate model for parallel computing. It has become an essential subject as it can establish a critical high performance computing and message passing system. In general, these systems consist of various processing elements, memory units and other resources. These components are interconnected through an interconnection network (IN). An interconnection network can be the backbone of Big data systems, which models the connections among the processing devices for the purpose of inter-device communication[3]. The suitability of an interconnection network topology is evaluated in terms of its degree, diameter, cost, fault tolerance, fault diameter, reliability, load balancing and other such performance attributes. In addition, cost effectiveness and time cost effectiveness are also considered while analyzing the performance of a network topology. Since the IN's provide various mechanisms for data transfer between processors and memory modules, their topological features vary to a great extent. This variation leads to improvement in Parallel Interconnection System. The variation in the parameters are often used by researchers to suggest a new network and its suitability for implementation.

For many scientific and engineering applications, the parallel processing has emerged as a powerful system to fulfil the ever growing demand for more computing power. The field of parallel processing has undergone significant changes over the past decade. In general, these systems consist of various processing elements, memory units and other resources. These components are interconnected through an interconnection network (IN). Thus, the interconnection network occupies a prominent place in the study of parallel system as if determines the performance of the entire system[2]. Since the IN's provide various mechanisms for data transfer between processing nodes or between processors and memory modules, their topological features vary great extent. This variation leads to classification of the Parallel Interconnection System. The degree, no of nodes, edges, diameter, connectivity, bisection width, average node distance and cost are a few parameters which are evaluated to compare the various interconnection systems.

However, there are major practical difficulties with the star graph related to its poor scalability and unsuitability for real applications. While for the hypercube for example, there is a vast body of parallel algorithm developed to solve numerous computationally intensive problem.

The current work tries to study a specific interconnection network architecture that are called as Hyper Double Star (HDS) for implementing on large multi-processes system. The details like construction and topological parameters need to be discussed and compared with the existing networks. The rest of the paper is organized as follows: The section 2 describe the background of current research. The section 3 and 4 presents briefly the proposed network and its parameters. In section 5 we discuss theoptimal routing for the proposed topology. Then the results of comparison of topological properties of HDS network against the parent networks are presented in Section 6. Finally, Section 7 concludes the paper.

2 BACKGROUND

2.1 Hyper Cube

A Hyper Cube is a parallel interconnection network where each node is having equal node degree. The node degree and dimension of hypercube are same. Also message passing to every node is easy with message density 1. It consists of 2^n nodes and the maximum distance is also n[7]. If each node in an n-dimensional hypercube is given a single n-bit binary address, then a connection will only be established between two nodes in the hypercube if and only if the binary addresses of those two nodes differ by a single bit. The hypercube is a popular interconnection network for parallel computer systems due to its attractive topological characteristics and ability to replicate other networks. However, as more nodes are added to the hypercube, the average number of edges per node will grow at a logarithmic rate.

2.2 Star Graph

The Star graph appeared as a very appealing alternatives to the n-cubes in terms of practically all the desirable characteristics of an interconnection structure. Like the hypercube, the star network is strongly hierarchical and symmetrical in terms of nodes and edges. The star network of dimension n, also known as the n-star graph and indicated by the symbol S_n, is regular in the sense that it's degree is n-1, number of node is n!, and it's diameter is $\lfloor \frac{3}{2}(n-1) \rfloor$. A star graph has a lower degree, node and a smaller diameter and a shorter average distance as compared to a hypercube. The restriction on the number of nodes, which is n! for an n-star network, is one of the most significant practical difficulties associated with the n-star graph[4, 7]. In comparison to n-cubes, it has been demonstrated that star graphs are optimally fault-tolerant and can support more processors with less interconnection hardware and shorter communication delays. In addition, it has been demonstrated that star graphs can handle more processors. In other words, star graphs have the majority of the desired characteristics of n-cubes at a far lower cost.

2.3 Double Star

The Double star is a new network topology of n dimensional star graph of degree '$n-1$'. In this topology each node is connected to a leaf node. The degree of leaf node is 1. The leaf nodes are then connected in star fashion as forming an outer ring. Thus each node is now having degree n. It becomes regular and is recursive in structure. The node address in DS has two parts $v_1(x, y)$ where x is a binary bit representing inner ring or outer ring and y is the n bit permutation denoting star address. In DS graph the number of nodes is $2n!$ and edges of the graph is $E = n^2 - n + n!$. The degree of n-dimensional DS(n) is n[6]. Hence it is regular and recursive in nature. The node address in DS has two pairs V(x, y), where x is a binary bit representing inner ring or outer ring and y is the n bit permutation denoting the star address part. The star part of the address helps to establish the connection between the consecutive basic modules when constructing higher dimension graphs.

2.4 Hyper Star

A hyper star is a multiprocessor interconnection topology. It is an undirected graph based on the Cartesian product of star graphs. It is a member of the Cayley class of symmetric graphs [8]. In hyper star, the base structure consists of m! star clusters connected in star manner. In Star clusters each node is connected to their corresponding nodes of other different star clusters with six edges from outer rings. In HS graph the number of nodes is $2n! \times m!$ and edges of this graph is $E = (n + m - 2) \times n! \times m!/2$. The degree of HDS graph is$(n + m - 1)$.

3 PROPOSED TOPOLOGY

After briefly studying the above mentioned topologies, the current work puts forward a new star based parallel interconnection network topology known as Hyper Double Star (HDS). It aims to pack a huge number of nodes at low diameter resulting in faster communication. The HDS is a derivative of the double-star and hence the star topology.

Construction

The proposed hyper double star is a directionless graph in which each vertex is a two tuple. In hyper double star, the base structure consists of n! number of double star clusters connected in star manner with each other. In DS clusters each node is connected to their corresponding nodes of other different double star clusters with six edges from inner as well as outer rings. The node address in all the clusters of DS have two parts$v_1(x, y)$ where x is a single binary bit representing inner ring or outer ring of the DS cluster. Next the second part y is the 2n number of alphanumeric bit permutation denoting hyper star address that is the position of the node in cluster (first n bits) and then the position of the cluster (second n bits). For example, the y part can be "123abc" in a three dimensional structure. In a single cluster the second n bit combination remains same for all vertices. The first n bit combination only change to denote distinct vertex addresses. In Hyper double star network, two vertices of inner or outer rings are connected if all their corresponding bit permutations are equal except for one pair of corresponding permutations in which exactly one pair of permutations corresponds to a star graph link. The Fig. 1 shows single cluster with node addresses, then in Fig.3 depicts a Hyper double star graph. The detail structure of HDS network is shown in Fig. 2 and Fig. 3. In Fig. 3 the two nodes of outer ring (1, 213abc) and (1, 213cba) are connected to each other because they have the same first permutation 123, while their second permutations abc and cba differ only in their first and last positions basing on the adjacency rule of star graph. So, basically the hyper double star is constructed by forming the Cartesian product of star graphs in its inner and outer rings.

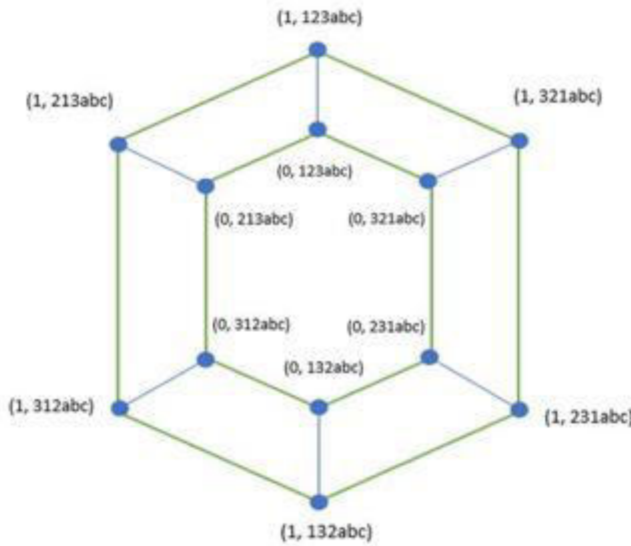

Fig. 1: Node Addresses of Single Cluster

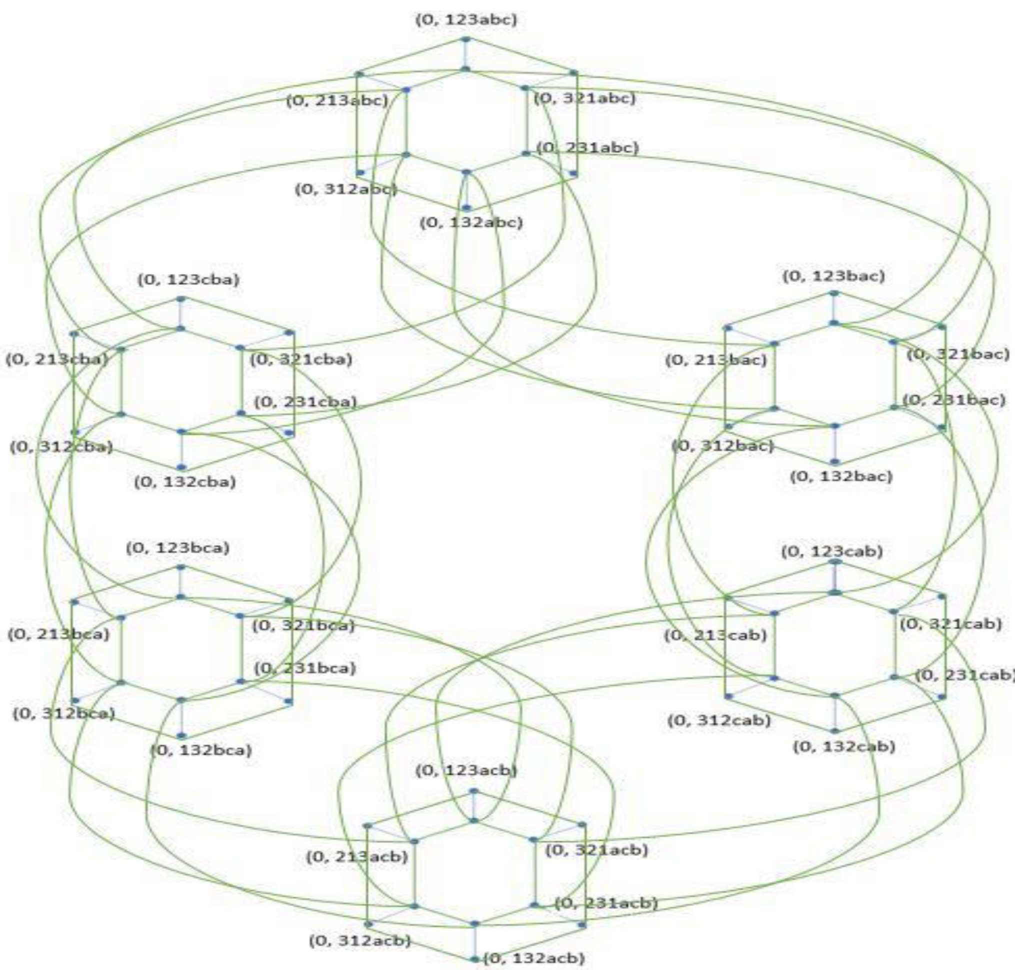

Fig. 2: Inner Ring Connection of HDS(3,3) Network

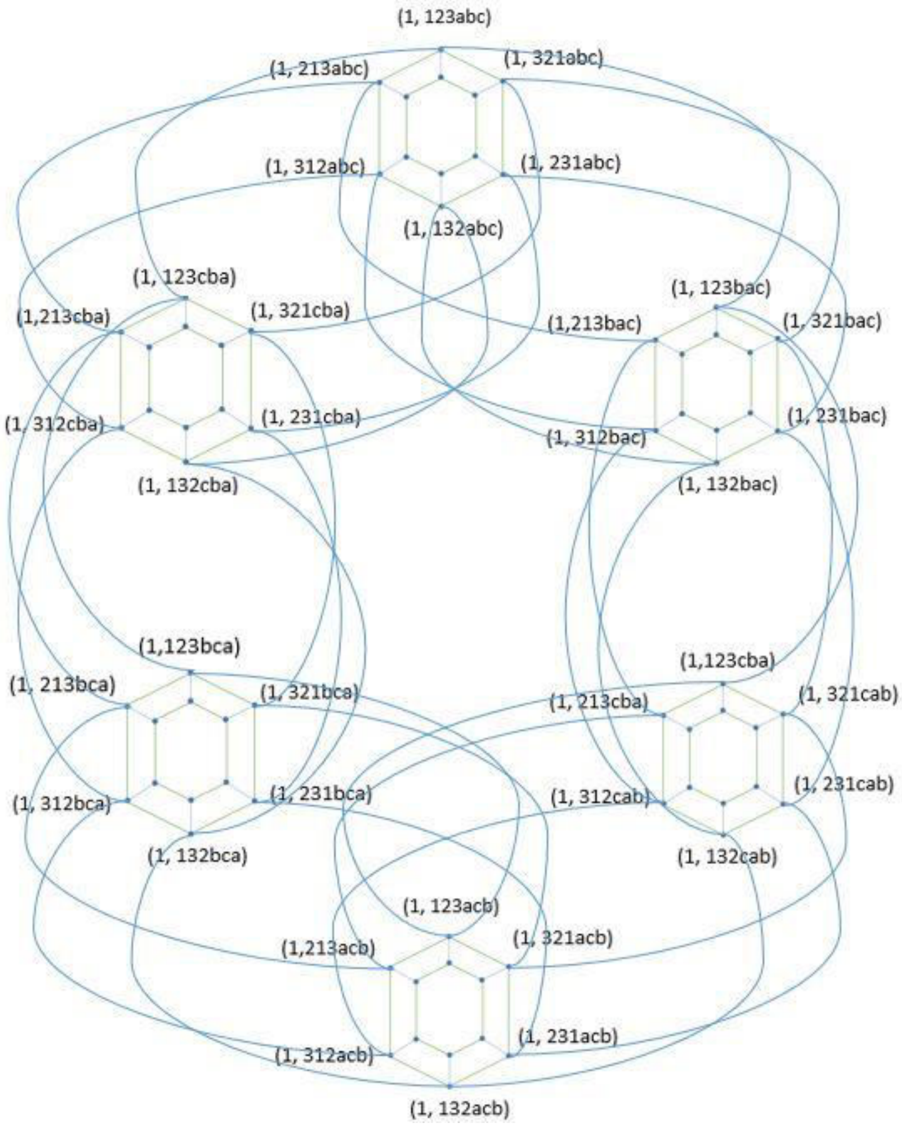

Fig. 3: Outer Ring Connection of HDS(3,3) Network

4 Topological Properties of HDS

This section is devoted to derive some of the topological properties of our proposed HDS network. To describe the topology, the graph theoretical notations are used.

Theorem 1: The node degree of $HDS_{(n,m)}$ graph is D = (n + m – 1).

Proof: The node degree of double star graph is n. Next, each nodes of DS clusters is connected to their corresponding nodes of other (m-1) different DS clusters due to star adjacency rule. So each node is having equal degree and thus the node degree is given as

$$D = n + m - 1 \quad (1)$$

r

Theorem 2: The total number of nodes in $HDS_{(n,m)}$ graph is V = 2n! × m!.

Proof: In HDS network the basic module is a DS_3 graph and the total no of node is $2n!$. In HDS graph it uses six DS clusters. Hence the total no of nodes in $HDS_{(n,m)}$ is given as

V = $2n!$ × m!. (2)

Theorem 3: The total number of edges in $HDS_{(n,m)}$ graph is E = 2m! $(n^2 - n + n!)$.

Proof: The total number of edges in the base graph is $(n^2 - n + n!)$. In hyper double star, the base structure consists of n! number of double star clusters connected in star manner. In DS clusters each node is connected to their corresponding nodes of other (m-1) different double star clusters with six edges from inner and outer rings. So the total number of edges in $HDS_{(n,m)}$ graph is given as

E = 2m! $(n^2 - n + n!)$ (3)

Theorem 4: The diameter of $HDS_{(n,m)}$ graph is $\lfloor\frac{3}{2}(n-1)\rfloor + \lfloor\frac{3}{2}(m-1)\rfloor + 1$.

Proof: In hyper double star, the base structure consists of n! number of double star clusters connected in star manner with each other. The distance between any two farthest nodes in DS cluster is Dg = $\lfloor\frac{3}{2}(n-1)\rfloor + 1$. Diameter of hyper star is $\lfloor\frac{3}{2}(n-1)\rfloor + \lfloor\frac{3}{2}(m-1)\rfloor$. Thus the distance between any two farthest nodes of HDS graph is given as

Dg = $\lfloor\frac{3}{2}(n-1)\rfloor + \lfloor\frac{3}{2}(m-1)\rfloor + 1$ (4)

Theorem 5: The cost of $\mathrm{HDS}_{(n,m)}$ graph is C = $(n + m - 1)\left(\lfloor\frac{3}{2}(n-1)\rfloor + \lfloor\frac{3}{2}(m-1)\rfloor + 1\right)$.

Proof: The cost of $\mathrm{HDS}_{(n,m)}$ graph is the product of its *Degree and Diameter*. So the cost of $\mathrm{HDS}_{(n,m)}$ graph is given as

C = $(n + m - 1)\left(\lfloor\frac{3}{2}(n-1)\rfloor + \lfloor\frac{3}{2}(m-1)\rfloor + 1\right)$ (5)

Theorem 6: The bisection width of the HDS network topology is $B_w = 2^m \times 2n!$

Proof: The bisection width helps to determine the distinct sub graphs from a single network by removal of certain number of edges. In HDS topology there are m! number of clusters with star graph inter connection. From the construction it is clear that there is a need to remove 2^m number of edges connecting 2n! nodes. Thus removal of $2^m \times 2n!$ Edges will result in disjoint HDS networks of just previous dimension. Hence, the bisection width is given by

$B_w = 2^m \times 2n!$ (6)

Theorem7: The HDS(n.m) network is regular and vertex symmetric Cayley graph.

Proof: The HDS network is a derived graph with base network is a star graph. Each vertex has equal degree as shown in Theorem 1. Next from the construction it is shown that the HDS being a product graph, the vertex has address as 2n bit permutations and the adjacent vertex is obtained by changing any bit positions similar to star graph. Since each vertex of HDS(n,m) satisfies the above condition, so the HDS is proved to be regular and vertex symmetric Cayley graph.

Table 1: Comparison of Topological Properties of Hypercube, Star, DS, HS and HDS

Parameters	Hypercube (n)	n-star	Double star(n)	Hyper star (n, m)	Hyper double star (n, m)
Degree	n	$n-1$	n	$n+m-2$	$n+m-1$
Node	2^n	$n!$	$n!+n!$	$n!\times m!$	$2n!\times m!$
Links	$n\times 2^{n-1}$	$n!(n-1)/2$	$n^2-n+n!$	$(n+m-2)\times n!\times m!/2$	$2m!(n^2-n+n!)$
Diameter	n	$\left[\frac{3(n-1)}{2}\right]$	$\left[\frac{3(n-1)}{2}\right]+1$	$\left[\frac{3(n-1)}{2}\right]+\left[\frac{3(m-1)}{2}\right]$	$\left[\frac{3(n-1)}{2}\right]+\left[\frac{3(m-1)}{2}\right]+1$
Cost	n^2	$\left[\frac{3(n-1)}{2}\right]\times(n-1)$	$n\left\{\left[\frac{3(n-1)}{2}\right]+1\right\}$	$(n+m-2)\left\{\left[\frac{3(n-1)}{2}\right]+\left[\frac{3(m-1)}{2}\right]\right\}$	$(n+m-1)\left\{\left[\frac{3(n-1)}{2}\right]+\left[\frac{3(m-1)}{2}\right]+1\right\}$

Performance Measures of Proposed Network (HDS)

Any parallel algorithm design must consider both time and cost-effectiveness in order to be successful. For this reason, we need to analyze the performance parameters of the particular network. The performance analysis determines the success or failure of a project forecast using various parameters. From performance analysis we can determine the total cost of a system as it reproduce important aspects of a multiprocessor system. To make the network more attractive, here we more emphasis on CEF and TCEF factors of HDS network.

Cost Effectiveness Factor

The cost-effectiveness of a parallel algorithm is considers the both of cost of processors and the cost of communication links. It is the ratio of cost effectiveness and efficiency of a network.

Theorem8: The cost effectiveness factor of $HDS_{(n,m)}$ is CEF(p) = $\frac{1}{1+\rho\times\frac{2m!(n^2-n+n!)}{\{2n!\times m!\}}}$

Proof : In general the number of links is a function of the number of nodes that is E = f(p)

The total number of processors is: $v = 2n!\times m! = p$

And the total number of links in EDS = $2m!(n^2-n+n!)$

$$=> E=\frac{v}{n!}(n^2-n+n!)=f(p)$$

Again, $g(p)=\frac{f(p)}{p}=\frac{\frac{v}{n!}(n^2-n+n!)}{p}$

Where g(p) is the ratio of number of links to the number of processors and $\boldsymbol{\rho}$ is the ratio of link cost to processor cost.

$$Hence, CEF(P)=\frac{1}{1+\rho\times g(p)}=\frac{1}{1+\rho\times\frac{2m!(n^2-n+n!)}{\{2n!\times m!\}}}$$

Time Cost Effectiveness Factor

Time cost effectiveness factor considers time for solution of a problem as a parameter for evaluating the performance. This factor evaluates circumstances in which a faster solution is preferable to a slower one. The above two factors characterize the profitable utilization of multiprocessor systems when the performance of parallel algorithms is known.

Theorem 9: The time cost effectiveness factor of the HDS(n,m) network is expressed as follows:

$$\frac{(1+\sigma)}{1+\rho\times\frac{2m!(n^2-n+n!)}{\{2n!\times m!\}}}$$

Proof: The Time Cost Effectiveness Factor in HDS(n,m) is derived as follows

$$TCEF(P) = \frac{(1+\sigma)}{1+\rho g(p)+\left(\frac{\sigma}{p}\right)}$$

Where $\boldsymbol{\rho}$ and g(p) are same as defined in Theorem 8.

$\boldsymbol{\sigma}$ is the ratio of cost of penalty to the cost of processors.

In Theorem:9, the value of $g(p) = \frac{2m!(n^2 - n+n!)}{2n!\times m!}$

As we know that $TCE(P) = \frac{(1+\sigma)}{1+\rho g(p)+\left(\frac{\sigma}{p}\right)}$

$$\textit{Hence}, TCEF(p, HDS) = \frac{(1+\sigma)}{1+\rho\times\frac{2m!(n^2-n+n!)}{(2n!\times m!)}}$$

Broadcasting in HDS

Broadcasting can be regarded as a communication pattern in the network where message from a node is copied to all other nodes. Availability of a link from each PE/NC to the NC at the next higher level enables very efficient communication. The broadcast process has a wide range of uses in parallel computing and distributed system control. For instance, in parallel computing, several tasks are distributed across different PEs, and the outcomes of these activities must be updated at other processors in order for the processing to proceed further. To enable such communications different algorithms namely one-to-all and all-to all broadcast are planned for the HDS network in this segment.

Broadcast Algorithm for HDS

Suppose S is the source node

D is the destination node

N_s is the neighbor of the source node

N_d is the neighbor of the destination node

The following two conditions must be satisfied by the proposed broadcasting process

- A node element can only transfer or (receive) the message to (from) one of its respective neighbors.
- To prevent message duplication, a processing element can only accept a message once throughout the entire communication process.

One-to-All

Step1: Initialization

Select a source vertex S in the hyper double star network.

Initialize a message queue and a visited set.

Enqueue S into the message queue.

Mark S as visited.

Step2: Single Step Broadcast

While the message queue is not empty, do the following:

Dequeue a vertex V from the message queue.

Process the message at V.

For each neighboring vertex N connected to V in the hyper double star topology:

Check if N has been visited.

If N has not been visited, enqueue N into the message queue and mark it as visited.

Step3: Cluster Broadcast

Identify the loaded nodes in the hyper double star topology (if any).

For each loaded node, perform the one-to-all broadcast algorithm described in steps 1 and 2 within the cluster.

Choose a source vertex within the cluster.

Apply Step 1 and Step 2 for the selected source vertex, considering the intra-cluster connectivity.

All-to-All

To achieve all-to-all broadcasting in the hyper double star topology, each node can individually execute the one-to-all broadcasting algorithm described above. By selecting different source vertices for each node and running the algorithm independently, messages can be broadcasted throughout the entire network.

Routing Time Complexity

The word simply refers to how long it takes for an algorithm to run; its length is determined by counting the number of simple operations the algorithm performs. For HDS, the routing and broadcasting algorithms are created with minimal time complexity. The proposed one to one routing algorithm of HDS has time complexity $O(n + m + 1) \cong O(n + m)$, when n is too large. In comparison to the HC and Star networks, the proposed HDS with a greater number of nodes, exhibits lower time complexity. It is explained through the spanning broadcasting tree (SBT) as shown in the Fig. 4 and 5 for HDS(3,3). The height of the SBT is 4 that is n+1, for n=3. It depict the best case time complexity.

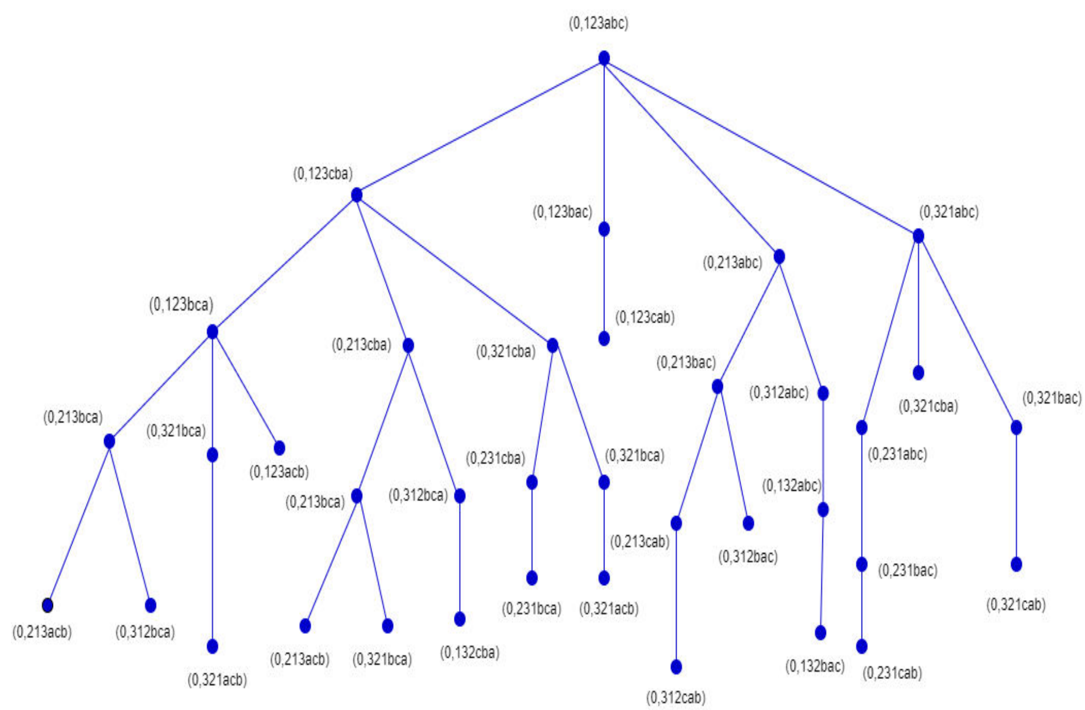

Figure 4: Spanning Broadcast Tree (SBT) of HDS(3,3)

Optimal Routing

Routing refers to the minimum distance path from a node P to another node Q in $HDS_{n_1,n_2 \ldots n_k}$ can be constructed by first including a sequence of E_1 Edges that corresponds to a minimum routing from $\rho_1(P)$ to $\rho_2(Q)$ in HDS_{n_1} followed by a sequence of E_2 edges to a minimum routing from $\rho_1(P)$ to $\rho_2(Q)$ in $HDS_{n_2 \ldots}$ And so on. This resultsis an optimal distributed routing algorithm that routes message in the product graph HDS.

Algorithm 1-1HDSRouting

{The message msg is originally at source node P}

$done_i$ *= false for 1≤ i≤k*

If address = P then meg_in = true else

msg_in = false

for I = 1 to k do

if ρ_j *(address) =* $\rho_j(P)$, *for all j such that i+1≤* $j \leq k$ *then*

if not done, then

if not msg_in then Receive (msg, $Parent_{n_i}(Z_i, \rho_i(address))$

destinated_to _many (msg, $Children_{n_i}(Z_i, \rho_i(address))$

done = true

msg_in = true

endif

endif

endfor

endMSG

***

5 RESULTS AND DISCUSSIONS

This section attempts to establish the significance of the proposed Hyper Double Star graph interconnection network. The different topological parameters are derived and compared with those of hypercube, star graph, double star graph and hyper star graph. All the parameters are listed in Table 1.

The Fig. 4 shows the comparison of degree in the proposed topology. It clearly indicates that the degree of HDS graph is high as compared to hypercube, star, double star and hyper star graph. The HDS being a product graph helps to bridge the gap between the consecutive dimensions unlike star graph.

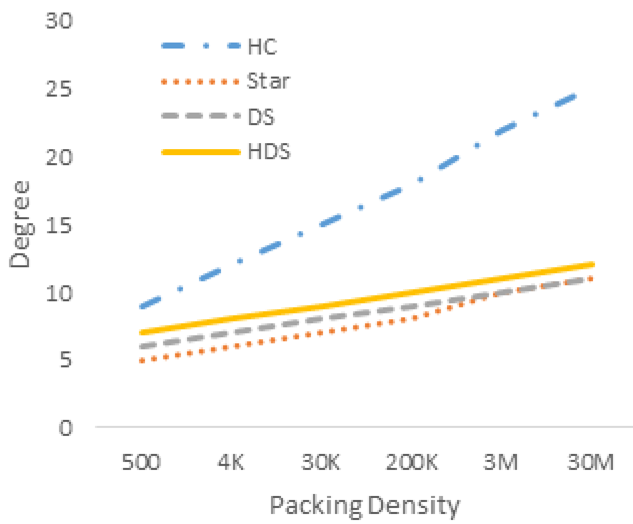

Fig. 4: Degree Comparison Against Packing Density

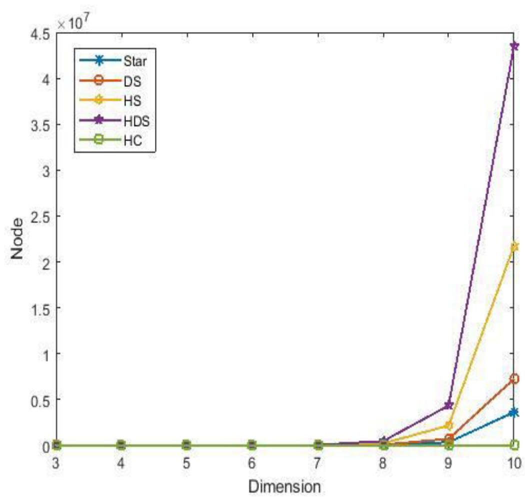

Fig. 5: Node Comparison

The Fig. 5 shows the comparison of number of node in proposed topology. It shows HDS graph has more number of node as compared to hypercube, star, double star and hyper star graph at same node degree.

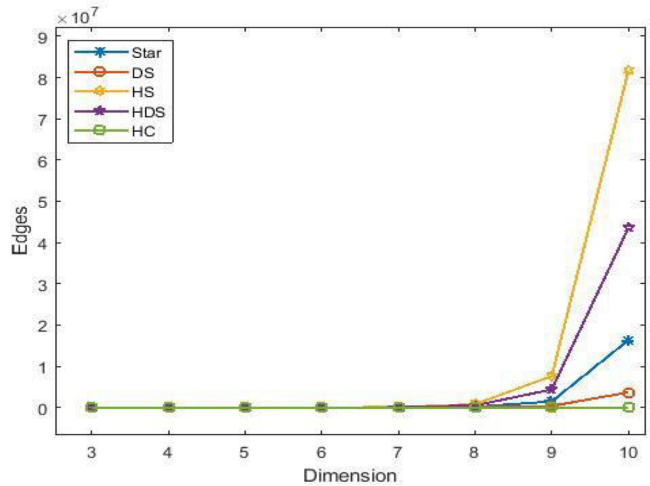

Fig. 6: Edge Comparison

The Fig. 6 shows the comparison of edges in proposed topology. It shows that HDS graph has more number of edges as compared to hypercube, star and double star graph and less number of edges as compared to hyper star graph. It falls in between and will have reduced link complexity while packing a sufficiently large number of vertices.

The Fig. 7 shows the comparison of diameter in proposed topology. It shows that the diameter of HDS is high as compared to hypercube, star, double star and hyper star graph. The comparison revels that travelling in the HDS network will be faster and cost effective with regard to message passing in massive parallel systems.

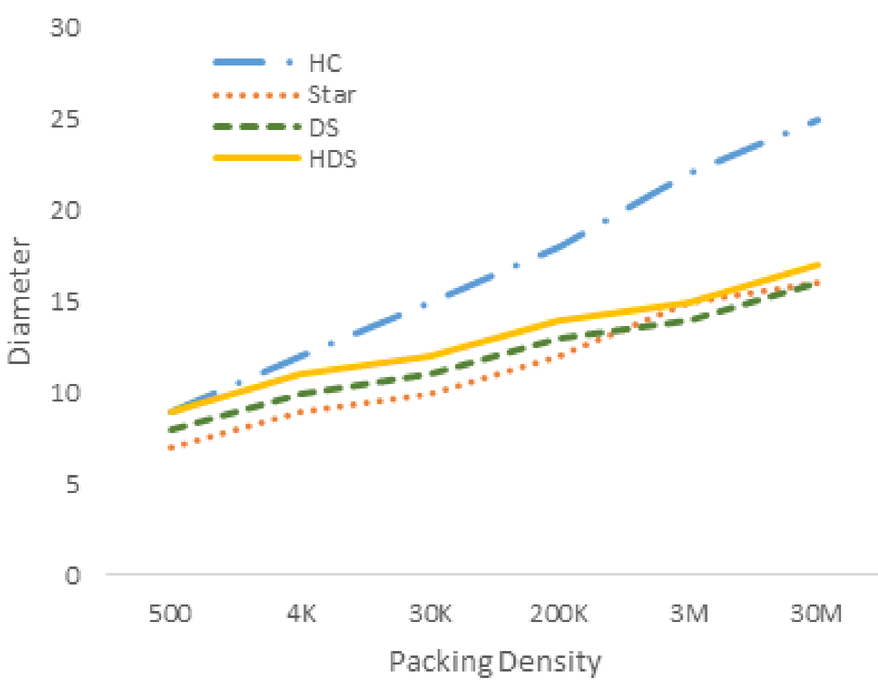

Fig. 7: Diameter Comparison with Packing Density

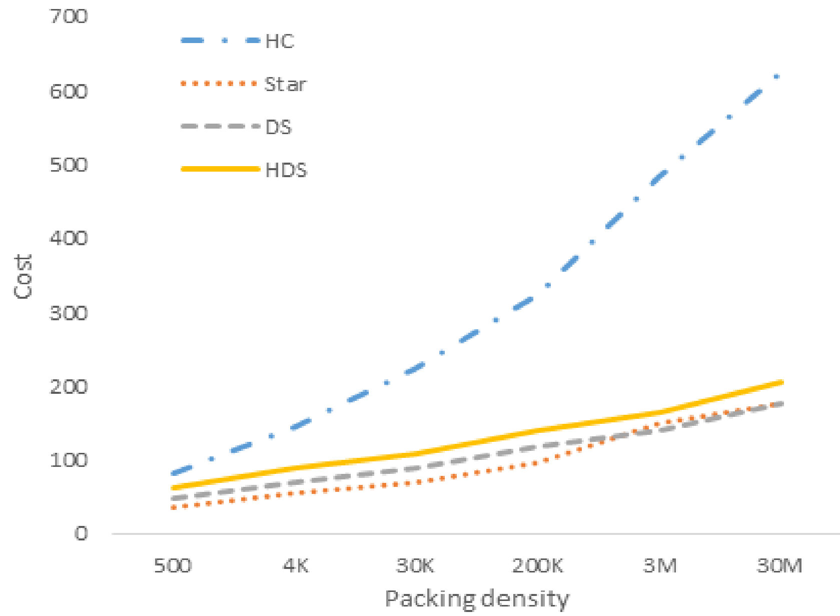

Fig. 8: Cost Comparison Against Packing Density

The Fig. 8 shows the comparison of cost in proposed topology. It shows that the cost of HDS packs more number of nodes at same cost in comparison to the parent graphs such as hypercube, star, double star and hyper star graph. The Fig. 9 depicts the comparison of routing time complexity of HDS topology with other contemporary networks. The major advantage of HDS is that it covers more number of nodes in less time which is a significant criterion in high speed parallel computing scenario. The hypercube does same at a very higher dimension. In the same time the star graph covers very less number of nodes as both lines converge around dimension 11 and 10 respectively.

Fig. 9: Routing Time Complexity Comparison with Packing Density

Table 2: The Degree and Diameter of the Hypercube, the Star Graph, the double star and the Hyper double star for Different Orders of Graph Sizes

	Smallest Hypercube			Smallest Star Graph			Smallest Double Star graph			Smallest Hyper Star Graph		
Packing density	Dim.	Deg.	Diam.	Dim.	Deg.	Diam.	Dim.	Deg.	Diam.	Dim.	Deg.	Diam.
500	9	9	9	6	5	7	6	6	8	(6,2)	7	9
4K	12	12	12	7	6	9	7	7	10	(7,2)	8	11
30K	15	15	15	8	7	10	8	8	11	(8,2)	9	12
200K	18	18	18	9	8	12	9	9	13	(9,2)	10	14
3M	22	22	22	11	10	15	10	10	14	(10,2)	11	15
30M	25	25	25	12	11	16	11	11	16	(11,2)	12	17

Table. 2 shows that degree and diameter of the hypercube, star graph, and hyper double star for various graph sizes. From Table.2 it is clear that, the hyper double star graph has better fit to the desired size than the star graph, double star graph and a lower degree and diameter than the hypercube. It indicates that the proposed HDS topology has better scalability and hence, is a better candidate for large scale high performance computing systems.

The cost effectiveness factor and time cost effectiveness factor with respect to network dimension are described in the figure 14 and 15 respectively. Similar to the Hypercube, THC also exhibits a monotonically decreasing trend on ρ. As a result, the network's cost effectiveness and time cost effectiveness decrease as its size increases. Hence the proposed THC network is definitely a cost-effective as well as a time cost-effective topology for massive computing.

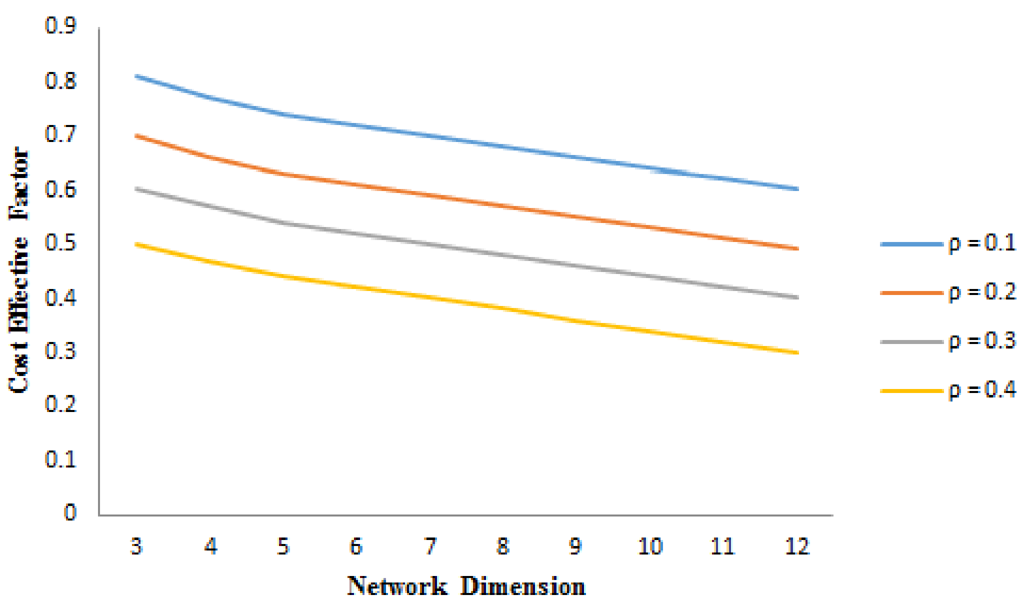

Fig. 10: Comparison of CEF wrt Network Dimension

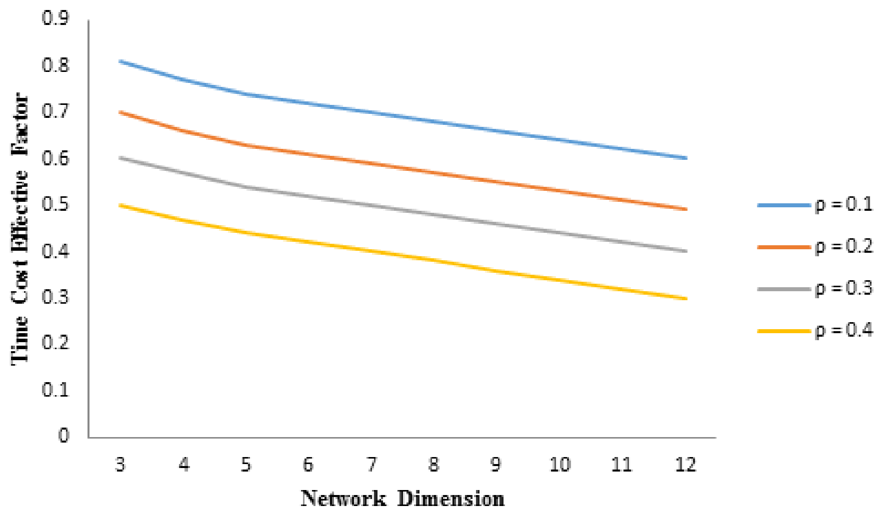

Fig. 11: Comparison of TCEF wrt Network Dimension

6 CONCLUSION

In this paper we have proposed the hyper double star parallel interconnection topology which is a Cartesian product of star graphs. For this interconnection topology various topological parameters are established like degree, node, edges, diameter and cost. The hyper double star network is an improvement over the star graph and hypercube for the following reasons: First, the hyper double star graph is regular symmetric better than the star graph and the hypercube graph. Second, the hyper double star can be easily reconstructed from its lower dimension with comparatively low link complexity. Third, the packing density is more at low node degree than the parent networks. Fourth, the hyper double star is a highly scalable and robust parallel interconnection network. In addition, it retains the original structure of star graph. In summary, the hyper double star improves on the attractiveness of the star graph as an interconnection network topology for large multi-processes system by solving its major drawbacks of poor scalability and suitability for real applications while preserving all of its attractive features. Routing in HDS is also faster while covering almost double size processing elements at all dimensional values.

REFERENCES

1. Y. Saad, M.H. Schultz, "Topological properties of hypercubes", IEEE Transactions on Computers, Vol. 37, pp.867–872, 1988.
2. T. Feng, "A survey of Interconnection Networks", IEEE Computers, 1(4), pp. (12-27), 1981.
3. C F Tsai, W C Lin and S. W. Ke, "Big Data Mining with Parallel Computing: A comparison of Distributed and Map Reduce Methodology", Journal of Systems and Software, Volume 122, Pages 83-92, 2016.
4. 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.
5. 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.
6. 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).

7. K. Day and A. Tripathy, “A Comparative Study of Topological properties of Hypercubes and Star Graphs”, IEEE Trans. Parallel & Distributed Systems, Vol.5.
8. Al-Ayyoub, A. E., & Day, K. (1998). The Hyperstar Interconnection Network. Journal of Parallel and Distributed Computing, 48(2), 175-199.
9. Konstantin Shvachko; Hairong Kuang; Sanjay Radia; Robert Chansler, ”The Hadoop Distributed File System”,IEEE-2010
10. Weizhong Zhao, Huifang Ma Qing He, ”Parallel K-Means Clustering Based on MapReduce”, Springer – 2009
11. Dhruba Borthakur, ”The Hadoop Distributed File System: Architecture and Design”, 2007
12. Abuelrub, Emadeddin Mohamed, ”Interconnection Networks Embeddings and Efficient Parallel Computations.” (1993). LSU Historical Dissertations and Theses. 5554.
13. K. Efe, “The crossed cube architecture for parallel computation”, IEEE Transactions on Parallel and Distributed Systems, Vol. 3, pp. 513–524, 1992.
14. D. Sarkar, “Cost and time cost effectiveness of multiprocessing”, IEEE Transaction on Parallel and Distributed Systems, Vol. 5, pp. 704–712, 1993.
15. N. Imani and H. Sarbazi-Azad, “Properties of a Hierarchical Network Based on the Star Graph”, Information Sciences, vol. 180, no. 14, pp. 2802–2813, 2010.