ON IMPROVING THROUGHPUT AND LATENCY OF MESH INTERCONNECTION NETWORKS

Thesis submitted in fulfillment of the requirement for the Degree of

Doctor of Philosophy

By

AKASH PUNHANI

Department of Computer Science & Engineering and Information Technology

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY

Waknaghat, Solan-173234, Himachal Pradesh, INDIA

July, 2017

DECLARATION BY THE SCHOLAR

I hereby declare that the work reported in the Ph.D. thesis entitled "**ON IMPROVING THROUGHPUT AND LATENCY OF MESH INTERCONNECTION NETWORKS**" submitted at **Jaypee University of Information Technology, Waknaghat, India** is an authentic record of my work carried out under the supervision of **Dr. Pardeep Kumar** and co-supervision of **Prof. (Dr.) Nitin**. I have not submitted this work elsewhere for any degree or diploma.

(Signature of Scholar)

Akash Punhani

Enrollment No: 136203

Department of Computer Science and Engineering and Information Technology

Jaypee University of Information Technology

Date:

Certificate

This is to certify that the thesis entitled, "**ON IMPROVING THROUGHPUT AND LATENCY OF MESH INTERCONNECTION NETWORKS**" which is being submitted by **Akash Punhani** in fulfilment for the award of degree of **Doctor of Philosophy in Computer Science and Engineering** by **Jaypee University of Information Technology, Waknaghat, India** is the record of candidate's own work carried by him under our supervision. This work has not been submitted partially or wholly to any other University or Institute for the award of this or any other degree or diploma.

Dr. Pardeep Kumar

Assistant Professor (Senior Grade)

Department of Computer Science & Engineering

JUIT, Waknaghat, Solan

Himachal Pradesh, India

Pin -173234

Nitin

Prof. Dr. Nitin

Professor

Department of Computer Science & Engineering

JIIT, Noida

Uttar Pradesh, India

Pin -201309

ACKNOWLEDGEMENT

It is immense pleasure to thank all those who have helped me and encouraged me in the completion of my research work leading to my doctorate degree in Computer Science and Engineering.

I want to express my heartfelt gratitude to my supervisor **Assistant Professor (Sr. Grade) Dr. Pardeep Kumar**, Department of Computer Science and Engineering, for his valuable guidance, support and encouragement throughout my research. I also want to thank my co-supervisor **Prof. (Dr.) Nitin**, Department of Computer Science and Engineering, JIIT for his guidance and showing patience to my queries related to the research work. This would not be wrong to say that they both have treated me as a younger brother.

I would also want to thank **Prof. (Dr.) S.P. Ghrera** (Head, Department of Computer science & Engineering and Information Technology), **Dr. Pradeep Singh**, **Dr. Shailendra Shukla**, **Prof. (Dr.) Karanjeet Singh** (Head, Mathematics Department), **Dr. Vivek Kumar Sehgal** and other DPMC members for providing their valuable suggestions and guiding me to the right path.

I would like to give special thanks to **Prof. (Dr.) Vinod Kumar,** Vice Chancellor of Jaypee University of Information Technology and **Prof. (Dr.) Samir Dev Gupta**, Director and academic head for providing an excellent research environment and learning ambiance at the campus.

I want to thank all the lab staff for providing the necessary support in the laboratories to conduct the experiments.

I would like to thanks **Dr. K.K. Mishra** Assistant Professor MNNIT Allahabad, **Prof. (Dr.) Shailesh Tiwari**, **Dr. Geeta Sikka**, Head and Associate Professor Dr. B.R. Ambedkar NIT Jalandhar for supporting me in various issues related to research and **Prof.(Dr.) D.S. Chauhan** VC, GLA University for his moral support to me.

The support provided by my colleague **Dinesh Kumar** is worthy and I would like to thank him for support he has provided.

I would like to thanks the reviewers of my thesis for providing me valuable suggestions in improving the quality of my thesis.

I am eternally grateful to my respected Father, Mother and affectionate younger brother **Rajeev Punhani** because without their sustained moral support it would have been impossible to succeed in writing this thesis.

TABLE OF CONTENTS

ABSTRACT

The primary components of digital system comprise of processing elements and memory units and interconnection networks are used to connect these components. With recent advancements in technology, it has become imperative to explore the high-speed interconnection networks due to the progress in the domains of processor and memory units. Initially, the bus-based architecture was used to connect the various nodes, however, with the increase in number of nodes, these buses became a bottleneck; and therefore, the dedicated buses were used. Regardless, the usage of large number of dedicated buses led to an increase in the mesh of wires, which created complexity in the implementation of the network. The idea of routing packets was hence initiated for the development of the mesh topology based on tile architecture. Again, this topology went through a large number of variants, yet most of them tried to reduce the diameter by introducing lengthy links on the topology, thus affecting the scalability of the topology. The diagonal (toroidal) mesh has long links and gets disconnected for an even number of nodes. Due to this, a modified diagonal mesh interconnection network is introduced in this research, which uses horizontal and vertical links on outer edges in place of toroidal links. This idea removes the drawback of the topology and determines that the diameter can only represent the upper bound of the distance. Further, the shuffle exchange network, introduced to the modified diagonal mesh interconnection networks, helped in reducing the average internode distance. The results pertaining to the proposed topology are compared with the torus topology. Apart from torus topology, another topology that reduces toroidal links via the combination of diagonal connected mesh and T-mesh topology is compared with the proposed work. The results revealed that the suggested topology performs better than the existing two topologies. For increasing the scalability, the toroidal links must be placed in an optimal manner. Further, the computational complexity can be reduced by using the heuristic search techniques. The proposed topology uses improved environmental adaptation method and is found to perform more efficiently than a topology with the same number of links. While the hardware performance is not eminently relevant here, the performance is also affected by the algorithm using it, which is the routing algorithm. The routers in the interconnection are miniature routers and have a limited amount of memory, where simple logics can be deployed. The most popular routing algorithm is XY routing algorithm, which is less efficient than the level based routing. The present study therefore

suggested the use of level based routing via dynamic programming so as to acquire productivity in terms of time and space complexity. Also, modified center concentrated mesh provides shortest path in case of no congestion, a feature that is less explored. In the previous algorithm, in cases of congestion, the routers used adaptive routing algorithms. These routing algorithms require special hardware to get the stress signal, which increases the complexity of routers and links in the network. Therefore, we proposed the usage of entropy to estimate stress generated in a particular direction. In an attempt to maintain the state of equilibrium, the algorithm has provided competitive results when comparing with the other routing algorithms in terms of throughput and latency.

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1
INTRODUCTION

1.1 Introduction of Interconnection Networks

A digital system consists of three components, the processors, memory units, and the interconnection networks that connect processors, intellectual property blocks, and memory units for communication. With the advancement of technology, the systems have become rapid and complex. The components that participate in processing, such as processors, memory units and digital processing units are defined as cores or processing units[1]. Previously, systems had adopted the bus-based topology in the interconnection networks and the interconnection buses either were shared among various cores or, in cases of high demand for communications, were dedicated buses from one core to another core. As most of the cores possess high communication demands, it resulted in routing of the wires from one node to another. With the advancement of technology, multiple cores can be designed within a single chip and with this the phase of the System on Chips (SOCs)) was initiated. The interconnection network for on-chip communication is termed as, Network on Chips (NOC), and the performance of an interconnection network depends on three key components, and they are as follows:

1. Topology
2. Routing Algorithm
3. Flow Control mechanism

1.1.1 Topology

Topology describes the process of establishment of connections between various nodes with communication links, in the interconnection networks. Mathematically, the topology is a graph *G(V, E)* where *V* is the set of the vertices and *E* is the set of edges. The direct and indirect topologies are the two significant classifications used in interconnection networks. In direct topology, every node is directly connected to its immediate neighbour whereas, in indirect topology, a layer of switches exists between the two nodes. In the current discussion, the focus of the study is mesh topology, which belongs to the family of direct topology.

1.1.2 Routing Algorithm

The topology defines the physical paths that exist between two nodes and a topology may possess more than one physical path from one node to another node. The routing algorithm defines a path from the source to the destination that a particular packet should follow. The routing algorithm is responsible for the connectivity, adaptivity, deadlock, livelock and fault tolerance in the interconnection networks[2].

1.1.2.1 Classification based on the path

Based on the path selection by the routing algorithm, the routing algorithms are categorised as follows:

1.1.2.1.1 Deterministic Routing Algorithm

In the case of deterministic routing algorithm, the router selects a fixed path to a particular destination, irrespective of the state of network. Deterministic routing algorithms are easy to implement and are free from deadlock, but will not produce better performance due to the lack of uniformity in the load distribution over the links of the network [1–3].

1.1.2.1.2 Oblivious Routing Algorithm

The oblivious routing algorithms are independent from the present network state. The deterministic routing algorithms are a subset of the oblivious routing algorithms, but the oblivious routing algorithm attempts to balance the network load by sending the traffic to any random node and passes it to the destination node. It may also include some extra hops in comparison to the shortest path between the source and destination [1, 4].

1.1.2.1.3 Adaptive Routing Algorithm

The adaptive routing algorithms are dependent upon the state of the network, thereby improving the performance of the network. In general, this routing algorithm does not return a single path like the next hop, but returns multiple paths. The single path is selected using the criteria set by the underlying application [1, 5–8]. The popular routing algorithms are IX/Y Routing algorithm and Odd even routing algorithm.

1.1.2.2 Classification based on the routing algorithm

The data structure used in the routing algorithm plays a significant role in the process implementation. They are further classified into two categories:

1.1.2.2.1 Table based Routing Algorithm

In the table-based routing algorithm, a path from source node to the other nodes is stored in a table. The storage of the route requires memories that are proportional to the number of nodes, which impacts the cost and scalability [9].

1.1.2.2.2 Finite State based Routing Algorithm

The finite state based routing algorithm attempts to minimise the memory consumption, by routing a packet from the source to destination based on logics or mathematical formulation [2, 10].

1.1.3 Flow Control Mechanism

The Flow Control mechanism determines the allocation of the networks to the packet, travelling the network. The flow control aids in ensuring the maximum throughput and delivery of packets with minimum latency. The most popular techniques in the flow control mechanism are:

1. Bufferless Flow control Mechanism: In the bufferless flow control mechanism, the packet may drop or misroute if the requested channel is not ideal at that particular moment of time [11].
2. ON/OFF Flow Control: It is a buffer-based approach. Under this method, the receiving node will send a stop signal if a lack of space arises, and the flow will initiate again only after receiving the continue signal [1].
3. Credit based flow control: Also, a buffer based approach; the credit based flow system sender is limited to sending flits only in the occupancy of credits from the receiver side based upon the buffer allocated to the sender in order to control the buffer overflow. The number of flits sent in a particular direction will be equal to the credit available with the router sending the flits [2].

1.2 Motivation to Our Approaches

The race of designing the supercomputers and the faster computer is running at an accelerated pace. These computers have multiple cores that are interconnected through the interconnection network. The central and key component on which the computer performance depends reflects the connection and communication of the cores. As the various Intellectual Properties Blocks are designed to perform at extreme capabilities, its performance as a complete system depends upon the interconnection network used. While designing the system on a chip, the tile based architecture has proved to be better in comparison to other approaches, as it utilises the optimal area requirement which is a crucial part of any VLSI[12] design. Reflecting the characteristics of cloud on a chip, the new and advanced chips also utilise the mesh architecture[13]. Therefore, this study attempts to develop topologies based on the basic idea of mesh topology and provide better performance. The next important point under discussion is that, even though the topology is efficient, it cannot work efficiently unless the routing algorithm is efficient and practical. Hence, a need to study both the topology and the routing algorithm arises.

1.3 Research Gaps Identified

1. Torus topologies have long edge lengths that are against the property of constant edge length and affect the scalability of the topology. However, the number of lengthy toroidal links should be avoided as it generates the scalability issue.
2. The links in the variants of the mesh are placed using the human perception that the placement will reduce the overall distance of the topology. However, there is a possibility of finding the optimal placement using an iterative search, therefore requiring an examination of the techniques to assess the efficiency of the links.
3. The deterministic routing algorithms can be further optimised to get the fast processing time and should try to utilise the available links in the network rather than targeting any specific links.
4. The adaptive routing algorithms require the stress signals to avoid the congestion. The detection and communication of stress signal will add an extra cost of links and cost of hardware required for management.

1.4 Problem Statement and Contributions

The current thesis explores the mesh interconnection network and its variants. Initially, the problem of the large toroidal link, which is a contradiction to the basic property of fixed edge length and disconnection of the diagonal mesh (toroidal) interconnection on the even number were studied [14–16].

From the study of the basics of interconnection networks and diagonal mesh interconnection network, the authors of the thesis have suggested replacing all the lengthy toroidal links from the diagonal mesh interconnection networks, with the horizontal and vertical links on the boundary nodes. The topology suggested has the potential to substitute the torus topology as it has limited number of toroidal links. The modified topology has the advantage of connectivity, for both odd and even number of links. Even though this was the scene, the modified diagonal mesh interconnection network (MDMIN) underperformed in terms of throughput and latency in comparison to the torus and Diagonal mesh topology. To overcome this drawback, shuffle exchange network on the boundary nodes were introduced. Further, the three-dimensional topology based on the two-dimensional MDMIN topology has been designed as well by authors of the thesis [16].

The diagonally connected mesh focused on adding the extra links diagonally in the two dimensional mesh. Though the diagonal links were added to reduce the diameter of the topology, it still is large in comparison to the other topologies[17]. With the objective of reducing the diameter further, few links have been introduced in the diagonally connected mesh and the research work regarding the same has been published[17].

The Diagonally connected T mesh (DCT) topology is the union of the two topologies with the aim of reducing the diameter of the resultant topology. The topology is designed using two singular topologies; that is, Diagonally connected mesh (DCM) and T-mesh. Both the topologies had adopted different approaches with the aim of reducing the diameter. DCM used the diagonal links and T Mesh used the toroidal links at the corner edges. The DCT topology uses both toroidal at corner edges and diagonal links at the same time. The cost of the topology will be slightly elevated than that of DCM, due to the addition of four extra links. But, the router complexity will not change as the maximum degree of the node does not exceed the maximum degree of router used in the network. The bisection bandwidth has increased, asserting that the topology is now increasingly fault tolerant[17].

While solving the above-described problem, few observations suggested that by reducing the diameter alone, the performance of the topology cannot be improved. Because, reducing the diameter produce a minor change in the path length or average internode distance of the topology. Secondly, the exploration of the placing of links is a complex search problem and to explore the topology iteratively, more time is needed. Therefore, motivating author of the current thesis to get inclined toward the soft computing approaches to perform heuristic search in the huge search space. The contribution regarding the same has been published in [18]. The author of the thesis suggested using the average path length as the fitness function for the heuristic search algorithms. The two search algorithms adopted to search the optimal links are genetic algorithm and Improved Environmental Adaptation Method(IEAM). The reason for selecting IEAM is becuase it is fast in comparison to various other search techniques[18].

The performance of the interconnection is not dependent on the topology used alone, but is also dependent on the routing algorithm. Most of the mesh interconnection networks use the destination-based routing algorithm and the most prominent routing algorithm for the simple mesh interconnection network is, the XY Routing algorithm. The Level based routing algorithm has challenged the performance of the XY routing algorithm and it uses ID's as addresses of the nodes. The algorithm uses division and modulus operator to calculate the port address instead of the subtraction operator used in XY routing algorithm, which to an extent is not efficient as claimed in the publication [19]. The deterministic routing algorithm is expected to follow the same path for every packet from specific source to destination and should be the shortest distance. The center concentrated mesh uses the diagonal links to the center nodes for reducing the diameter, but the routing algorithm suggested is not using the topology efficiently and following the longer routes.The contribution to this research work has been published in [20, 21]

The authors of the thesis have suggested a modification, asserting that the packets should not be directed to the center in the instance of a shortest path to the destination. The authors compare the routing algorithm with the table based approach which requires more memory, thereby describing the importance of the finite state routing machines. The authors of the thesis also compared the performance of the routing algorithm with odd-even routing algorithm[20]. The importance of the routing algorithm motivated the author to study and improve the routing algorithm at computational level to make it faster. And the authors have identified the repetition of the computation steps in the Level based routing algorithm and have applied dynamic programming to reuses the solution provided at the previous computations at the next stage of computations.

Storing the node index and level after the first computation to reduce the computation cost at router were suggested [21].

1.5 Thesis Road Map

The main contribution of the thesis is elaborated with the help of the various chapters. In the initial part of the thesis Chapter 1, the importance and organisation of the thesis is elaborated. Chapter 2 provides the deep background of the mesh interconnection networks. Chapter 3 gives the detailed insight of Modified Diagonal Mesh Interconnection Network both in the two-dimensional and three-dimensional perspectives. Chapter 4 describes the technique for optimal link selection in the mesh. Chapter 5 describes the Center concentrated mesh routing algorithm pitfalls and suggested the Modified Center Concentrated Mesh routing algorithm. Chapter 6 gives the detailed insight of the dynamic programming based routing algorithm that can prove to be efficient in taking the routing decision.

CHAPTER 2
LITERATURE SURVEY

The period of 1960 to 1970 witnessed the advent of major design implementations in the field of parallel processing. To date, various parallel processing technologies such as parallel hardware architectures, interconnections networks and programming paradigms are available in the market [22]. The Flynn's taxonomy has classified the parallel computer into four categories based upon the way the instruction executes, and the system processessing of data. In the 1980s, a search for distributed architectures was initiated in which the independent processor and local memory were connected using the interconnection network. Efforts were made by many researchers to accommodate a large number of processors in the architecture and to satisfy the demands of the application using these architectures[23]. The researchers suggested various interconnection networks like rings, mesh, trees and hypercube and some advanced topologies like reconfigurable topologies based on the underlying programming logic[1].

From the topologies mentioned above, the mesh topology is the two-dimensional topology that is based on tile architecture. Tile based architecture suggests the fixed allocation of the rectangular area for cores and the additional space for global wires for the communication between the cores [12]. The tile structure has the potential to improve the performance and the modularity of the complete system [12]. The global wiring system can aid in optimising the electrical properties of the system, which in turn reduces the heat dissipation by ten times and increases the propagation velocity by three times[24]. Direct and Indirect networks are the two classifications of interconnection networks, and the difference between the two can be understood through the definitions given below:

Definition 1: Direct Interconnection network is the type of network under which the node is directly connected to its adjacent neighbour, due to which the packets are transferred precisely between the nodes [1].

Definition 2: Indirect Interconnection network is the type of network under which the terminal nodes are not directly connected. But, the nodes are connected to the switching nodes which results in the indirect forwarding of packets [1].

Mathematically, the direct interconnection network is a graph *G (V, E),* where *'V'* represents the set of nodes and *'E'* represents the channel connecting them.

2.1 Topological Properties of Direct Interconnection Networks

The performance of direct interconnection networks is a critical factor that determines its effectiveness. The four elements that describe the performance are network degree, diameter, edge length and bisection width. The definitions of the four properties are as follows:

Definition 3: Network degree is the maximum number of ports per node, for all nodes in the network [1].

Definition 4: Network diameter: Diameter of a network (D) is largest, minimal path length over all pairs of nodes in the topology. [1].

Definition 5: Bisection Width of network is the minimum number of links that are to be removed from the network, when the network is divided into two equal sets of nodes [25].

Definition 6: Edge Length is the advised length of the connection in the networks, in order to keep the length of the edge of fixed size; as it may affect the scalability of the network[26].

Along with these properties, the routing algorithm and flow control mechanism also influences the performance of the interconnection network.

2.2 Various Two Dimensional Topologies based on the Mesh Topologies

2.2.1 Two-dimensional Mesh Interconnection Networks

It is the simplest form of the mesh interconnection network, and the nodes are connected by horizontal and vertical connections. The number of nodes in X and Y direction can perchance be a positive integer, say m and n respectively. The total number of nodes in the topology is m × n. The degree of nodes in the mesh is two for the corner nodes, three for the outer nodes except the corner nodes and four for the remaining nodes. Figure 2.1 describes the simple mesh interconnection network. The simple mesh interconnection network can be pictured as graph $G = \{V, E\}$, where V is the set of vertices. The vertices of the graph represent the nodes in the mesh interconnection network and E describes the links of the mesh interconnection network. As the nodes of the mesh can be represented in a two-dimensional plane, it is addressed by the coordinates starting from *(0, 0)* to *(m-1, n-1)*. The node *(x, y)* is connected to other nodes; the set of such nodes has been labelled *N*. The set *N* contains the ordered pair of coordinates of the neighbour node to node with coordinates (x, y), which can be generated using the equation (2.1)

$$N = \begin{Bmatrix} (x, y+1) & y < n-1 \\ (x, y-1) & y > 0 \\ (x+1, y) & x < m-1 \\ (x-1, y) & x > 0 \end{Bmatrix} \tag{2.1}$$

Figure 2.1 describes the mesh interconnection of 4 × 4 nodes.

The Intel's Teraflops Research Chip uses 10 × 8 2D mesh having 80 cores [27], Tilera is the 8 × 8 64-node chip that is using 2D mesh [28], and the TRIPS processor [29] also applies the mesh topology.

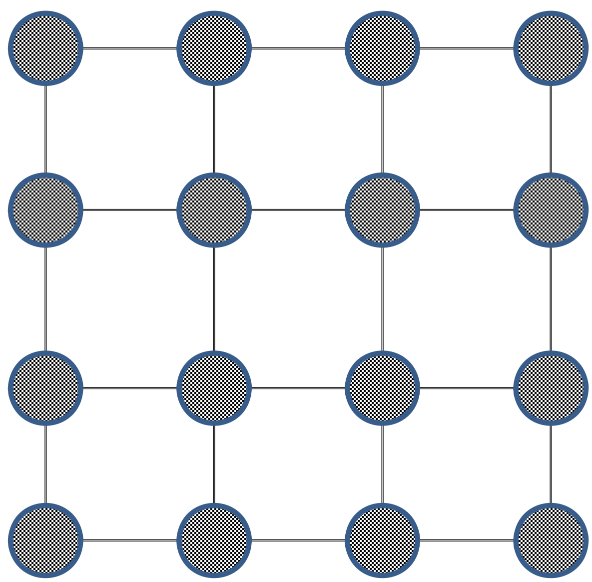

Figure 2.1: Simple Mesh Interconnection of 4 × 4 Mesh

2.2.2 Torus Mesh Interconnection Networks

Torus Mesh Interconnection Networks is a variant of the mesh interconnection network and has been used in supercomputers. The Torus topology is generated through the implementation of wrap around links to the simple mesh topology. The addition of toroidal links to the 2D mesh generates a uniform degree in the topology and reduces the diameter. Due to the addition of the toroidal links, the bisection width is doubled in comparison to the 2D mesh. And this results in reduced uniformity of the edge length of the links. Hence, the topology will consist of links that that are both long and short.

The mathematical equation considering that the topology as a graph *G ={V, E}*, then the set containing the ordered pairs of coordinates for the four neighbouring nodes *N* is given by the equation (2.2).

$$N = \begin{Bmatrix} (x, (y+1)\%n) \\ (x, (y-1)\%n) \\ ((x+1)\%m, y) \\ ((x-1)\%m, y) \end{Bmatrix} \qquad (2.2)$$

In the above stated equation, *m* is the number of nodes in *X* direction, and *n* is the number of nodes in *Y* direction. The nodes are numbered in the coordinate system from *(0, 0)* to *(m-1, n-1)* respectively, and Figure 2.2 describes the torus topology. The two or three-dimensional torus existsexists, but as per the equation (2.2) the different sizes has the possibility of existing in various dimensions. The machine like HPGS1280 uses rectangular Tori; the IBM Blue Gene uses 3-D torus topology [30], and other popular systems using torus are CrayXT3 and CrayXT4[31].

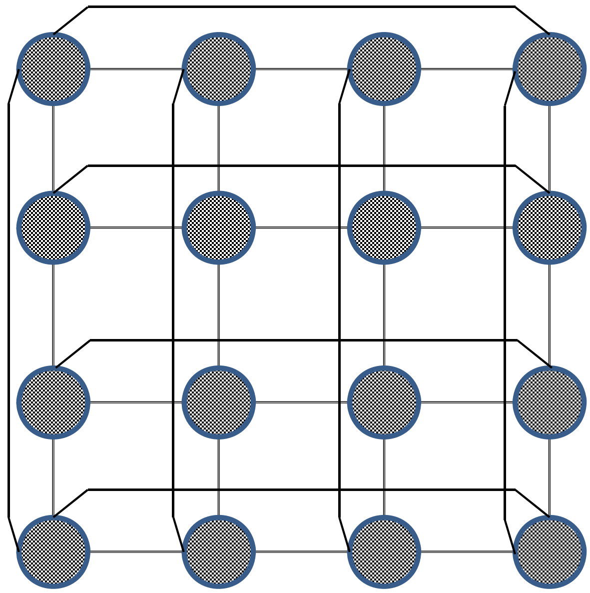

Figure 2.2: Simple Torus of 4 × 4

2.2.3 Twisted Torus Mesh Interconnection Networks

The twisted torus mesh have the most successful implementation in Illiac IV[30, 32]. The twisted torus can be two-dimensional and three-dimensional tori. Figure 2.3 describesthe twisted torus with a 4 × 4 node. The twisted toridial vertical link can be interpreted through the equation (2.3) given below:

$$(x^{'}, y^{'}) = \begin{Bmatrix} (x + m/2)\% m, n-1) & if\ y = 0 \\ (x - m/2)\% m, 0) & if\ y = n-1 \end{Bmatrix} \tag{2.3}$$

In the above equation, *m* is the number of nodes in *X* direction and *n* is the number of nodes in *Y* direction and only a single node coordinates are generated as the result.

The square twisted torus has rows and columns that are of equal size and, it was assessed that the twisted torus does not possess the potential to achieve a better performance; even though the symmetry of the torus network is disturbed, which is the valuable property while designing routing algorithms [33].

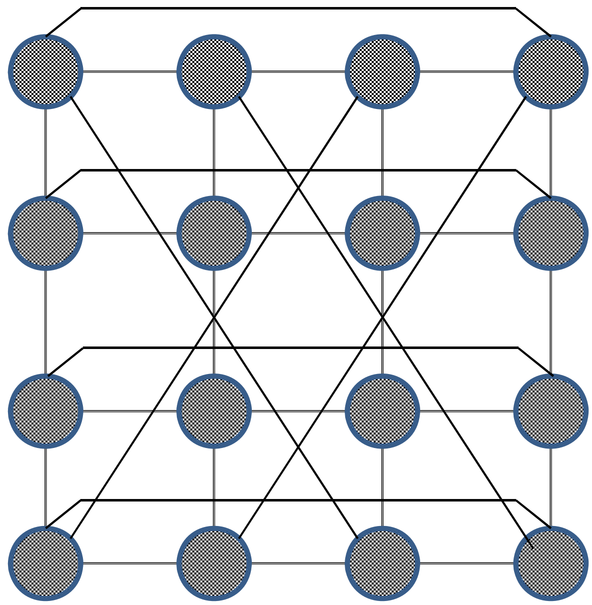

Figure 2.3: Twisted Torus of 4 × 4

2.2.4 Folded Torus

The torus network possesses a drawback in its successful implementation, that is, it does not possess a uniform edge length. The toroidal links are long in comparison to that of the inner links, which in turn hinders the efficient performance of the network. On the application of the toroidal links, the end to end latency will be high for the packet travelling the same number of hops without using toroidal links. This problem can be eliminated by arranging the nodes in a fashion, such that each node is at a consistent position. In the previously discussed torus topology, all the nodes were placed in the coordinate system according to the node number but in this approach, the nodes are connected as described in the equation (2.4) below

$$x'_i = \begin{Bmatrix} 2x_i & if\ x_i < \frac{n}{2} \\ 2k - 2xi - 1 & if\ xi \geq \frac{n}{2} \end{Bmatrix} \tag{2.4}$$

The above equation (2.4) describes the coordinates in the *X* dimension; as the topology having two dimensions with each having *n rows* and *n* columns of the node that are numbered from 0 to n − 1. The x_i describes the position of the node in the simple torus and x'_i describes the new physical space in the topology. The same equation is applicable in describing the *y* position of the node in the *XY* coordinate system. The other parameters, such as the bisection width of the topology remains constant[1]. Figure 2.4 describes the Folded Torus topology.

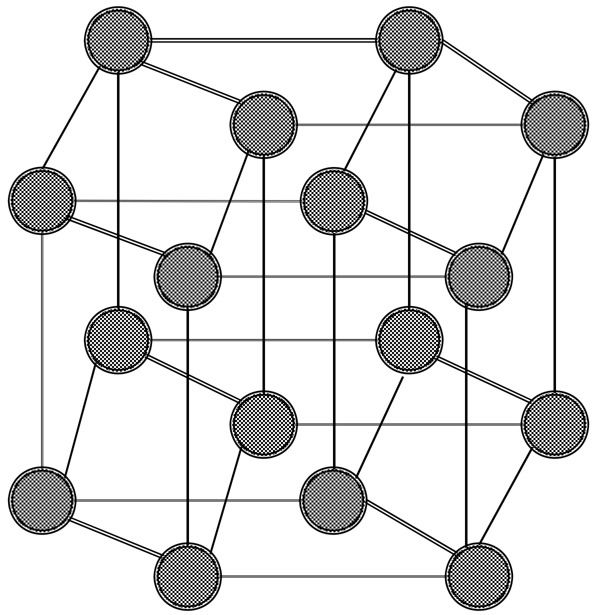

Figure 2.4: Folded Torus of 4 × 4

2.2.5 X-Mesh Interconnection Networks

X-Mesh has some of the links that connect horizontally, and another set of links that connects diagonally. The horizontal and diagonal links reflect a wrapped around nature, which makes the overall topology a torus. The diagonal links acts as connections to other nodes, based on the position of the node in the topology. If the sum of *X* and *Y* coordinates is even, then the authors [34] suggested to add links to the nodes on the north-east and south-west of the node. And, if the sum is odd then the node should be connected to the nodes on the north-west and south-east of the node. The mathematical formulation for the set *N* containing ordered pair of neighbour nodes and is calculated as *(f(x+1),y) , (f(x-1),y), (f(x-1),f(y-1)), (f(x+1),f(y+1))* when $x + y$ comes out to be even and when odd it will be connected to the nodes *(f(x+1),y) , (f(x-1),y), (f(x-1),f(y+1)), (f(x+1),f(y-1))*. The function *f(x)* is described by the equation 2.5.

$$f(x) = \begin{cases} x & if -\frac{k}{2} < x \leq \frac{k}{2} \\ x - k & if\ x > \frac{k}{2} \\ x + k & if\ x \leq \frac{-k}{2} \end{cases} \tag{2.5}$$

The above described equation describes the function that is used to calculate the x or y coordinates values, as the input to generate the coordinate of the neighbour node. As the authors [34] have assumed that the mesh possess the k nodes and node (0, 0) then it is the center node. The topology has a bisection width greater than that of the simple mesh network, but the degree of X-Mesh is constant for every node that is four. Still, the topology consists of the toroidal links which will be of a larger length and has the potential to affect the performance of the network[34]. Figure 2.5 shows the X-Mesh interconnection network.

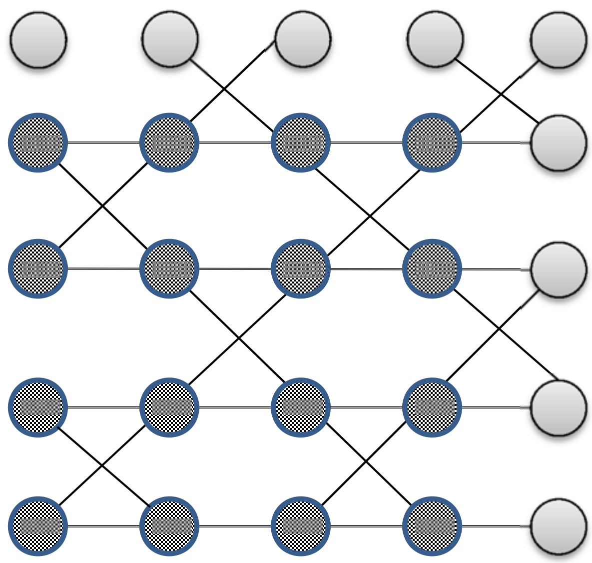

Figure 2.5: X-MESH 4 × 4

2.2.6 C^2 Mesh Interconnection Networks

C^2 Mesh is a square mesh, possessing n nodes in both the X and Y dimensions. C^2 Mesh has two variants for the two-dimensional $n \times n$ mesh. The first variant represents the odd number of nodes and another one for the even number of nodes. For the odd value of n, there is a direct connection between the center node to the corner edge node of the mesh. In the case of even number of nodes in rows and coloums, the four corner nodes are connected to the nearest center node. The nearest center is the node, which can be assumed as the center because the structure does not possess any common center node. The central idea behind the practice is, to reduce the diameter of the whole topology, and it has been reduced significantly. However, the issues that came into existence is the routing algorithm suggested by the authors[35] which states, always to reach the center node before going to the destination node. The logic of redirection to the center makes a huge congestion at the center nodes. The Neighbour nodes *N* of the topology are represented with equations (2.6) and (2.7) given below:

$$N = \left\{ \begin{array}{ll} (x, y+1) & \text{if } y < m \\ (x, y-1) & \text{if } y > 0 \\ (x+1, y) & \text{if } x < m \\ (x-1, y) & \text{if } x > 0 \\ \left(\frac{m}{2}, \frac{m}{2}\right) & \text{if } (x, y) = (0,0) or (0, m) or (m, 0) or (m, m) \\ \{(0,0), (0, m)(m, 0)(m, m)\} & \text{If } x = \frac{m}{2} \text{ and } y = m/2 \end{array} \right\} \quad (2.6)$$

$$N = \left\{ \begin{array}{ll} (x, y+1) & \text{if } y < m \\ (x, y-1) & \text{if } y > 0 \\ (x+1, y) & \text{if } x < m \\ (x-1, y) & \text{if } x > 0 \\ \left(\frac{n}{2}-1, \frac{n}{2}-1\right) & \text{if } x = 0 \text{ and } y = 0 \\ \left(\frac{n}{2}-1, \frac{n}{2}\right) & \text{if } x = 0 \text{ and } y = m \\ \left(\frac{n}{2}, \frac{n}{2}-1\right) & \text{if } x = m \text{ and } y = 0 \\ \left(\frac{n}{2}, \frac{n}{2}\right) & \text{if } x = m \text{ and } y = m \\ (0,0) & \text{if } x = \frac{n}{2}-1 \text{ and } y = \frac{n}{2}-1 \\ (0, m) & \text{if } x = \frac{n}{2}-1 \text{ and } y = \frac{n}{2} \\ (m, 0) & \text{if } x = \frac{n}{2} \text{ and } y = \frac{n}{2}-1 \\ (m, m) & \text{if } x = \frac{n}{2} \text{ and } y = \frac{n}{2} \end{array} \right\} \quad (2.7)$$

From the above two equations, it can be observed that the nodes are bidirectional links. For an even number of nodes, there are four centres nodes and the edge connects the corner node to the nearest center [35]. In the above equation for the topology of n × n, the value of m = n-1, as the nodes are numbered from (0, 0) to (n-1, n-1). The equation (2.6) is for odd number of nodes and equation 2.7 is for even number of nodes. The C^2 mesh has been described in Figure 2.6

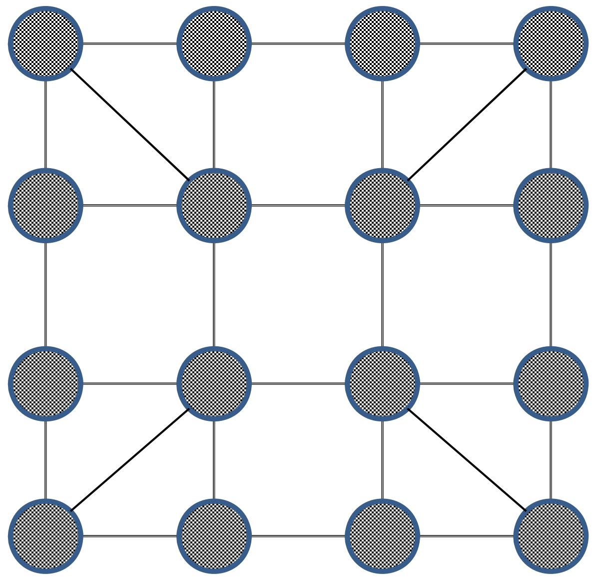

Figure 2.6: Center Concentrated Mesh

2.2.7 D-Mesh

D-Mesh suggests that going first in X or Y direction will be optimal to connect the nodes diagonally, as diagonal is always the shortest path. The diagonal links decreases the diameter of the topology and a reduction in diameter will increase the number of links in the topology. The growth in the number of links will also improve the bisection width of the topology and the degree of each node has increased to 8. The authors [36] have not developed any mathematical formulation for creating the topology, but for the better understanding, the mathematical representation of the links to the neighbour node N are given by the equation (2.8) and (2.9). Let *(x, y)* be the coordinates of any node in the mesh topology that has been generated from equation (2.1),), then the set N of additional links from the node will be connected to the nodes in the cross product of set X' and Y', given by the equation (2.8) and (2.9)

$$X' = \begin{Bmatrix} x+1 & x < m-1 \\ x-1 & x > 0 \end{Bmatrix} \tag{2.8}$$

$$Y' = \begin{Bmatrix} y+1 & y < n-1 \\ y-1 & y > 0 \end{Bmatrix} \tag{2.9}$$

Figure 2.7 describes the 4 × 4 D-Mesh topology.

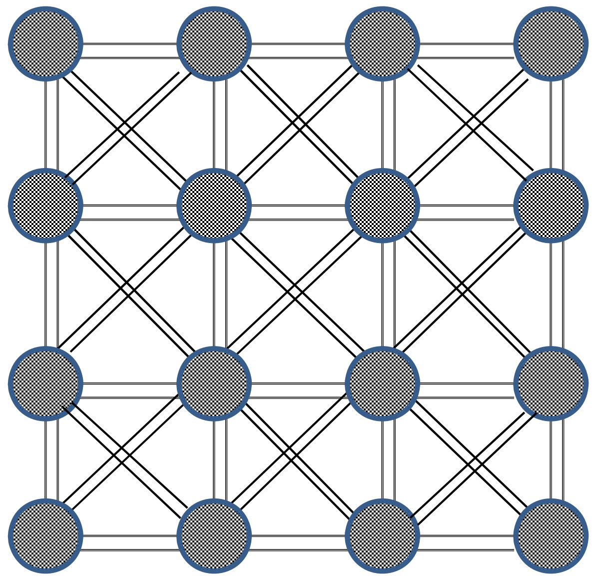

Figure 2.7: D-Mesh

2.2.8 Hybrid NOC Topology

The Hybrid NOC topology is a hybrid of the links that exist in three different topologies. The topology has evolved from the torus, folded torus and mesh topology and each of the links has specific objectives in the topology. The torus links reduces the diameter of the topology, folded torus links groups odd and even the nodes together, and the mesh links keep the adjacent nodes together. The degree of the topology is four. If the topology has n^2 nodes placed in n rows and n columns, the diameter of the hybrid NOC topology is equal to n [28]. The bisection width comes out to be 3n for the topology. The topology has been described below in Figure 2.8.

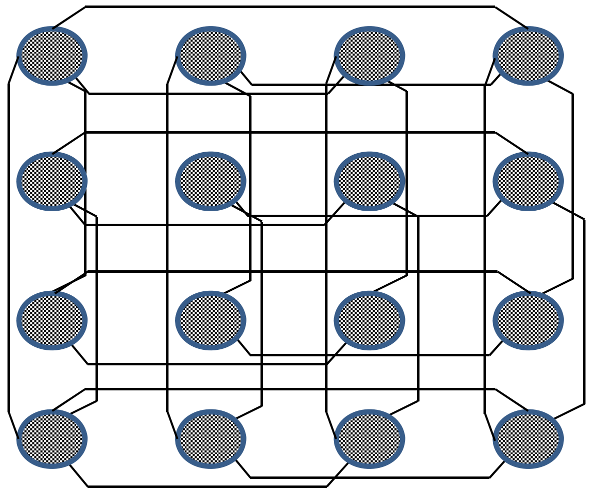

Figure 2.8: Hybrid NOC Topology

2.2.9 T-Mesh

T-mesh is designed by the addition of four extra-long links to the mesh topology, with the objective to reduce the diameter of the topology. The reduction of the diameter reduces the communication delay. The diameter of the T-mesh is *n* and *n-1* for odd and even number of nodes, respectively. Likewise, the degree of the topology is not visibly affected as the extra links lay at the corner nodes [37]. The bisection width of the topology has increased by two. And Figure 2.9 presents the topology. The four extra – links added to the mesh topology are generated by equation (2.1) with m number of nodes in X direction and n number of nodes in Y direction, and is interpreted by the equation (2.10)

$$N = \begin{cases} (0, n-1), (m-1, 0) & if\ x = 0\ and\ y = 0 \\ (m-1, n-1), (0,0) & if\ x = m-1\ and\ y = 0 \\ (m-1, n-1), (0,0) & if\ x = 0\ and\ y = n-1 \\ (0, n-1), (m-1, n-1) & if\ x = m-1\ and\ y = n-1 \end{cases} \tag{2.10}$$

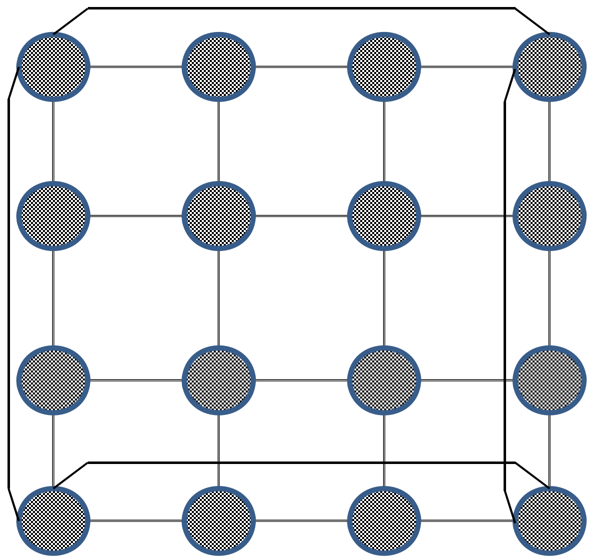

Figure 2.9: T-Mesh of 4 × 4

2.2.10 Diagonal Connected Mesh

The Diagonal Connected Mesh topology uses the mesh connected crossbars[31]. The central and prime advantage of implementing the mesh connected crossbar is the reduction of the wiring density. The mesh connected crossbar also has the advantage of providing a greater bandwidth as compared to the other topologies. The Figure 2.10 describes the Diagonal Connected Mesh (DCM). The maximum degree of the DCM router is six, and it implies the cost of the hardware and complexity of the hardware is more. The equation (2.11) describes the diameter of the DCM.

$$D = \max(m, n) - 1 \tag{2.11}$$

Here m, n is the number of nodes in the row and columns of the topology.

The bisection width of the topology is given by equation (2.12).

$$D = (n/2) * 2 + n \tag{2.12}$$

Assuming the topology is of $n \times n$ nodes and $n/2$ will return an integral part of the result.

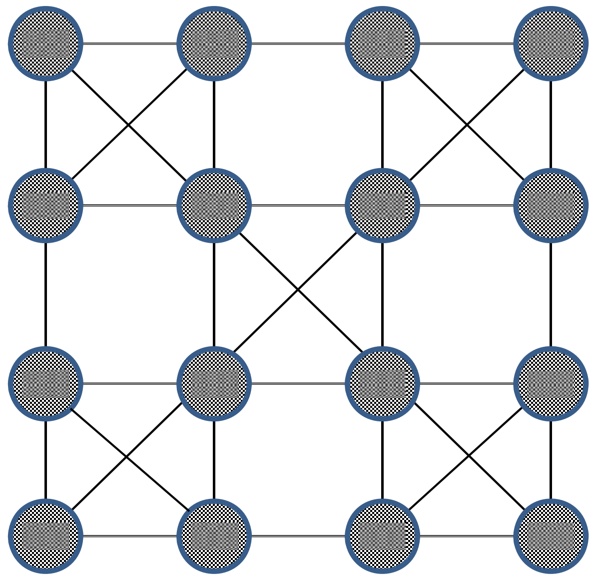

Figure 2.10: Diagonal Connected Mesh

2.2.11 Diagonal Mesh

Arden proposed the diagonal (toroidal) meshes. The topology is connected diagonally, instead of the usual horizontal and vertical connections. The Figure 2.11 describes the topology. The degree of the topology is constant for every node that has the value of four. For the even number of nodes, the diameter will be infinity and the bisection width will be zero. For an odd number of nodes, the bisection width is 4n[38].

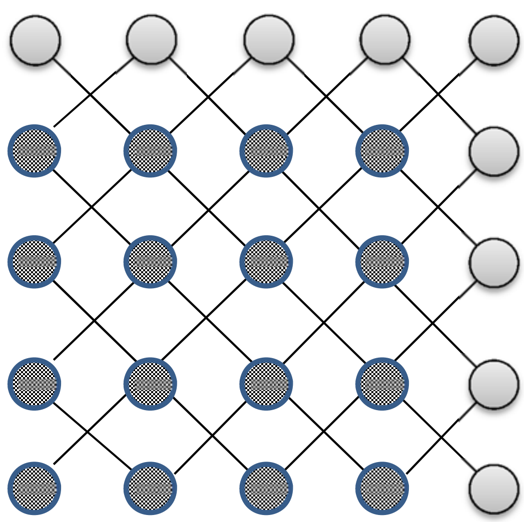

Figure 2.11: Diagonal Mesh

2.2.12 X-Torus

The Figure 2.12 describes the X-Torus topology and has the degree of 6. The bisection width of the topology is higher in comparison to that of mesh. But, the bisection width and the degree of the topology are different for an odd number of nodes and even number of nodes. The diameter of the topology is at the distance of $k/2$ and $k/2+1$ for even and odd value of k and the topology of the dimension $k \times k$. The bisection width of the topology is polynomial function of the order of k^2[39, 40].

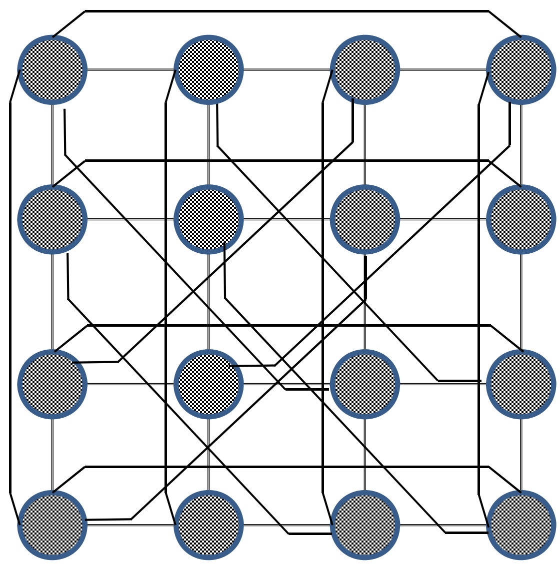

Figure 2.12: X Torus Topology 4 × 4

2.2.13 xtorus

The xtorus topology was designed to reduce the communication cost of the networks[41]. The diameter of the topology is n-1 for both the even and odd number of nodes and the degree of each node is five. The bisection width in comparison to the torus topology has increased by 2. The xtorus topology is described below in the Figure 2.13

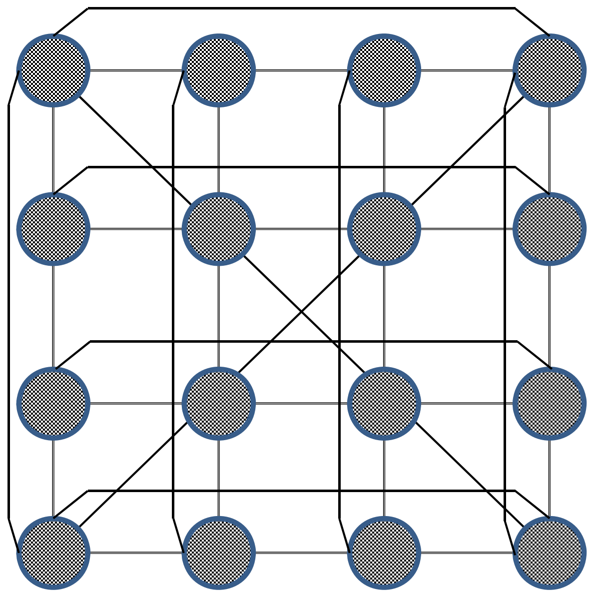

Figure 2.13: xtorus

2.2.14 xxtorus

The xxtorus topology is generated though the annexation of two diagonal links on the diagonal corner nodes. The degree of the current topology will further increase by one, as the nodes at the corner will require 6 edges. The diameter of the suggested topology is less in comparison to the xtorus and the bisection width of the topology will be further increased by two. Figure 2.14 describes the xxtorus topology.

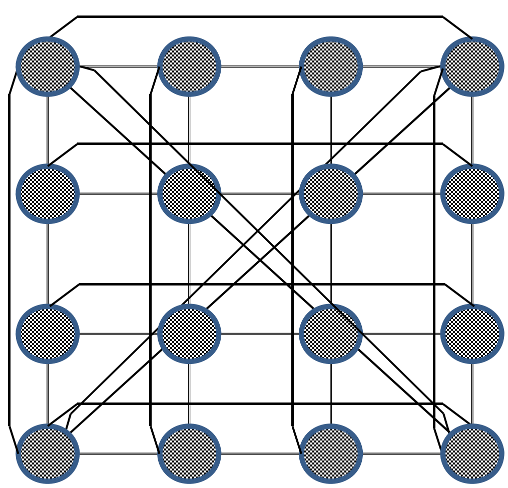

Figure 2.14: xxtorus 4×4

2.2.15 Dtorus

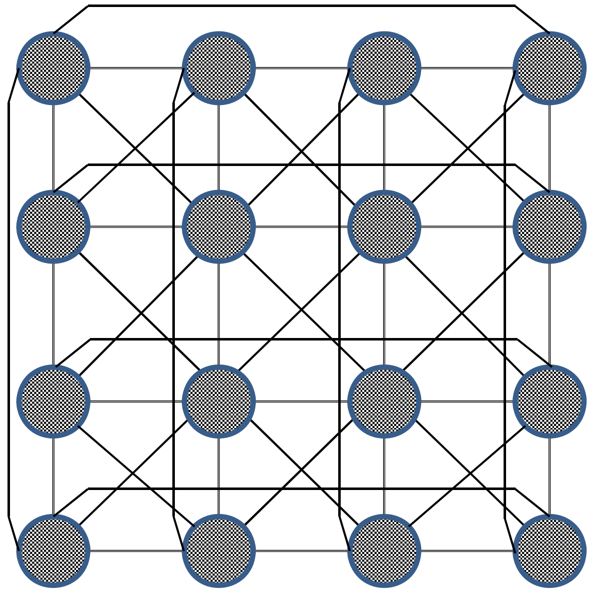

Figure 2.15: Dtorus topology of 4 × 4

The Dtorus topology is the variant of Dmesh topology, and has the extra horizontal and vertical toroidal links [42, 43]. The degree of the Dtorus topology is 8, and the diameter of the topology is *n-1* for $n \times n$ nodes. The bisection bandwidth of the topology is *4n-2*. The Figure 2.15 describes the 4×4 Dtorus topology.

2.2.16 Xmesh

The Xmesh topology can be designed by adding the two diagonal links to the diagonal corner nodes of the mesh topology[44]. The maximum degree of the topology is 4 and the diameter of the topology is *n-1*, and the bisection width of the topology is *n+4*. The Figure 2.16 describes 4×4 Xmesh topology.

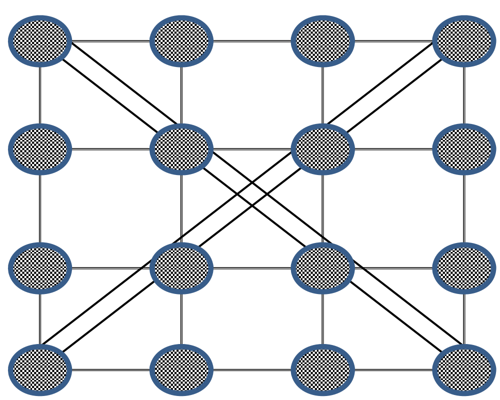

Figure 2.16: Xmesh of 4 × 4

2.2.17 SD-Torus

The SD-Torus topology is designed by adding the diagonal links along the minor diagonal of the topology. The topology is described below in the Figure 2.17. The degree of the topology is uniform for every node that has a value of 6. The equation (2.13) gives the diameter of the SD torus topology

$$D = \frac{2n}{3} \tag{2.13}$$

Moreover, the bisection width can be given as $B = 3n$

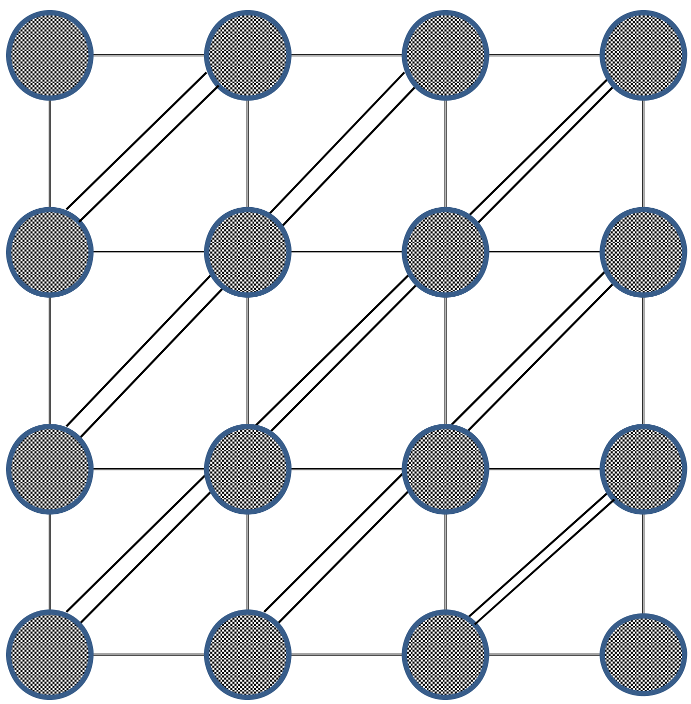

Figure 2.17: SD Torus Topology

2.2.18 CC torus

The C^2 mesh topology has partially inspired the CC torus topology. The CC torus topology connects the center node of the topology to the corner nodes and center nodes with nearest horizontal and vertical node of the boundary nodes [45]. Figure 2.18 describes the CC torus topology.

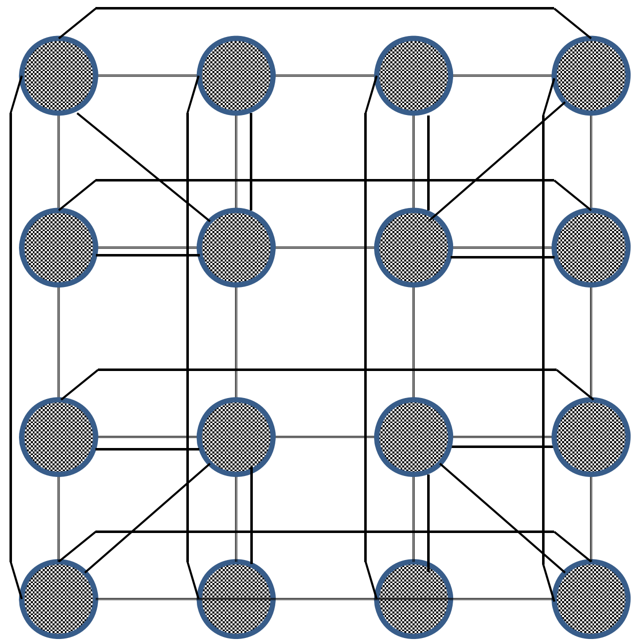

Figure 2.18: CC Torus Topology

For an odd number of nodes, the degree of the topology is 12 and 7 in the case of even number of nodes. The diameter of the topology is *n-1*. The bisection width of the topology, for the even number of the node is same as that of the torus topology. The bisection width for the odd number of nodes will require additional removal of 5 links in comparison of the toroidal links.

2.2.19 C - Mesh

The C-Mesh stands for the Concentrated mesh. The authors have suggested reducing the hop counts of the topology by connecting four processing units to each switch[46]. By connecting the four processing units, the hop count can be reduced. However, it will also increase the traffic on the link and router. The bisection width of the network is also less in comparison to mesh, so the topology is suitable for applications that have a low traffic requirement and is less reliable and fault tolerant. The Figure 2.19 describes the C-Mesh.

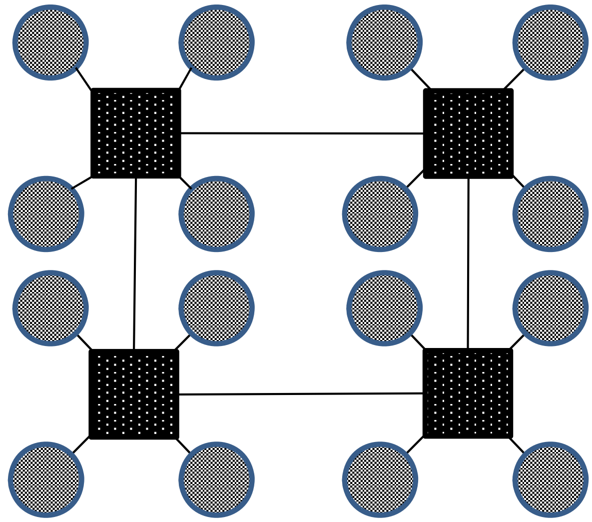

Figure 2.19: C-Mesh of 16 nodes

2.2.20 Hexagonal Mesh

In the hexagonal mesh, the links are similar to the SD torus, except that the toroidal links are missing from the topology. The maximum degree of the nodes in the topology is 6. And for the hexagonal mesh with n nodes arranged in rows and columns, the diameter of the topology can be expressed as $2(n-1)^{0.5}$. The Figure 2.20 describes the Hexagonal Mesh[47].

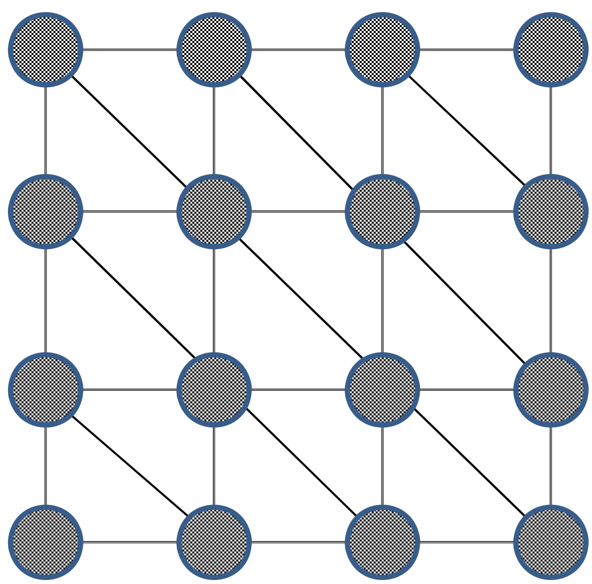

Figure 2.20: Hexagonal Mesh

2.2.21 NR Mesh

The NR Mesh is a fault tolerant topology and each of its processing elements has four routing elements. In case of a router failure in the NR Mesh, the packet has the opportunity to reach the destination using the alternate path. As the fault tolerance is dependent on the bisection width, the bisection width of the topology of $n \times n$ is *3n-1*. The degree of the router comes out to be eight, but the degree of the processing element is now four in comparison to one, so this incurs an extra cost for the design. The diameter of the topology is *2n-4* when reporting if considering mesh having the diameter of *2n* [48]. The Figure 2.21 describes the NR Mesh.

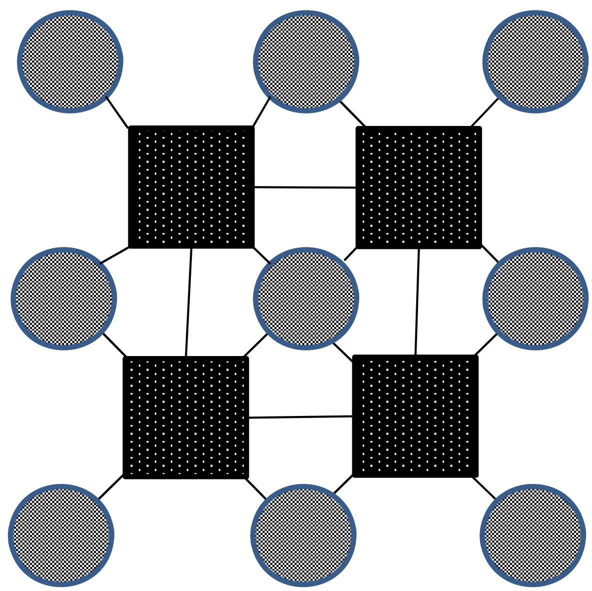

Figure 2.21: NR Mesh

2.2.22 Dual Connected Mesh Structure (DCS)

The DCS topology uses the two links, that is, master link and a slave link to the processing element. DCS supports the fault tolerance at the switch level. Another synonym for DCS is NR/2 Mesh [48]. The degree of the router is six, but the degree of the core port is two which was earlier considered to be 4 in the case of NR Mesh[49]. The bisection width of the topology is 2n+1 for n^2 nodes. The diameters of the topology are one hop less than that of the mesh topology. The Figure 2.22 describes the DCS topology.

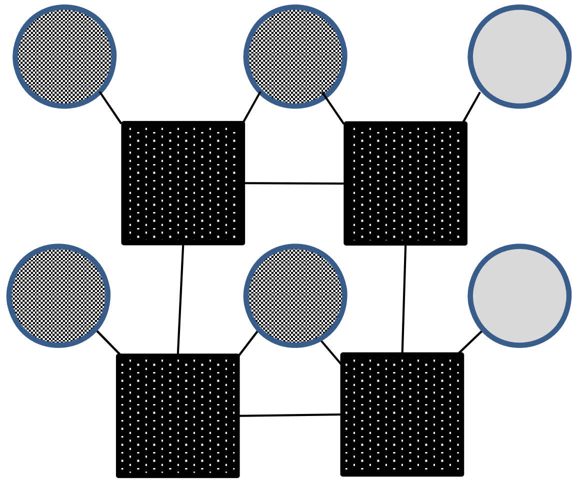

Figure 2.22: Dual Connected Mesh Structure

2.2.23 Honeycomb Mesh

Honeycomb mesh topology is encompassed of a fixed degree and the edge length of the topology is constant. The topology is comparable to the most popular topologies like a hypercube. The most important property of this topology is the planar nature; the diameter of the topology comes out to be $1.63n$ for n^2 nodes. The degree of the nodes is three. The bisection width is $0.82n$ *for* n^2 nodes. The Figure 2.23 represents the Honeycomb mesh[50].

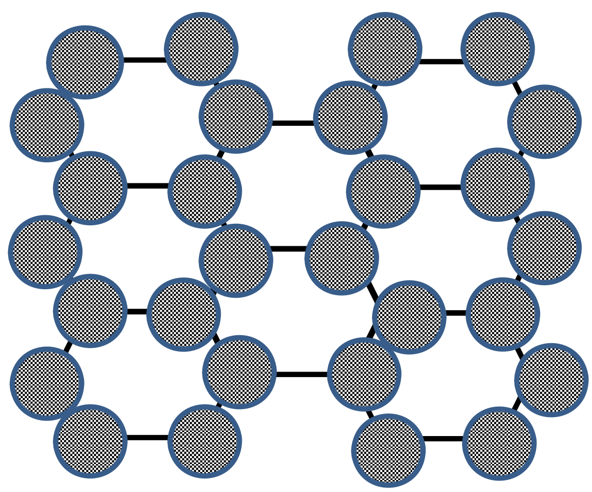

Figure 2.23: Honeycomb Mesh

2.2.24 Honeycomb Torus

Honeycomb torus is the variant of the honeycomb mesh. Honeycomb torus has three types of toroidal links, to which the degree is three. The diameter of the topology is 0.81n for the topology n^2 nodes. The bisection width of the topology is double times more than that of the honeycomb mesh. The bisection width is 2.04n when the topology is assumed to have n^2 nodes[50]. The edges length of the topology is not uniform due to the existence of the toroidal links. The Figure 2.24 describes the topology.

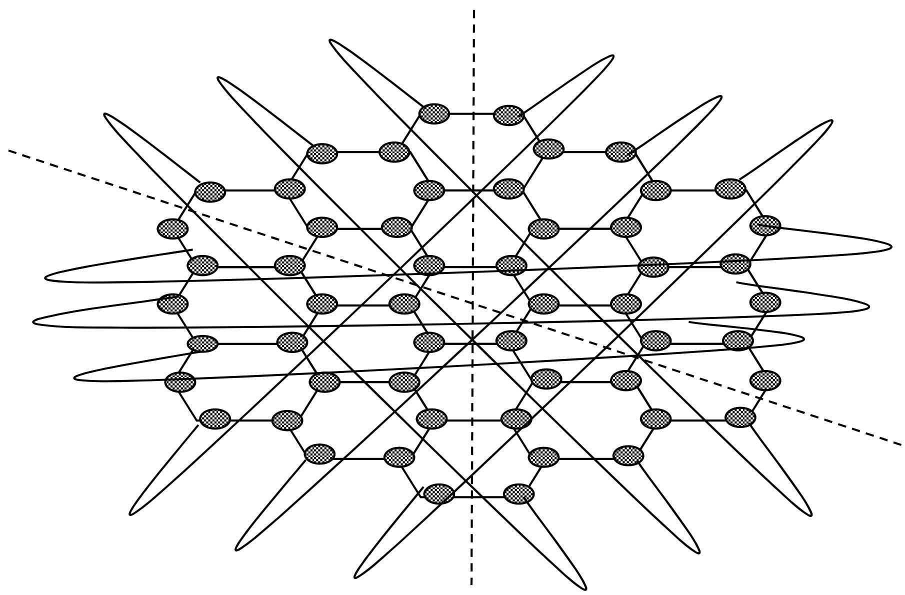

Figure 2.24: Honeycomb Torus

2.2.25 Honeycomb Rhombic Mesh

The honeycomb rhombic mesh is also another kind of the organisation of the honeycomb mesh, where the degree of the rhombic honeycomb is three. The diameter for n^2 nodes is 2.83n. The bisection width of the suggested topology is comparatively less in comparison to that of the honeycomb mesh that is 0.71n [50]. The Figure 2.25 gives the pictorial representation of the topology.

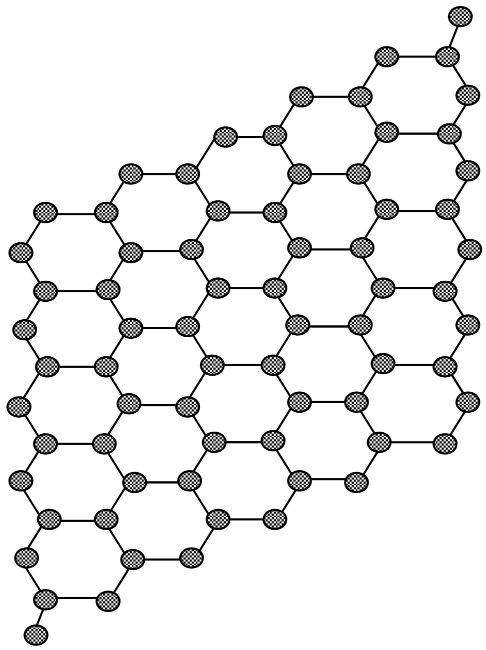

Figure 2.25: Honeycomb Rhombic Mesh

2.2.26 Honeycomb Square Mesh

The Honeycomb Square Mesh (HSM) is the organisation of hexagons, which results in the development of a topology in the form of the square. Like other honeycombs, the HSM also possess a degree of 3. The diameter of the honeycomb square mesh for n^2 nodes it is equal to 2n. The bisection width is 0.5n[50]. The Figure 2.26 shows the organisation of nodes in Honeycomb Square Mesh topology.

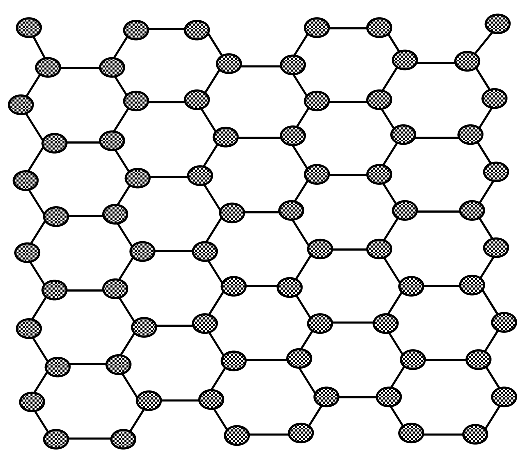

Figure 2.26: Honeycomb Square Mesh

2.2.27 Honeycomb Rhombic Torus

The Honeycomb Rhombic Torus has been the derived from the honeycomb rhombic mesh. The degree of the rhombic torus is three. The diameter of the honeycomb rhombic torus is 1.06*n*, and the bisection width of the honeycomb rhombic torus is 1.41*n* [50]. The edge length of the topology is not uniform due to the existence of the toroidal links. The Figure 2.27 represents the Honeycomb Rhombic Torus topology.

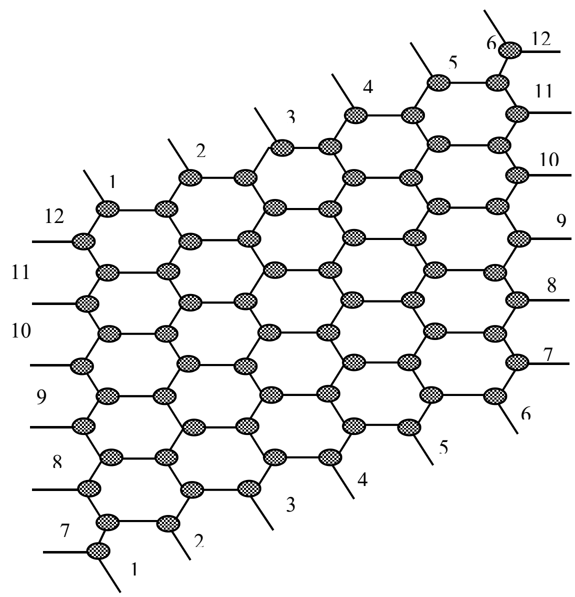

Figure 2.27: Honeycomb Rhombic Torus

2.2.28 Honeycomb Square Torus

The Honeycomb Square Torus has also been the derived from the honeycomb square mesh. The degree of the honeycomb square torus is three. The diameter of the honeycomb square torus is 2n, and the bisection width of the honeycomb rhombic torus is 0.5n [50]. The edge length of the topology is not uniform due to the existence of the toroidal links. The Figure 2.28 presents the Honeycomb Square Torus topology.

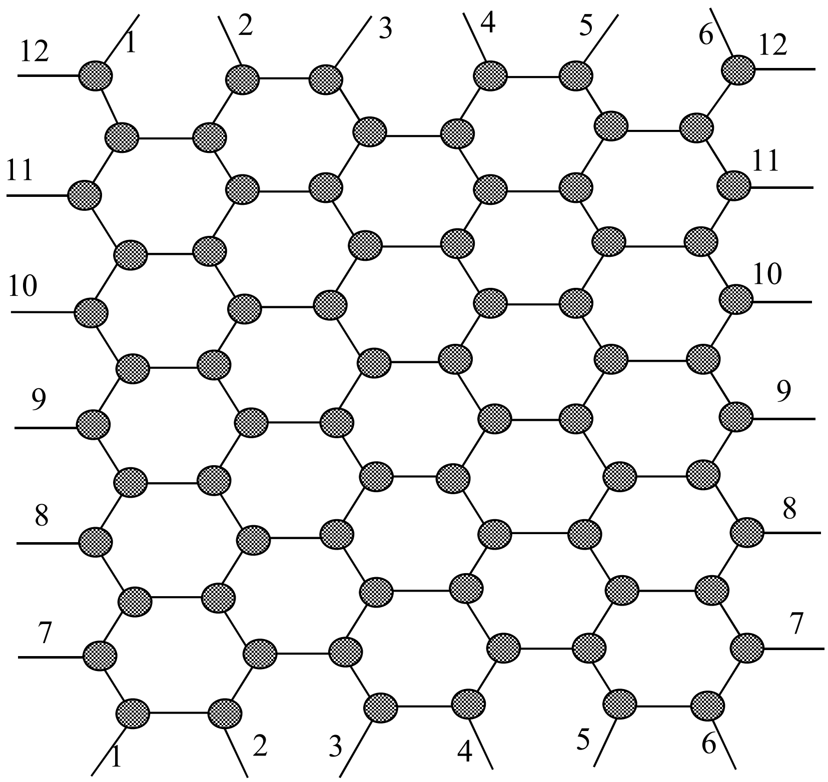

Figure 2.28: Honeycomb Square Torus

2.2.29 Structural Diametrical 2D Mesh Topology

The proposed topology has been designed and developed using a mesh consisting of even number of rows and columns. The mesh is further divided into the 2×2 mesh and the extreme corners are connected to the other extreme corners of the mesh. The degree of the topology is 7, and the diameter is equal to the maximum of the number of rows or columns. The bisection width of the topology is twice the minimum of the number of rows and columns[51]. The Figure 2.29 represents the topology.

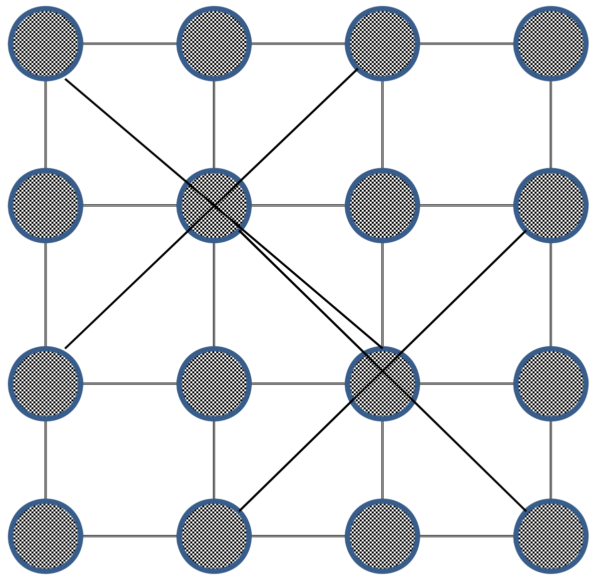

Figure 2.29: Structural Diametrical 2D Mesh Topology

2.2.30 Diametrical 2D Mesh Topology

The authors have suggested the Diametrical 2D Mesh Topology and it uses the diametrical links [52]. The topology was designed to reduce the diameter of the topology as well as the energy consumption of the topology. The degree of the topology is 7. However, the bisection width of the topology has been increased by 8. The diameter will be equal to n, where n is the maximum number of nodes in the *X* or *Y* dimension. The topology has been described in the Figure 2.30 below.

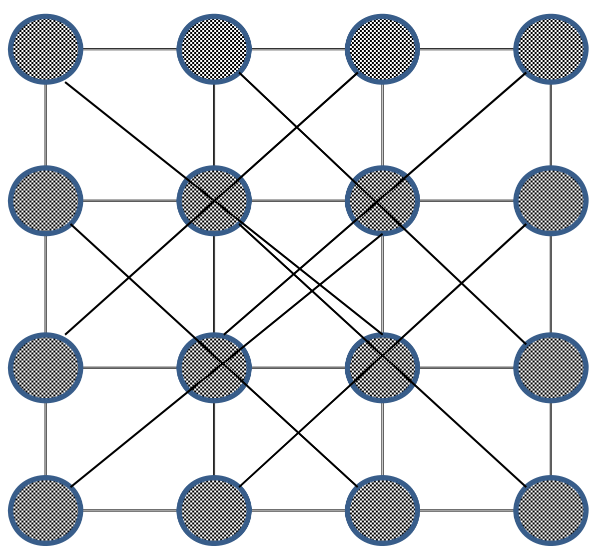

Figure 2.30: Diametrical 2D Mesh Topology

2.2.31 Double Loop (2m) Networks

The Double Loop (2m) Networks topology consists of two rings, each having 2*m* links. The rings represent the loop that the topology derived its name from (DL(2m) Network). The degree of each node will be three and the bisection width of the topology is four. The diameter of the topology is $m+1$[53]. The Figure 2.31 represents the DL (2m) with m having value as 16.

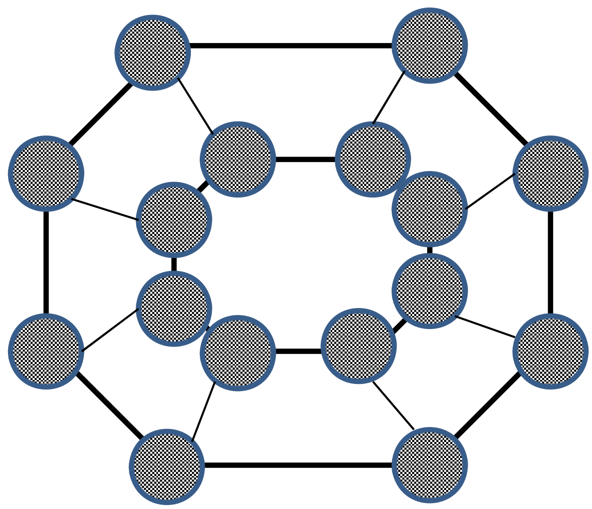

Figure 2.31: DL (2m) Networks

2.2.32 Shortly Connected Mesh

Under Shortly Connected Mesh, the idea is to opt for the shortest path while routing as it will result in acquiring links that are at the shortest distance from the node. The routing nodes in the topology have a special class of nodes called the bridge nodes. The bridge nodes exploit the diagonal properties of the mesh. The resultant topology is reflected in the Figure 2.32. Further, another particular type of nodes referred as the intermediate nodes with diagonal links are added to the topology to improve the performance. The degree of the topology will be equal to 8, and with the local port, 9. The diameter of the topology will be similar to that of the topology will be $n-1$. The bisection width of the topology is $n-1$[54].

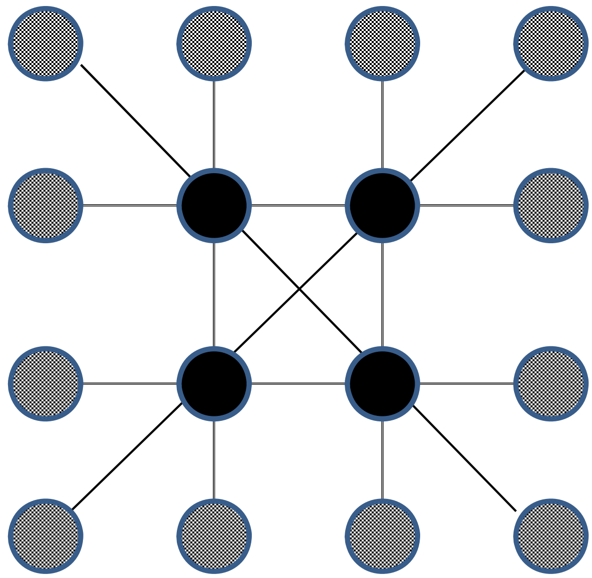

Figure 2.32: Shortly Connected Mesh

2.2.33 Cross Bypass Mesh

The cross bypass mesh is the topology developed on the idea of C^2 mesh and 2DDgl mesh network. The 3×3 mesh with diagonal links is the building block for designing the topology. The Cross Bypass Mesh topology is spread to develop topologies in large dimensions. The degree of the topology is nine and the diameter of the topology is *n-1*. The bisection width of the topology is increased to 2*n* for the even topology and 2*n+1* for the odd topology [55]. The Figure 2.33 describes the actual topology.

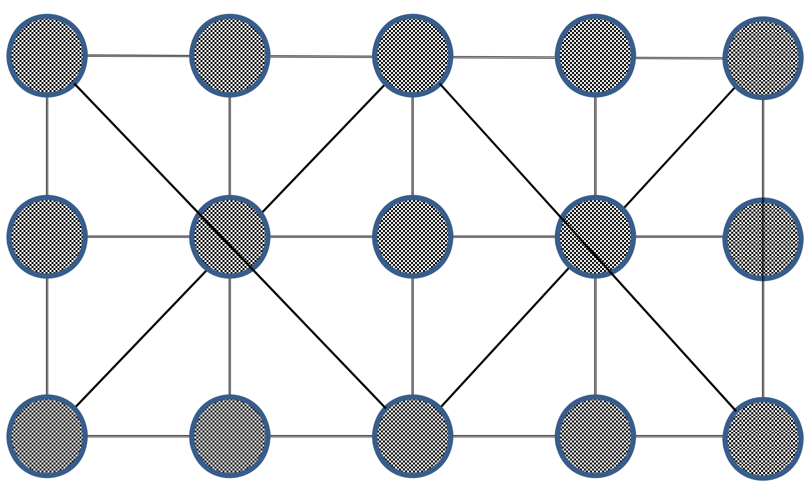

Figure 2.33: Cross Bypass Mesh

2.2.34 Cross Bypass Torus

The Cross bypass torus is a further modification of the cross bypass mesh topology and the degree of the topology is identical to that of the cross bypass mesh topology. The diameter for n ×n nodes arranged in n rows and n columns is (3n-2)/4. For even value of n, the bisection width of the n × n cross bypass torus topology is 3n and 3n+2 for odd values of n [56]. These extra links aid in enhancing the throughput of the network. The Figure 2.34 shows the layout of Cross Bypass Torus of 6 × 3.

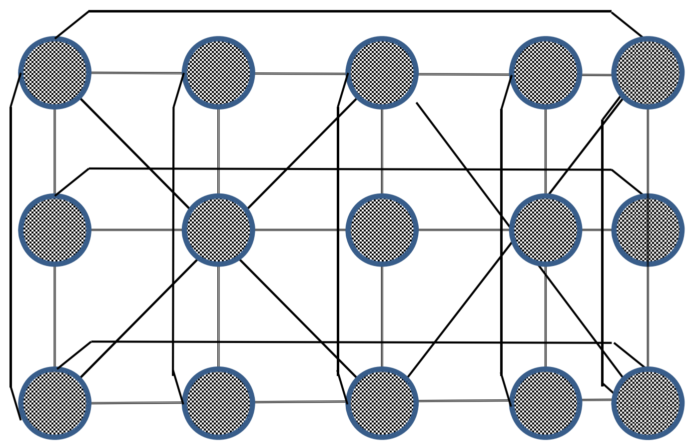

Figure 2.34: Cross Bypass Torus

The detailed comparisons of the assorted variants of the mesh topology are compared and the summarised report is shown in Table 2.1. While comparing these topologies, a fixed number of nodes are taken into consideration. The summarized table compares the topology on four significant topological parameters such as; degree, diameter, bisection width and the edge length. On carrying out the study it has been identified that, a specified number of variants of the mesh topology possess a disparate degree for core; so as to enable fault tolerance. The same is reported in the table.

Table 2.1: Comparison of the various variants of Mesh topologies

S.no.	Topology	Dimension	Degree (Core)	Degree(router)	Diameter	Bisection Width	Edge length
1	Two Dimensional Mesh	N×N	1	4	2N-2	N	Constant
2	Simple Torus	N×N	1	4	2*floor(N/2)	2N	Variable
3	Twisted Torus	N×N	1	4	N-1	2N	Variable
4	Folded Torus	N×N	1	4	N	2N	Constant
5	X MESH	N×N	1	6	2N	2N	Variable
6	C2 MESH	N×N	1	5,8	N	N	Variable
7	D-MESH	N×N	1	16	8N	8N	Variable
8	Hybrid NOC	N×N	1	4	N	3N	Variable
9	T MESH	N×N	1	4	N(ODD), N-1(EVEN)	N+2	Variable
10	Diagonal Connected Mesh	N×N	1	4	N-1	N/2*2+N	Variable
11	Diagonal Mesh	N×N	1	4	INF(EVEN)	(EVEN)0, 4N(ODD)	Constant
12	X TORUS	N×N	1	6	N/2,N/2+1	N*N	Variable
13	Xtorus	N×N	1	6	N-1	2N+2	Variable
14	Xxtorus	N×N	1	6	< N-1	2N+4	Variable
15	D Torus	N×N	1	8	N-1	4N-2	Variable
16	XMESH	N×N	1	4	N-1	N+4	Variable
17	SD Torus	N×N	1	6	2N/3	3N	Variable
18	CC Torus	N×N	1	12,7	N-1	2N+5	Variable
19	C Mesh	N×N	1	8	(N/4)-1	(N/4)	Constant
20	Hexagonal Mesh	N×N	1	6	$2(N-1)^{0.5}$.	2N-1	Variable
21	NR MESH	N×N	4	8	2N-4	3N-1	Constant

22	Dual Connected Mesh	N×N	2	6	2N-3	2N+1	Constant
23	Honey comb Mesh	N×N	1	3	1.63N	2.31N	Constant
24	Hexagonal Torus	N×N	1	6	0.58N	4.61N	Variable
25	Honey comb Torus	N×N	1	3	0.81N	2.04N	Variable
26	Honey Comb Rhombic Mesh	N×N	1	3	2.83N	0.71N	Constant
27	Honey Comb Square mesh	N×N	1	3	2N	0.5N	Variable
28	Honey Comb Rhombic Torus	N×N	1	3	1.06N	1.41N	Constant
29	Honey Comb Square Torus	N×N	1	3	N	N	Variable
30	Structural Diametrical 2D Mesh Topology	N×N (EVEN)	1	7	N	2N	Variable
31	Double Loop(2m) network	2×M	1	3	M+1	4	Variable
32	Shortly connected mesh	N×N	1	8	N-1	N-1	Variable
33	Cross Bypass Mesh	N×N	1	9	N-1	2N(Odd) 2N+1(Even)	Variable
34	Cross Bypass Torus	N×N	1	9	(3N-2)/4	3N(Odd) 3N+2(Even)	Variable

2.3 The Various Types of Traffics Used to Analyse the Performance of the Topologies

To test the performance of the topologies, various traffic patterns are used. The different types of traffic patterns recognised in the study are defined as follows:

2.3.1 Uniform Traffic

In uniform traffic, each source is equally likely to send to each destination. The uniform traffic is the most commonly used traffic pattern in network evaluation. The uniform traffic is very benign because, it balances load even for topologies and routing algorithms that normally have very poor load balance.[1]. The uniform traffic is independent of distance between the nodes and the main advantage of uniform traffic is that it can be traced analytically.

2.3.2 Bit Complement Traffic

The bit complement is the complement of the source address. The number generated will behave as the destination address[1]. This traffic belongs to family of permutation traffic and the main objective of bit compliment traffic is to stress the load balance of the topology.

2.3.3 Neighbour Traffic

In neighbour traffic, the destination is always adjacent to the source. Moreover, this can be either in horizontal, vertical or in diagonals of the node. Neighbour traffic is practised by assuming the source coordinate as $S(x, y)$ which is incremented or decremented either in the X or *in* Y direction for horizontal and vertical neighbours. For diagonal neighbours, it is incremented in both X and Y direction[1]. The neighbour traffic gives the impact of communication locality on the performance of the interconnection network and in the case of neighbour traffic radius, neighbourhood is fixed to one[57].

2.3.4 Tornado Traffic

Tornado traffic and its destination is set for the packet by incrementing and decrementing the source coordinates $S(x,y)$ by $n/2$ number of nodes either in X or Y coordinates or in both[1]. It also studies the impact of communication locality on the performance of the interconnection network but the radius of neighbourhood in case of tornado traffic is half of the diameter of the topology.

2.3.5 Bit Reversal Traffic

In bit reversal, the bits of the source address are swapped so that the most significant bit reaches the least significant bit and the second most significant bit is swapped with second most significant bit[1]. It belongs to family of permutation traffic and the main objective of this type of traffic is to stress the load balance of the topology[57].

2.3.6 Bit Transpose Traffic

In case of bit transpose traffic, bits of the source address are left rotated n/2 times in a circular manner[1]. It belongs to family of permutation traffic and the primary objective of this type of traffic is to stress the load balance of the topology[57].

2.3.7 Hot Spot Traffic

In hot spot traffic, a fixed number of nodes sends the data to a specific node by creating a hot spot in the topology[1]. The main objective of this traffic is to create the hotspot in the topology, so as to study the behaviour of the topology with hotspots[58]

CHAPTER 3
A MODIFIED DIAGONAL MESH INTERCONNECTION NETWORK AND ITS VARIANTS

3.1 Introduction

In the era of Nanotechnology, scientists were successful in deploying a large number of processing elements on a single chip. However, an enormous amount of communication that needs is to be served by the network residing on the chip for various processing elements present on chip. The units that participate in the communication of various cores are referred as Network on Chip (NOC)[31]. This leads to the development of mesh topology. In addition, the popularity of the mesh topology lies in the simplicity of the mesh network. The other factors that influence the popularity of mesh architecture are given as efficient layout and addressing scheme[59]. The popularity of mesh interconnection network was found to attract diverse research community, which led to the development of various types of mesh topologies such as X mesh, D mesh, T-mesh, C^2 mesh [60–66] and torus-like X torus, SD torus, xx torus [39, 42, 67–69]. Majority of these developed mesh variants were found to reduce the communication delay by decreasing the hop count and primarily focuses on the first three topological properties [1, 70]. The length of the links is overlooked in most of the topologies. Also, a diagonal toroidal mesh topology have been proposed in the past [38]. The topology highlighted an excellent performance and bisection width. Nevertheless, the major issues existed in the diagonal toroidal mesh topologies were the toroidal links were found to be present in the topology. The toroidal links are avoided in the various topologies, as they are the main source that creates the question of scalability. The study in the chapter has introduced four topologies. The three topologies comprising of diagonal toroidal mesh topology have been implemented, and their performance is studied based on the various traffic patterns. The fourth topology developed was inspired by the two topologies Diagonal Connected Mesh and T-Mesh.

3.2 Modified Diagonal Mesh Interconnection Network

A Modified Diagonal Mesh interconnection network (MDMIN) is developed by modifying Arden's Diagonal(Toroidal) Mesh interconnection network [38]. This approach was proposed to overcome the problems such as disconnection of Diagonal Mesh for even number of nodes in rows and columns. Besides, this approach is used to reduce the lengthy toroidal horizontal and vertical links. Figure 3.1 describes the MDMIN topology.

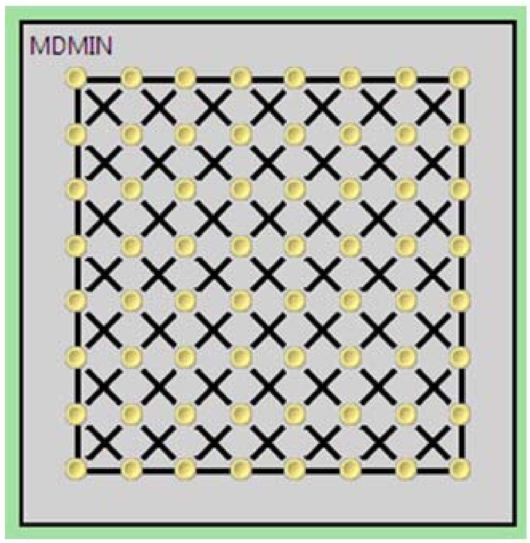

Figure 3.1: Modified Diagonal Mesh Interconnection Networks

The connection in the modified diagonal mesh interconnection network is described by the equations (3.1) to equation (3.3). If (x, y) be the coordinates of the node in the topology of m × n, the X′ represent the X coordinates of neighbour nodes, and are calculated using equation (3.1)

$$X' = \left\{ \begin{array}{ll} (x+1),(x-1) & 0 < x < m-1 \\ (x+1), x & x = 0 \\ (x-1), x & x = m-1 \end{array} \right\} \tag{3.1}$$

Similarly, the Y′ represents the Y coordinates of the neighbour node and are represented by equation (3.2)

$$Y' = \left\{ \begin{array}{ll} (y+1),(y-1) & 0 < y < n-1 \\ (y+1), y & y = 0 \\ (y-1), y & y = n-1 \end{array} \right\} \tag{3.2}$$

The overall coordinates of the neighbour nodes are the given by equation (3.3) where $X' \times Y'$ represents the Cartesian product of the two set and "–" represents the set difference operation.

$$E = X' \times Y' - (x, y) \tag{3.3}$$

The Table 3.1 compares the MDMIN with other topologies on the diameter and average path length of the topologies.

Table 3.1: Diameters and average path length of various topologies

Number of nodes N	**Diameter**				**Average Path Length**			
	Mesh	**Torus**	**Diagonal Mesh**	**MDMIN**	**Mesh**	**Torus**	**Diagonal Mesh**	**MDMIN**
4	2	2	Infinite	1	1.33	1.33	Infinite	1.00
9	4	2	4	2	2.00	1.50	1.93	1.56
16	6	4	Infinite	3	2.67	2.13	Infinite	2.10
25	8	4	8	4	3.33	2.50	3.16	2.64
36	10	6	Infinite	5	4.00	3.09	Infinite	3.18
49	12	6	12	6	4.67	3.50	4.40	3.71
64	14	8	Infinite	7	5.33	4.06	Infinite	4.25
81	16	8	16	8	6.00	4.50	5.64	4.79
100	18	10	Infinite	9	6.67	5.05	Infinite	5.32

Figure 3.2 represents the graph that compares the diameter of various topologies and shows that diameter of MDMIN is always less than the other three topologies. Figure 3.3 describe the average path length of the different topologies and it can be observed that the average path length of MDMIN is less than that of simple mesh and diagonal mesh, but average internode distance is found to be greater than torus topology.

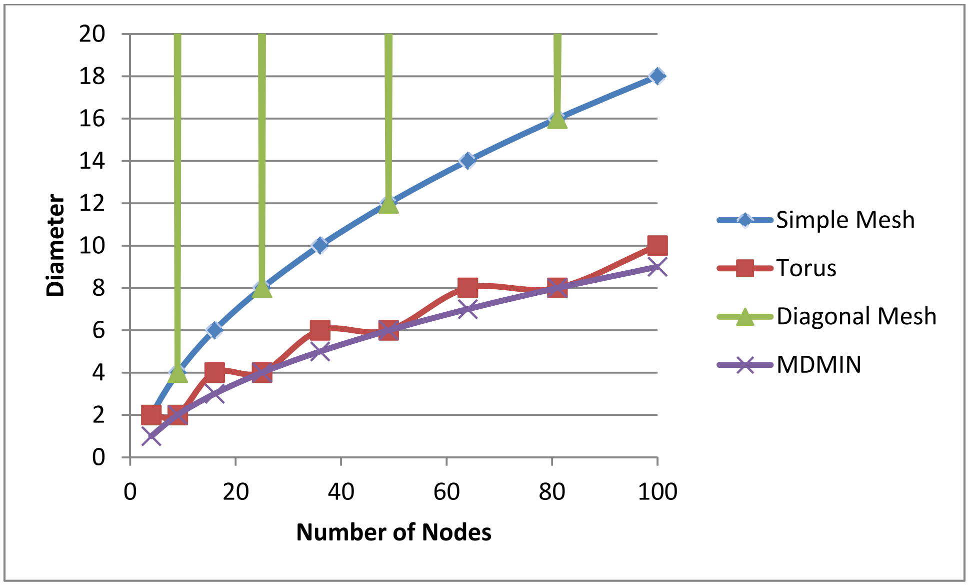

Figure 3.2: Comparison of Diameters of the Different Topologies with MDMIN

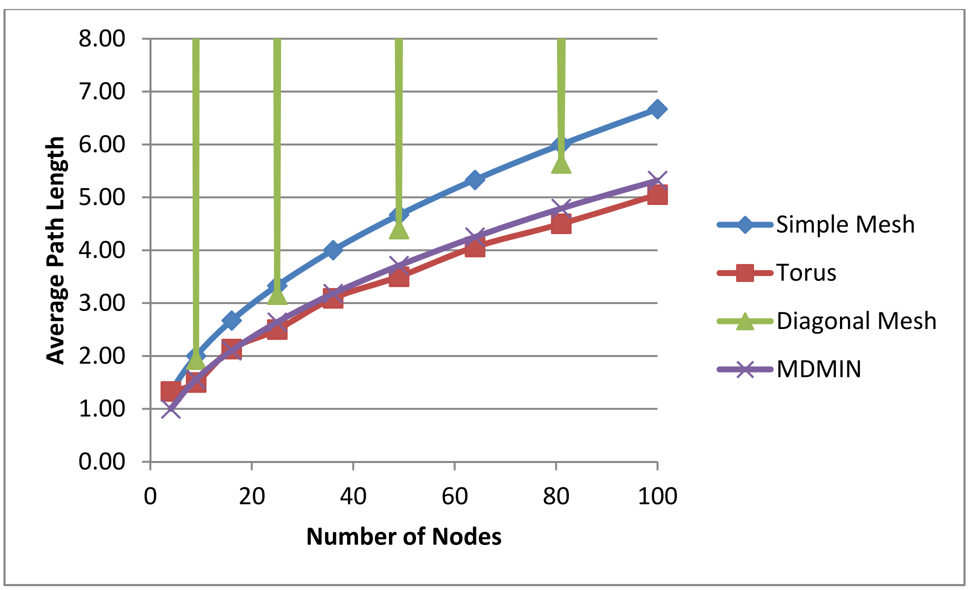

Figure 3.3: Comparison of Average Path Length of the Different Topologies with MDMIN

For the study of four topologies simple mesh, torus, diagonal mesh and MDMIN are simulated using NS2 with 5 rows and 5 columns. The CBR random traffic with the User Datagram Protocol is used to test the four topologies. The rate of the traffic was 2Mbps. The network has duplex links each having the channel capacity of 100Mb and the delay of 5ms. The performance parameters for the analysis are given by throughput and latency.

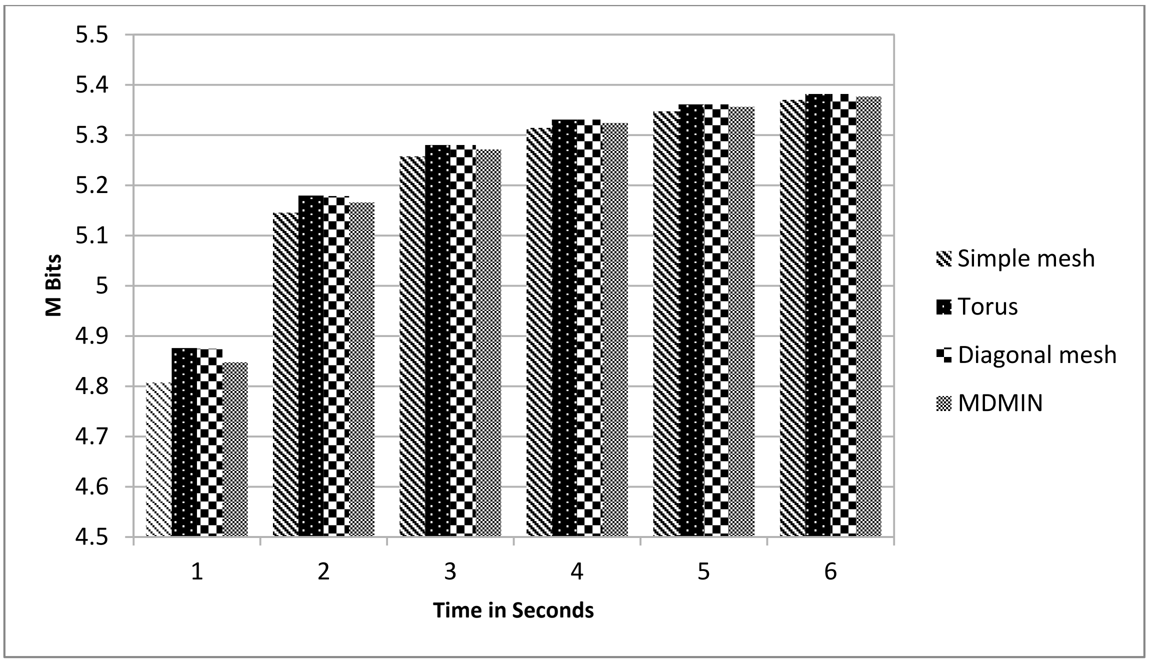

Figure 3.4: Comparison of Bits Transferred by Various Topologies with MDMIN after different time intervals

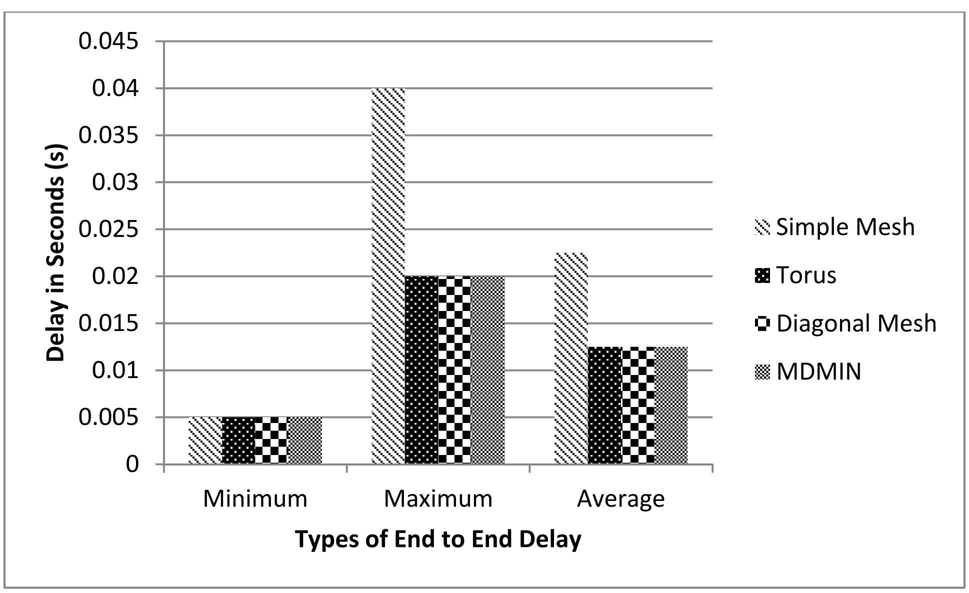

Figure 3.5: Comparison of End to End Latency of Different Topologies with MDMIN

Figure 3.4 shows that the data transferred of MDMIN is slightly less than diagonal mesh and torus topology. Figure 3.5 shows the comparison of the end to end delay of four topologies and the delay of MDMIN is found to be same as that of torus and diagonal mesh topology. From the results, it can be concluded that structural properties of the MDMIN is better than simple 2D mesh interconnection network and diagonal mesh (toroidal) interconnection networks. However, the simulation results suggest that this approach is found to be better only in comparison to simple mesh.

3.3 Modified Diagonal Mesh Shuffle-Exchange Interconnection Networks

Further improvement in the MDMIN leads to the development of Modified Diagonal Mesh Shuffle-Exchange Interconnection Network (MDMSEIN). In MDMSEIN, horizontal and vertical links on the external nodes are replaced with shuffle exchange network with an objective to reduce the average path length of MDMIN over the torus mesh and diagonal mesh. Figure 3.6 describes the basic shuffle exchange network.

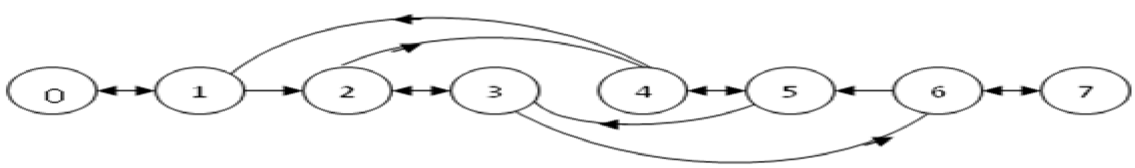

Figure 3.6: The Architecture of Shuffle-Exchange Network of 8 Nodes

Selecting the shuffle exchange reduces the diameters along the horizontal and vertical node of the network. The diameter of the shuffle exchange network of 'N' nodes is equal to $2*log_2(N-1)$[71]. Moreover, MDMSEIN of the 8×8 is described in Figure 3.7. It may be noted that the topology is complex in nature and incur extra cost at design time. Similarly, the running cost will be high due to the extra links and higher number of ports in the router.

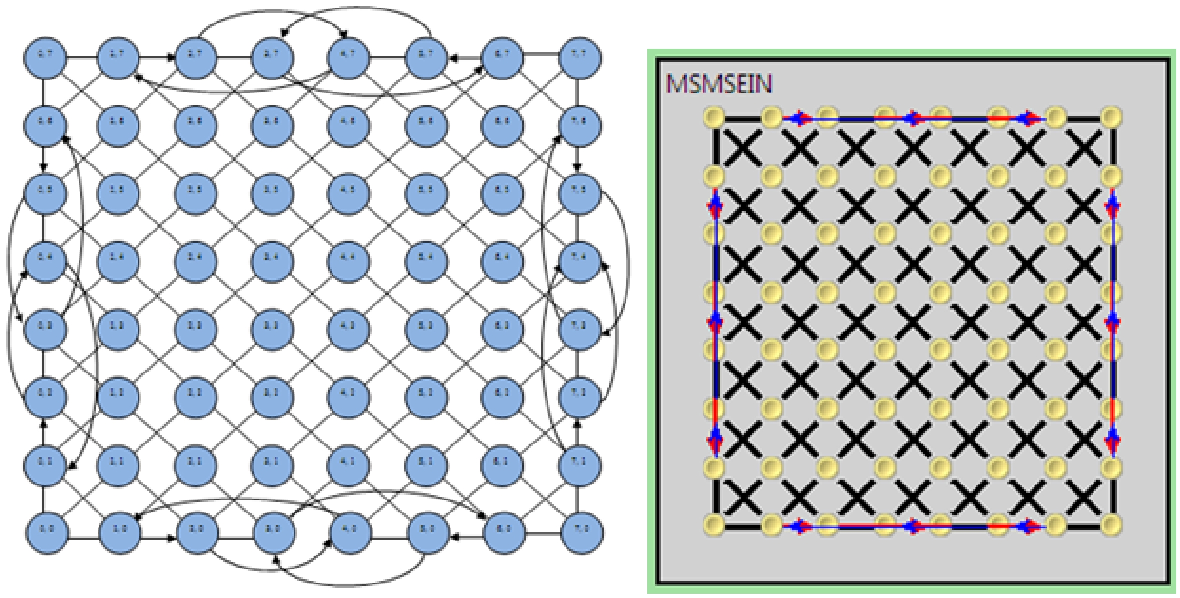

Figure 3.7: The Modified Diagonal Mesh Shuffle-Exchange Interconnection Networks

The assumptions are made such that all the nodes of topology are placed in the first quadrant of the coordinate system. For any node, the coordinates are as (x, y) and the set N describes the coordinates as a pair of neighbour is given by the equation (3.4).

Where D is chosen as a set of coordinates for nodes that are diagonally connected to neighbour of node with coordinate (x, y) and S is a set that contains the coordinate of nodes that are neighbour and a part of the shuffle exchange network and k = n-1 for n nodes in a row or column.

$$\mathrm{N} = D \cup \{S - (x, y)\} \tag{3.4}$$

$$D = X_i \times Y_i \tag{3.5}$$

Where *Xi* and *Yi* are the sets given by the notations as follows:

$$Xi = \begin{Bmatrix} (x+1), (x-1) & 0 < x < k \\ (x+1) & x = 0 \\ (x-1) & x = k \end{Bmatrix} \tag{3.6}$$

Similar equation regarding *Yi* can be used to represent the *Y* coordinate points, and set S can be described by the equation (3.7)

$$S = \left\{ \begin{array}{ll} (x, 2y\%k) & x = 0 \; or \; x = k \; and \; y\,\%2 = 0 \; and \; y\%k \neq 0 \\ (x, y+1) & x = 0 \; or \; x = k \; and \; y\,\%2 = 0 \\ (x, 2y\%k) & x = 0 \; or \; x = k \; and \; y\%2 \neq 0 \; and \; y\%k \neq 0 \\ (x, y-1) & x = 0 \; or \; x = k \; and \; y\,\%2 \neq 0 \\ (2x\%k, y) & y = 0 \; or \; y = k \; and \; x\%2 = 0 \; and \; x\%k \neq 0 \\ (x+1, y) & y = 0 \; or \; y = k \; and \; x\,\%2 = 0 \\ (2x\%k, y) & y = 0 \; or \; y = k \; and \; x\%2 \neq 0 \; and \; x\%k \neq 0 \\ (x-1, y) & y = 0 \; or \; y = k \; and \; x\%2 \neq 0 \\ \phi & otherwise \end{array} \right\} \tag{3.7}$$

Two distinct cases are studied to achieve a better understanding of these equations:

Case 1: When the node considered is an internal node of the MDMSEIN interconnection of 8 × 8.

Let the node have the coordinates (1, 5), initially determine the X_i and *Yi* based on the formulae given above. The X_i evaluates to {0, 2, similarly is evaluated to {4, 6}. Now the Cartesian product of X_i and Y_i is evaluated as D= {(0, 4), (0, 6),(2, 4),(2, 6)}. Since, the x, y is not the boundary values, values of *S* are also evaluated using the above equation and the result obtained is given by ϕ. So the node (1, 5) connected to N is given by {(0, 4), (0, 6), (2, 4), (2, 6)} From the Figure 3.7, this can be verified that there exists a link from node (1, 5) to these four nodes.

Case 2: when the node is a boundary node.

Now consider the nodes (0, 7) based on equations it can be found that node (0, 7) is connected to three nodes {(0, 6,),), (1, 7,),), (1, 6)}. From the Figure 3.7, it can be verified that the node (0, 7) has the links to these three nodes.

3.3.1 Testbed for Testing the MDMSEIN

The results are evaluated using a system with Intel Core 2 CPU T5200@1.60 GHz, 2 GB of RAM which runs on Windows 7 and OMNeT++ Simulator version 4.4.1 for windows. Table 3.2 describes the initial network parameters used for the simulation.

Table 3.2 Primary network parameters used in the experimental setup

Network Parameters	Simulation Values (Scenario 1)	Simulation Values (Scenario 2)
Channel bandwidth	1 Gbps	1 Gbps
Packet size	1024 Bytes	1024 Bytes
Channel delay	100ms	100ms
Number of nodes	64	64
Switch delay	0 ms	0 ms
Hotspots	-	5% and 10%
Number of runs	5	5

The test for the latencies on the various traffic patterns like uniform traffic, bit complement traffic, neighbour traffic and tornado traffic have been performed.

3.3.2 Results and Discussion

The performance of the five topologies at different loads and the latencies of the topologies on the uniform and neighbour are presented in Table 3.3, and

Table 3.4 presents the result of tornado and bit complement traffic. Table 3.5 represents the performance of the five topologies on hotspot traffic of 5 and 10% of uniform traffic.

3.3.2.1 Uniform Traffic:

From the results, it can be inferred that both the MDMIN and MDMSEIN interconnection are comparable to each other in performance. The mesh is seen to have a higher latency at small loads 10.24µs and are expressed in terms of inter packet arrival delay. The torus network shows an increase in latency at quite lower load in comparison to that of MDMIN and MDMSEIN. Besides, torus topology was seen to have improvements after a load with the for-packet injection rate reaches 10.24µs.From the results obtained it was noticed that DMESH provides better results due to higher degree bisection width and comprising of uniform degree throughout in the topology. The graph showing the comparison is shown in Figure 3.8

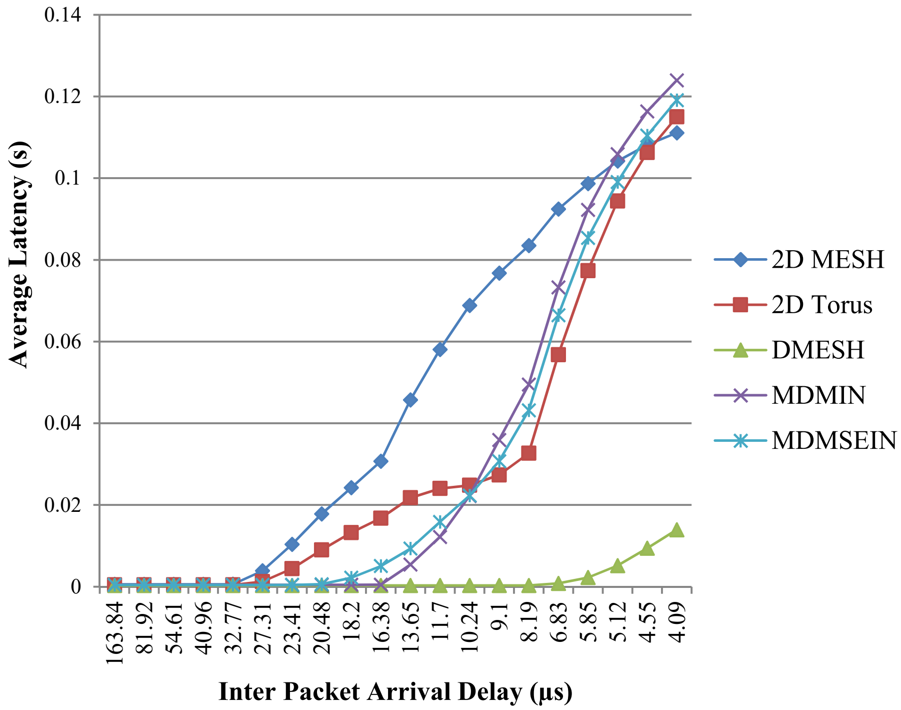

Figure 3.8: Average Latency on Uniform Traffic

3.3.2.2 Bit Complement Traffic

The MDMSEIN is found to have lower latency in comparison to MDMIN, torus and mesh topology. Figure 3.9 represents the graph showing the comparison of topologies. The MDMIN is also observed to provide better performance on small loads that is up to the inter packet arrival delay of 16.38 µs. However, this approach was seen to have higher latency at higher loads with inter packet arrival delay of less than 16.38 µs. The MDMSEIN provided an improvement of 9% based on the relation described by the equation (3.8) given below.

$$MMRL = \frac{1}{n}\sum_{i=1}^{n} L_{\mathrm{MDMSEIN}} - L_{torus}\, X\ 100 \tag{3.8}$$

The average latency of MDMSEIN is also comparable to that of DMESH for the inter-packet arrival delay of 13.65µs. Nevertheless, it was seen to increase with a higher load with inter packet arrival delay less than 13.65 µs.

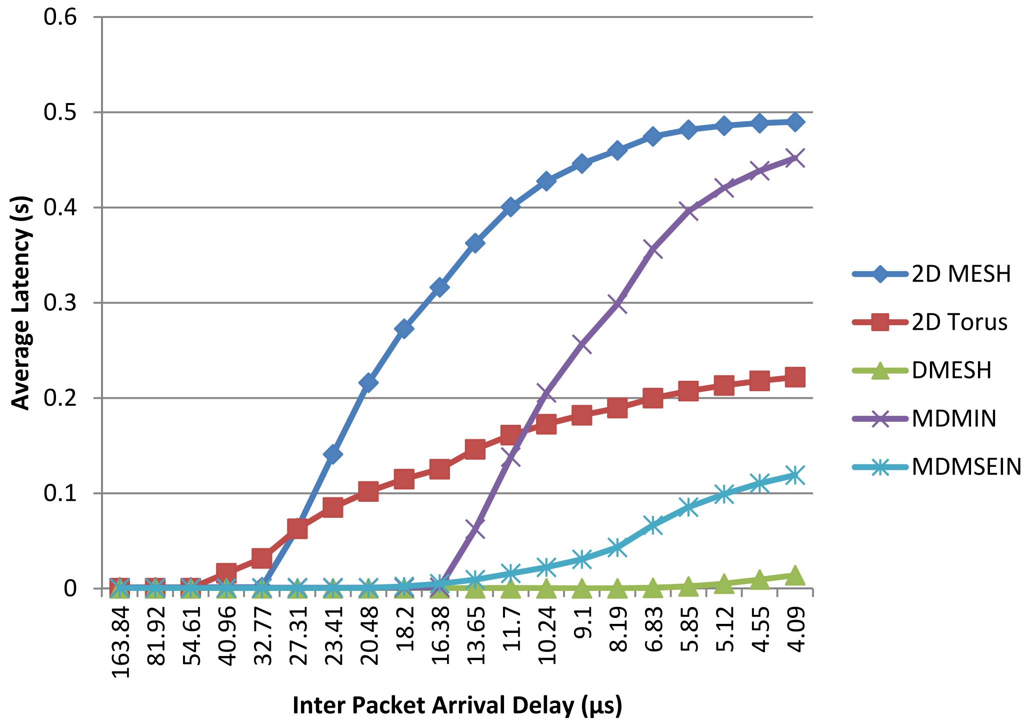

Figure 3.9: Average Latency on Bit complement Traffic

3.3.2.3 Tornado Traffic

The comparison of the all the topologies was shown in the Figure 3.10. The latency of the three topologies Torus, MDMIN and MDMSEIN were found to be similar upto the inter packet arrival delay of 20.48 µs. However, these are found to have lower latency at the lower inter arrival packet delay in the torus topology when compaered with MDMIN and MDMSEIN. Moreover, the topologies MDMIN and MDMSEIN is found to be better than mesh topology while the complexity of these topologies is found to be very high in comparison to the mesh topology. The area required, cost and complexity of the router with an increase in the number of ports is found to be significant challenges.

3.3.2.4 Neighbour Traffic

The performance of the MDMSEIN and MDMIN is compared to that of the torus. From this comparison, it was ascertained that the performance of the proposed methodologies is better than that of the simple mesh as described in Figure 3.11. From the graph plotted it can be observed that at higher load, and at the inter packet arrival delay of 13.65 μs, the average latency is less in comparison to other topologies.

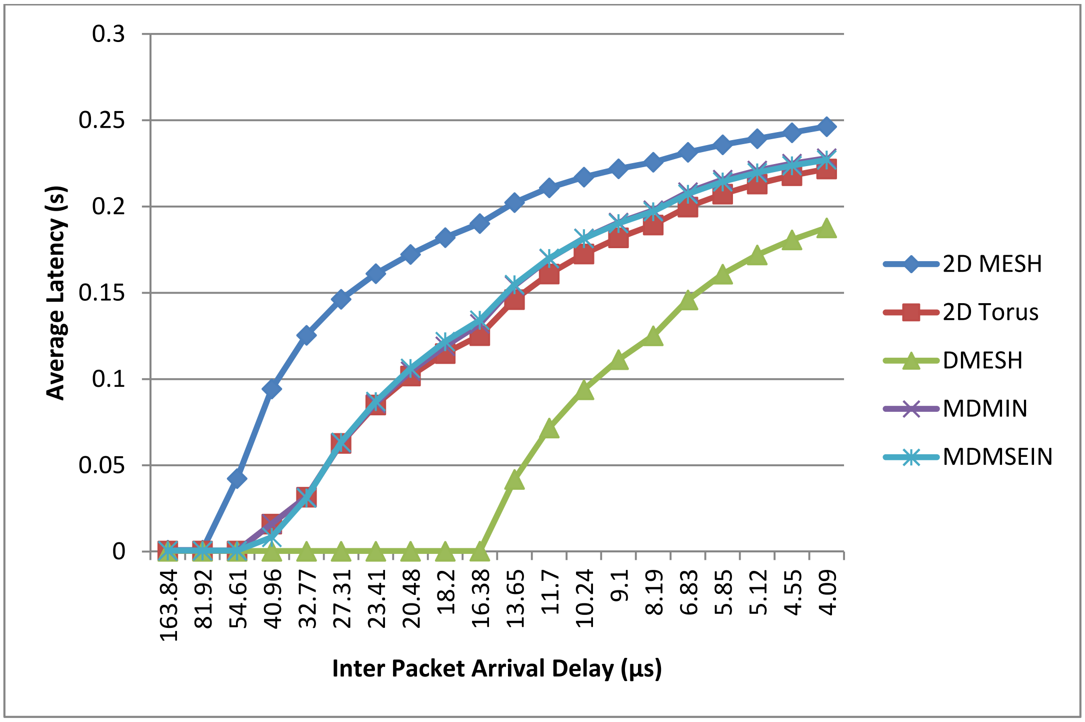

Figure 3.10: Average Latency on Tornado Traffic

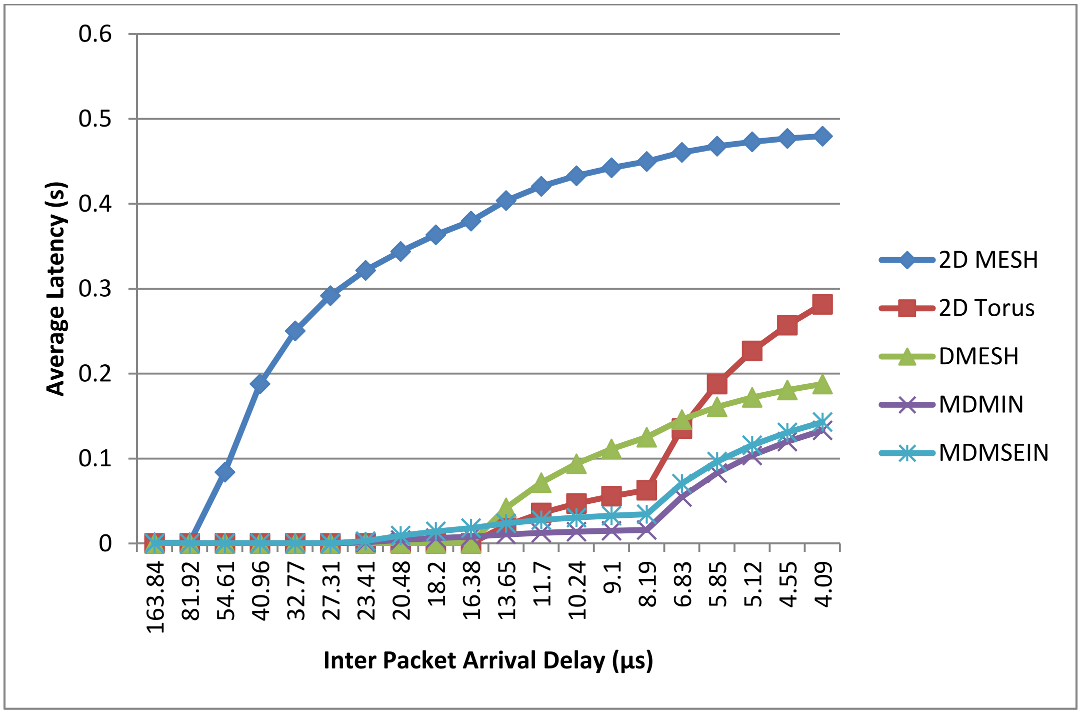

Figure 3.11: Average Latency on Neighbor Traffic

3.3.2.5 Scenario 2

In scenario 2, the uniform traffic with 5% and 10% of the hotspot is applied, and the Figure 3.12 shows proposed topologies behaves similar to that of torus topology, but at a hotspot of 10% Figure 3.13, the proposed topologies was found to have a lower latency at the inter packet arrival delay of 23.41 μs and lower.

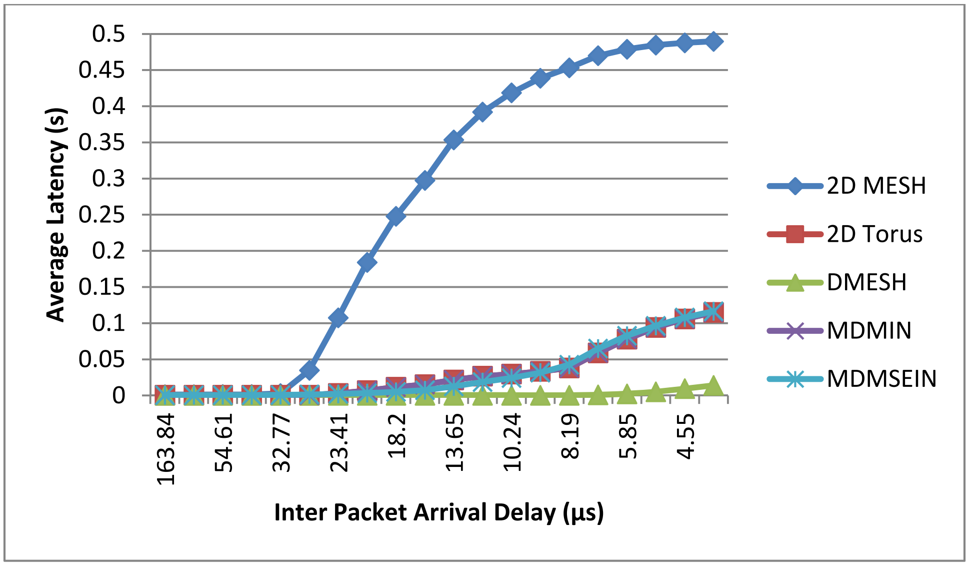

Figure 3.12: Average Latency on Uniform traffic with 5 % Hotspot

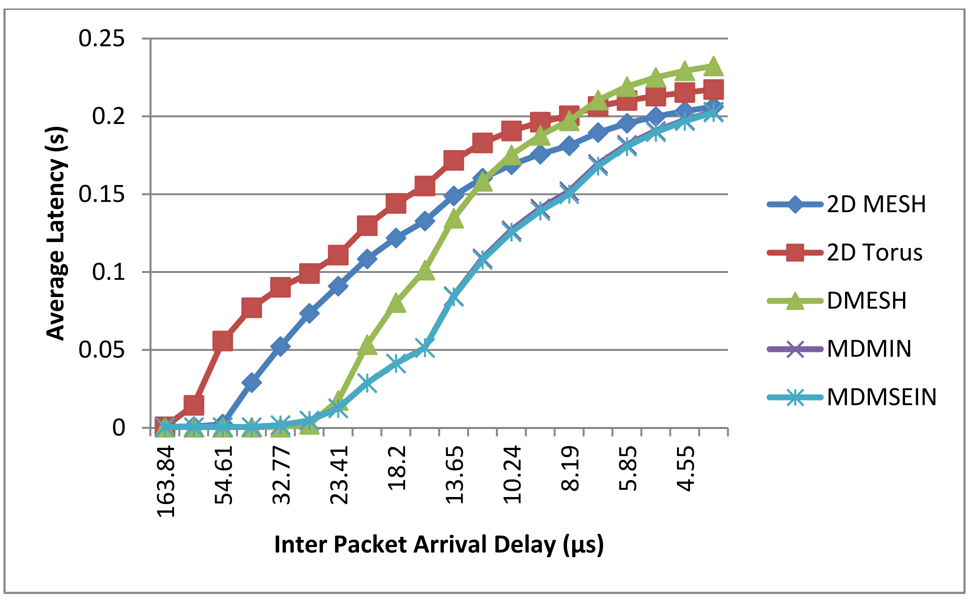

Figure 3.13: Average Latency on Uniform Traffic with 10% Hotspot

Table 3.3: Average latency of networks on uniform and neighbour traffic

Inter-packet Arrival delay (µs)	Traffic Type									
	Uniform Traffic					Neighbour Traffic				
	Topology					Topology				
	2D Mesh (s)	2D Torus (s)	DMESH (s)	MDMIN (s)	MDMSEIN (s)	2D Mesh (s)	2D Torus (s)	DMESH (s)	MDMIN (s)	MDMSEIN (s)
163.84	0.000571	0.000435	0.000284	0.000454	0.000431472	0.000872	0.000435	0.000284	0.00057	0.000571
81.92	0.000571	0.000435	0.000284	0.000455	0.00043185	0.000872	0.000435	0.000284	0.00057	0.000571
54.61	0.000578	0.00044	0.000286	0.000459	0.00043649	0.08403	0.000446	0.000286	0.000582	0.000581
40.96	0.000573	0.000436	0.000284	0.000455	0.000432323	0.187913	0.016037	0.000284	0.016165	0.008367
32.77	0.000582	0.00044	0.000284	0.000456	0.00043336	0.250268	0.03164	0.000284	0.031762	0.03112
27.31	0.003872	0.001237	0.000286	0.00046	0.000437953	0.291865	0.06283	0.000286	0.062943	0.063398
23.41	0.010367	0.004401	0.000285	0.000459	0.000437461	0.321579	0.08511	0.000285	0.085956	0.087063
20.48	0.017812	0.008986	0.000285	0.000462	0.000613115	0.343896	0.101856	0.000285	0.104557	0.106215
18.2	0.024221	0.013237	0.000286	0.000468	0.002190442	0.363343	0.114888	0.000286	0.11903	0.121761
16.38	0.03071	0.016765	0.000286	0.000485	0.005062601	0.379638	0.125295	0.000286	0.132014	0.134127
13.65	0.045713	0.021761	0.000287	0.005415	0.009326045	0.403725	0.146114	0.000287	0.154138	0.154806
11.7	0.058063	0.024017	0.000287	0.012164	0.015851163	0.420582	0.160981	0.000287	0.169725	0.170016
10.24	0.068851	0.024815	0.000288	0.02221	0.022305454	0.432932	0.172559	0.000288	0.181632	0.18146
9.1	0.076763	0.02734	0.000289	0.035915	0.03071708	0.442354	0.181918	0.000289	0.190756	0.190223
8.19	0.083489	0.032658	0.00029	0.049456	0.04314316	0.449775	0.189419	0.00029	0.197881	0.19709
6.83	0.092411	0.056771	0.00081	0.073252	0.066463112	0.460431	0.199804	0.00081	0.208386	0.207242
5.85	0.098646	0.077359	0.002236	0.092251	0.085361825	0.467812	0.207278	0.002236	0.215666	0.214358
5.12	0.104185	0.094408	0.005157	0.105886	0.099065755	0.472943	0.213176	0.005157	0.220917	0.219636
4.55	0.108115	0.106266	0.009407	0.116335	0.110401698	0.476959	0.217998	0.009407	0.224916	0.223759
4.09	0.111097	0.114996	0.013942	0.123939	0.119092355	0.479603	0.221837	0.013942	0.228076	0.227043

Table 3.4: Average latency of networks on tornado and bit complement traffic

Inter-packet Arrival delay (μs)	Traffic Type									
	Tornado Traffic					Bit Complement Traffic				
	Topology					Topology				
	2D Mesh (s)	2D Torus (s)	DMESH (s)	MDMIN (s)	MDMSEIN (s)	2D Mesh (s)	2D Torus (s)	DMESH (s)	MDMIN (s)	MDMSEIN (s)
163.84	0.000871	0.000435	0.00027	0.00057	0.000571	0.001202	0.000435	0.00027	0.000781	0.000431
81.92	0.000872	0.000435	0.00027	0.00057	0.000571	0.00121	0.000435	0.00027	0.000784	0.000432
54.61	0.042426	0.000446	0.000274	0.000582	0.000581	0.001229	0.000446	0.000274	0.000804	0.000436
40.96	0.094318	0.016037	0.00027	0.016165	0.008367	0.001257	0.016037	0.00027	0.000793	0.000432
32.77	0.125469	0.03164	0.00027	0.031762	0.03112	0.00141	0.03164	0.00027	0.000803	0.000433
27.31	0.146279	0.06283	0.000274	0.062943	0.063398	0.063418	0.06283	0.000274	0.000824	0.000438
23.41	0.161146	0.08511	0.000271	0.085956	0.087063	0.140894	0.08511	0.000271	0.000832	0.000437
20.48	0.172328	0.101856	0.000271	0.104557	0.106215	0.215966	0.101856	0.000271	0.000867	0.000613
18.2	0.182082	0.114888	0.000274	0.11903	0.121761	0.272583	0.114888	0.000274	0.00091	0.00219
16.38	0.190243	0.125295	0.000333	0.132014	0.134127	0.316167	0.125295	0.000333	0.001229	0.005063
13.65	0.202359	0.146114	0.041954	0.154138	0.154806	0.362601	0.146114	0.041954	0.062334	0.009326
11.7	0.210864	0.160981	0.07168	0.169725	0.170016	0.400509	0.160981	0.07168	0.137958	0.015851
10.24	0.217126	0.172559	0.093936	0.181632	0.18146	0.427525	0.172559	0.093936	0.205394	0.022305
9.1	0.221934	0.181918	0.111313	0.190756	0.190223	0.446052	0.181918	0.111313	0.256323	0.030717
8.19	0.225751	0.189419	0.125182	0.197881	0.19709	0.459777	0.189419	0.125182	0.298464	0.043143
6.83	0.231503	0.199804	0.145931	0.208386	0.207242	0.47448	0.199804	0.145931	0.356343	0.066463
5.85	0.235832	0.207278	0.160886	0.215666	0.214358	0.481536	0.207278	0.160886	0.396095	0.085362
5.12	0.239367	0.213176	0.172027	0.220917	0.219636	0.485781	0.213176	0.172027	0.420605	0.099066
4.55	0.242848	0.217998	0.180726	0.224916	0.223759	0.488456	0.217998	0.180726	0.438444	0.110402
4.09	0.246374	0.221837	0.187751	0.228076	0.227043	0.489804	0.221837	0.187751	0.451893	0.119092

Table 3.5: Average latency of network on uniform traffic with hotspot

Inter-packet Arrival delay (µs)	Uniform traffic with Hotspot									
	5% Hotspot					10 % Hotspot				
	Topology					Topology				
	2D Mesh (s)	2D Torus (s)	DMESH (s)	MDMIN (s)	MDMSEIN (s)	2D Mesh (s)	2D Torus (s)	DMESH (s)	MDMIN (s)	MDMSEIN (s)
163.84	0.001177	0.000435	0.000284	0.000453	0.000431	0.000569	0.000435	0.000283	0.000454	0.000431
81.92	0.001183	0.000434	0.000284	0.000453	0.000431	0.000569	0.000434	0.000283	0.000454	0.000431
54.61	0.001201	0.000439	0.000286	0.000458	0.000436	0.000577	0.000439	0.000285	0.000459	0.000436
40.96	0.001221	0.000435	0.000284	0.000454	0.000431	0.002107	0.000435	0.000283	0.000495	0.000523
32.77	0.002649	0.000438	0.000284	0.000527	0.000589	0.003388	0.000438	0.000283	0.001346	0.001393
27.31	0.03464	0.000978	0.000286	0.001352	0.001426	0.005	0.000978	0.000285	0.002288	0.001897
23.41	0.107418	0.002538	0.000285	0.001881	0.002088	0.010098	0.002538	0.000284	0.002997	0.003008
20.48	0.184085	0.005703	0.000285	0.002797	0.003233	0.017877	0.005703	0.000284	0.003461	0.004555
18.2	0.247826	0.010838	0.000286	0.003704	0.004973	0.025569	0.010838	0.000285	0.004143	0.005937
16.38	0.297272	0.014392	0.000286	0.005102	0.007201	0.031722	0.014392	0.000284	0.005718	0.007656
13.65	0.353349	0.021876	0.000287	0.009276	0.012612	0.044382	0.021876	0.000285	0.010251	0.013437
11.7	0.391859	0.028363	0.000287	0.014995	0.018594	0.05593	0.028363	0.000286	0.015592	0.020092
10.24	0.418498	0.03211	0.000288	0.023588	0.024221	0.066774	0.03211	0.000542	0.023898	0.025613
9.1	0.438623	0.035943	0.000289	0.035756	0.031637	0.075076	0.035943	0.001189	0.035473	0.032666
8.19	0.453113	0.04016	0.00029	0.047328	0.04252	0.082217	0.04016	0.001695	0.046453	0.042187
6.83	0.46981	0.056401	0.00081	0.06989	0.064774	0.091524	0.056401	0.00229	0.066687	0.062875
5.85	0.478846	0.074441	0.002236	0.087831	0.082668	0.098019	0.074441	0.0026	0.083731	0.079318
5.12	0.484525	0.089554	0.005157	0.10175	0.096373	0.102852	0.089554	0.004474	0.097166	0.092024
4.55	0.487648	0.101506	0.009407	0.11222	0.107773	0.106141	0.101506	0.008247	0.107669	0.102775
4.09	0.48969	0.110675	0.013942	0.120021	0.116495	0.108823	0.110675	0.012252	0.115666	0.111449

3.4 Three-dimensional Topology based on Modified Diagonal Mesh Interconnection Networks

The three-dimensional topologies are practically more popular in comparison to two-dimensional topologies due to the fact that the inter-hop distance between the various nodes reduces. Besides, an increase in the dimension of the topology is found to reduce the diameter of the topology. The topology becomes fault tolerant with the increase in the number of links that increases the bisection width of the topology. The degree of the topology is found to be slightly increased in this process. Figure 3.14 represents the Three - Dimensional topology based on Modified Diagonal Mesh Interconnection Network (3D MDMIN). The 3D MDMIN is compared with the popular topologies such as 3D Mesh, 3D torus and 2D torus to ascertain the performance of the system. The 3D MDMIN topology uses the 2D MDMIN as the building block. The multiple layers of the topology is placed over the other. These layers are connected to each other by the mathematical equation (3.9)

$$E = \left\{ \begin{array}{ll} (x, y, z \pm 1) & if\ z < l - 1\, and\ z > 0 \\ x, y, z - 1) & if\ z = l - 1 \\ (x, y, z + 1) & if\ z = 0 \end{array} \right\} \tag{3.9}$$

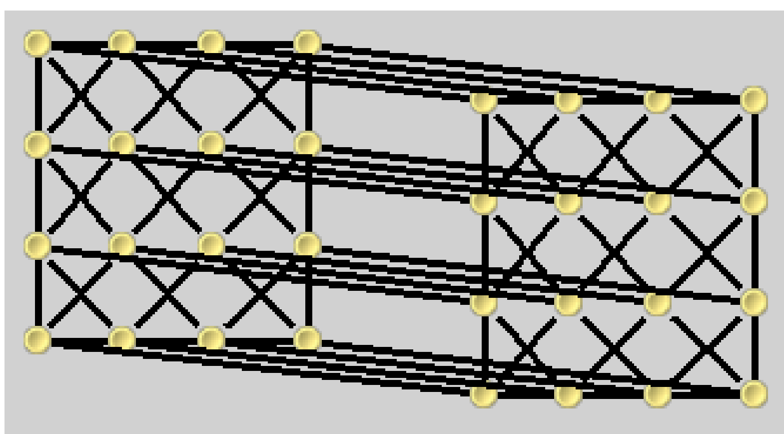

Figure 3.14: Proposed Topology Based on MDMIN 4 × 4 × 2

3.4.1 Experimental Setup for the Testing of the Topology

The topology was tested using six traffic patterns and is compared with the three existing topologies. Table 3.6 describes the hardware employed in testing the performance of the topology.

Table 3.6: Describes the configuration of hardware used for testing the topologies

S.no.	Hardware Configuration	Specification value
1	Processor	Intel®core™2 CPU T5200@1.6 GHZ
2	Ram	3 GB
3	Operating System	Windows 7 32 Bit
4	OmNeT++ Simulator version	4.4.1

The 32 nodes topology has been tested using the omnet++ using the configuration as stated in Table 3.7.

Table 3.7: Parameters used in omnet++ for testing the topologies

S.no.	Parameter Name	Parameter Value
1	Simulation Time	0.5 s
2	Warm-up Period	50 ms
3	Traffic Patterns	Uniform Traffic Bit Complement Traffic Neighbor Traffic Tornado Traffic Bit Transpose Traffic Bit Reversal Traffic
4	Inter-Packet Arrival Delay	163.84 μs 81.92 μs 54.61 μs 40.96 μs 32.77 μs 27.31 μs 23.41 μs 20.48 μs 18.20 μs 16.38 μs
5	Topologies	2DTorus (4X8) 3D mesh (4X4X2) 3D Torus (4X4X2) 3D MDMIN (4X4X2)
6	Channel Data Rate	1Gbps
7	Number of runs	5

3.4.2 Results and Discussion

3.4.2.1 Uniform Traffic:

In the case of uniform traffic, the performance of the topology is found to be better than the other three topologies. The graph of average latency is shown in the Figure 3.15. From the Figure 3.15, it is observed that latency of 3D MDMIN on comparison to the inter packet arrival is found to have a delay of 23.41 μs. Nevertheless, with a decrease in the inter packet arrival delay an increase in the latency of other topologies were observed. From Figure 3.16, the average throughput of the 3D MDMIN is found to be marginally more than the other three topologies under consideration.

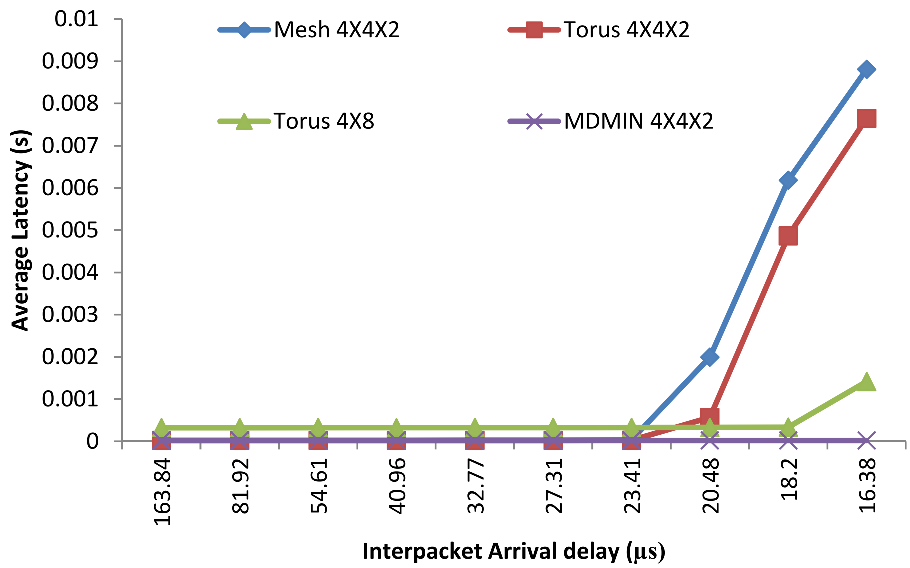

Figure 3.15: Comparison of Average Latency 3D-MDMIN at Uniform Traffic

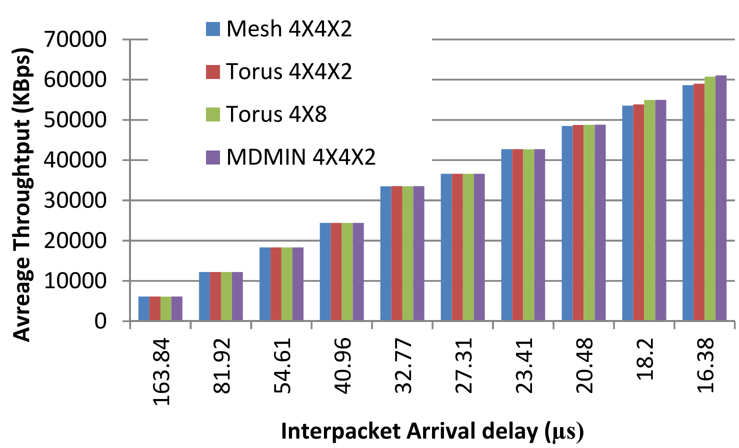

Figure 3.16: Comparison of Average Throughput at Uniform Traffic

3.4.2.2 Bit Complement Traffic

In the case of the bit-complement traffic, the two-dimensional torus topology is found to perform better in comparison to the other topologies. From Figure 3.17 it can be observed that the 3D MDMIN has exact latency trends as that of the 3D torus. The throughput graph is shown in the Figure 3.18 also shows the same story. The effective performance of the 2D torus on the bit complement is due to the placement of the nodes is favoured by bit complement traffic.

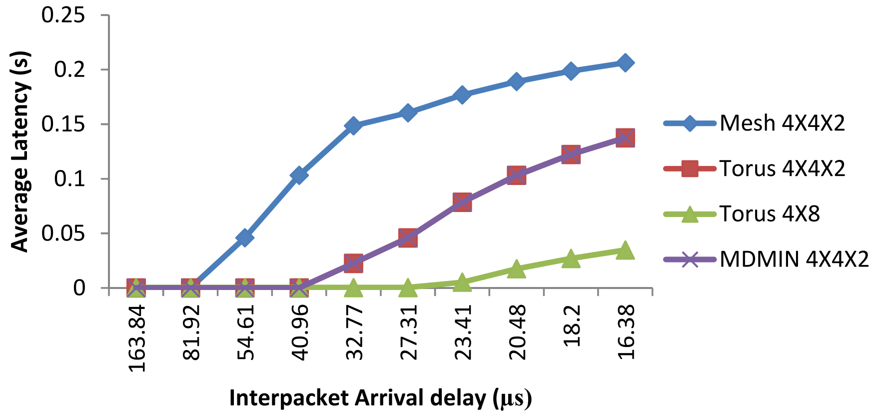

Figure 3.17: Comparison of Average Latency 3D-MDMIN at Bit Complement traffic

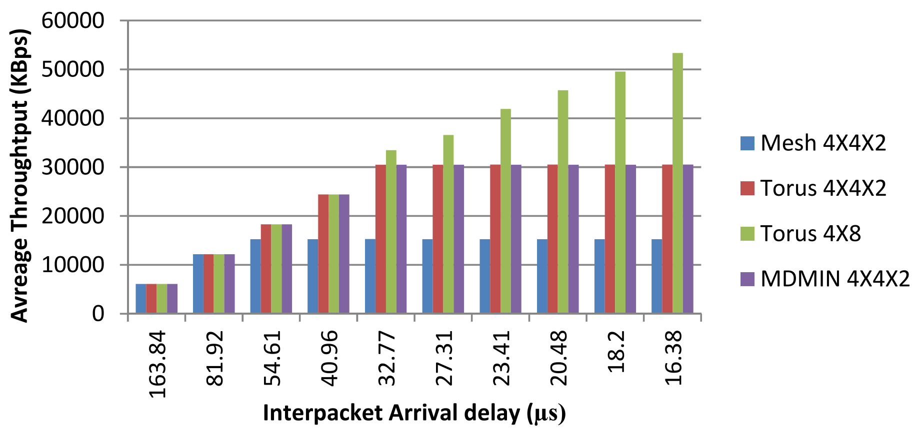

Figure 3.18: Comparison of Average Throughput at Bit Complement Traffic

3.4.2.3 Neighbour Traffic

In the case of neighbour traffic, there are two types of neighbour; one of the neighbour, that may be situated on the diagonals as in the case of proposed topology, while the other is located on the horizontal and vertical adjacent nodes. The analysis is performed by considering the horizontal or vertical neighbour against the designed topology. The results obtained indicate that the proposed topology shows a better performance than the other two 3-Dimesional topologies, but it is slow in comparison to 2 D torus topology. From Figure 3.19, it can be observed that 3D MDMIN has same latency till 27.31 µs but at higher load (less inter arrival packet delay) 3D MDMIN is found to have a lower latency than 3D torus. Figure

3.20 illustrates that 2D torus provides best throughput in comparison to other 3 topologies. 3D MDMIN also is seen to perform better than 3D mesh and 3D torus due to the presence of diagonal links in the topology.

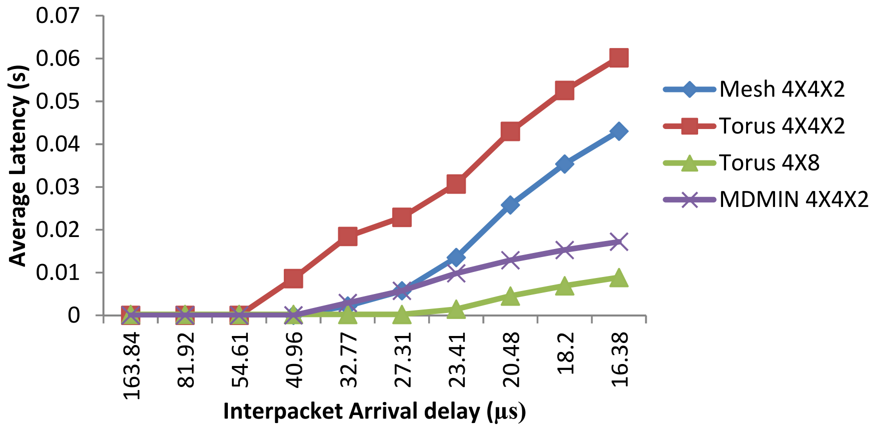

Figure 3.19: Comparison of Average Latency 3D-MDMIN at Neighbour Traffic

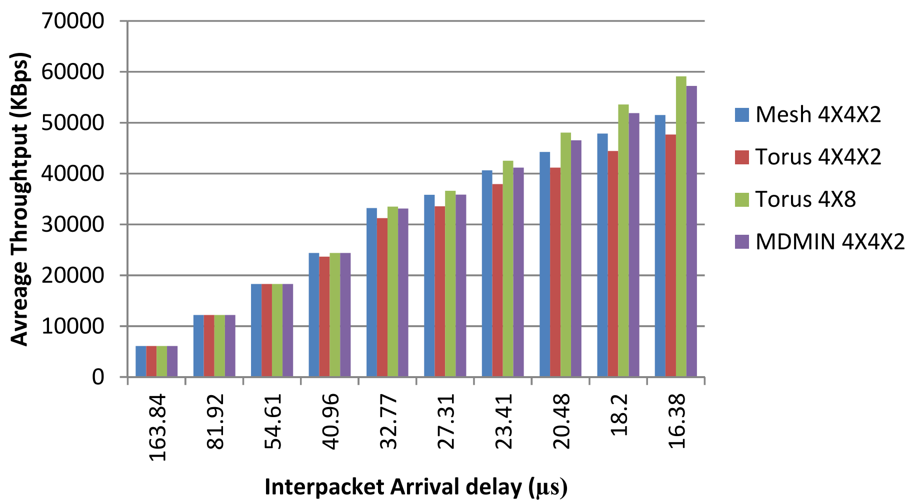

Figure 3.20: Comparison of Average Throughput at Neighbour Traffic

3.4.2.4 Tornado Traffic

The tornado traffic comprises values that are shuffled by at least half of the bits. It is considered to be worst digital permutation traffic. The performance of the 3D MDMIN is found to be similar to other three-dimensional topologies. The performance of the 2D torus is less than 3D topologies due to the hotspot effect connected at the corner nodes. The Figure 3.21 and Figure 3.22 describes the performance of the two-dimensional torus having poor performance in comparison to other topologies.

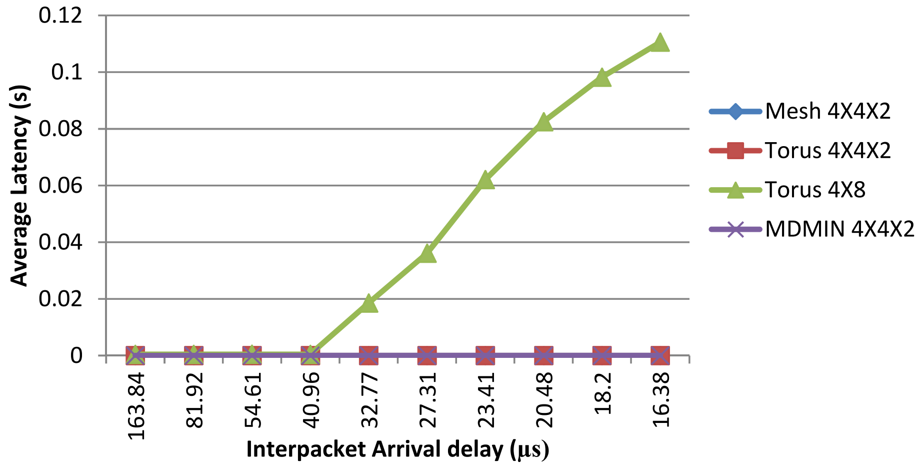

Figure 3.21: Comparison of Average Latency 3D-MDMIN at Tornado Traffic

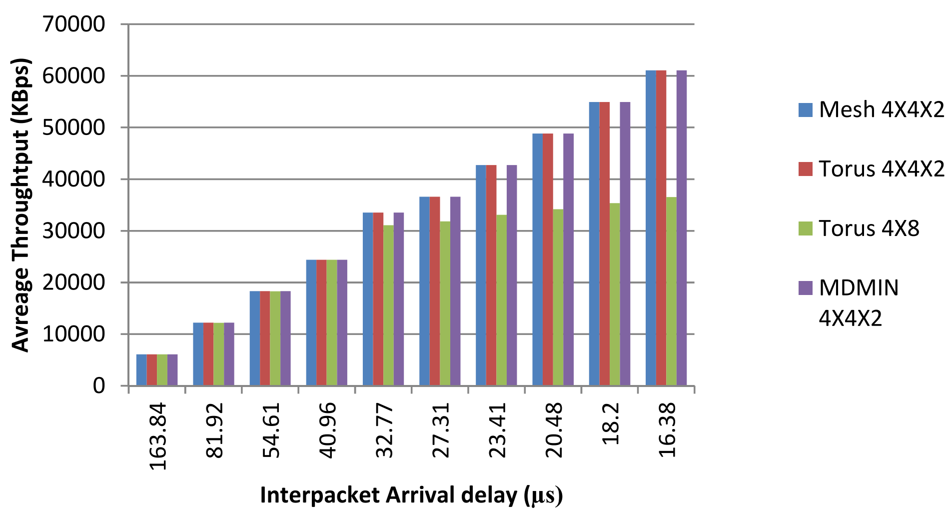

Figure 3.22: Comparison of Average Throughput at Tornado Traffic

3.4.2.5 Bit Transpose Traffic

This traffic is similar to the bit permutation and guarantees that at least half of the nodes are affected by the performance. The performance of the proposed topology is not found to be efficient in the presence of traffic. The designed system is dominated by 2D and 3D torus topologies as described in Figure 3.23 and Figure 3.24. From Figure 3.23 it is clear that 3D MDMIN is following the same trend as of 3D Mesh. Similarly, Figure 3.24 shows that the throughput is almost equal to 3D Mesh. However, the performance is found to be less in comparison to 3D torus and 2D torus.

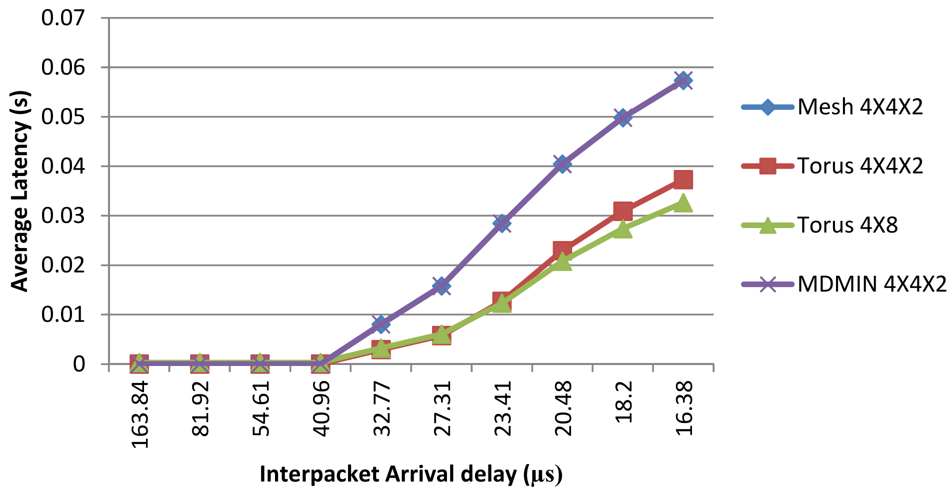

Figure 3.23: Comparison of Average Latency 3D-MDMIN at Bit Transpose Traffic

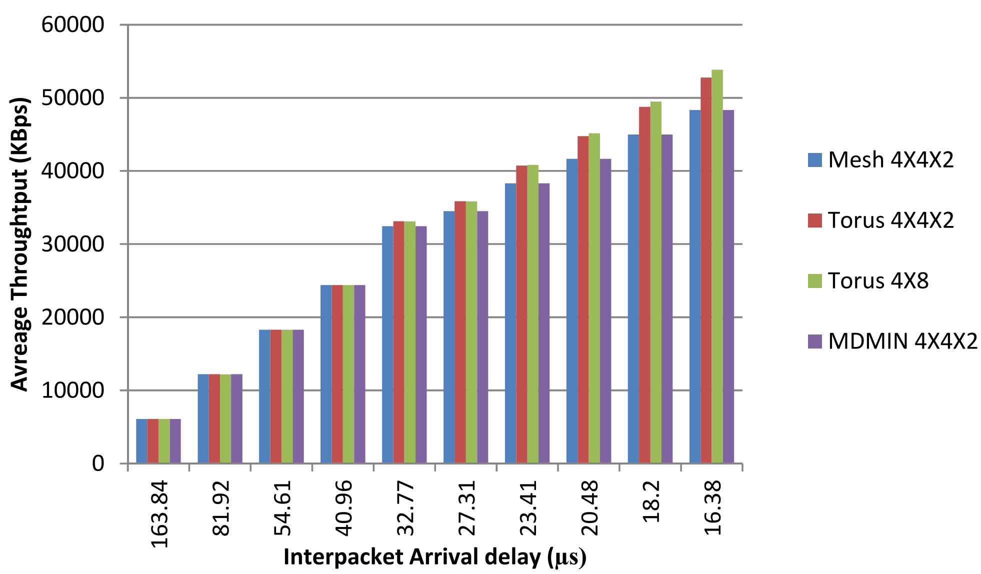

Figure 3.24: Comparison of Average Throughput at Bit Traversal Traffic

3.4.2.6 Bit Reversal Traffic

This traffic based on the bit permutation and guarantees that all the bits in the source and destination are changed from their positions. This is considered to be one of the worst traffic patterns. From the details described in Figure 3.25 and Figure 3.26, it is clear that proposed topology is best in the case of the Bit Reversal Traffic. From Figure 3.25 it is ascertained that after the inter packet arrival delay of 40.96 µs, latency of 3D MDMIN is very less in comparison to other topologies. Figure 3.26 shows that the high throughput gains are achieved at higher offered loads special at the inter packet arrival delay of 20.48 µs and lower

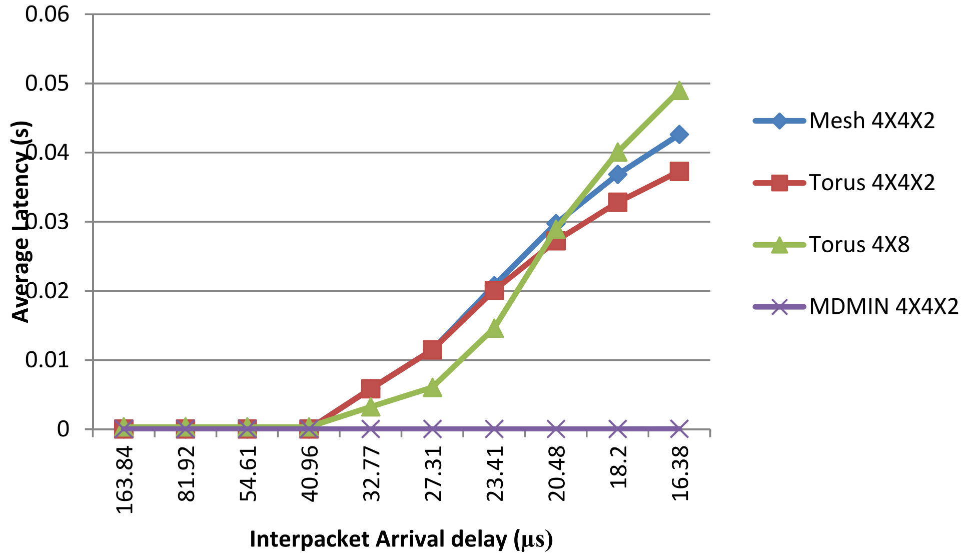

Figure 3.25: Comparison of Average Latency 3D-MDMIN at Bit Reversal Traffic

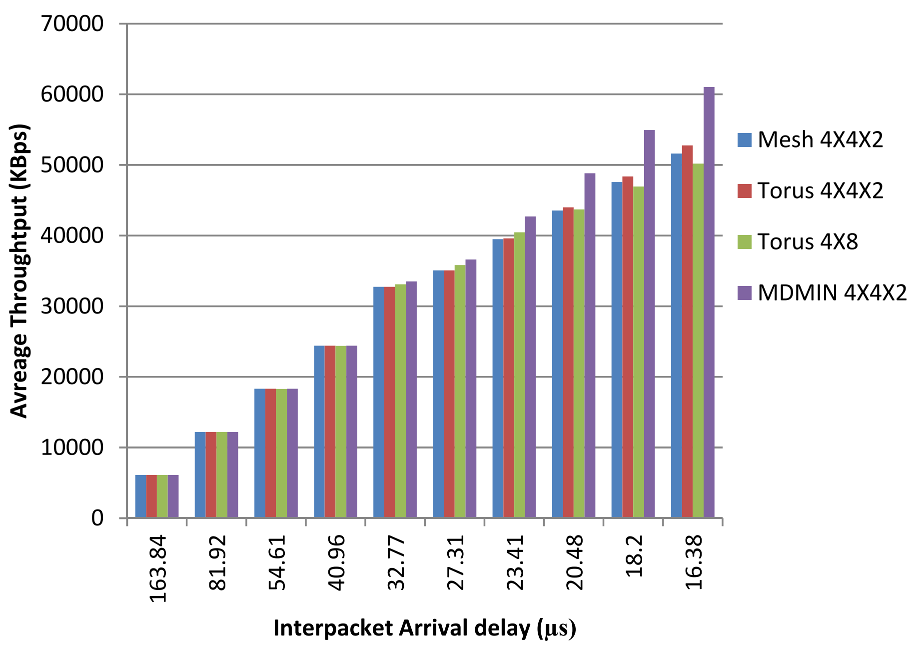

Figure 3.26: Comparison of Average Throughput at Bit Reversal Traffic

3.5 Diagonal Connected T Mesh

The Diagonal Connected T Mesh(DCT) is developed by the combination of the DCM and T Mesh. DCT is described in Figure 3.27 as shown below. From the figure, it can be observed that the degree of DCT is found to have a maximum value of 6, the bisection width has been increased by 2 in comparison to that of DCM which is found to be higher than T Mesh. The mathematical representation of the topology is described by the equation (3.10)

$$DCT = DCM \cup Tmesh \tag{3.10}$$

This means the neighbour nodes in the DCT mesh can be represented by equation (3.11).

$$DT(x, y) = D(x, y) \cup T(x, y) \tag{3.11}$$

In the above equation D(x, y) represents the set of nodes that are neighbour in DCM to node with coordinate (x, y) and T(x, y) represents the set of nodes that are neighbour to node with (x, y) in T mesh.

The design cost of the topology is not affected by the degree of the router and a maximum of 6, as that of DCM only four extra links are to be added which will have a marginal effect on the cost.

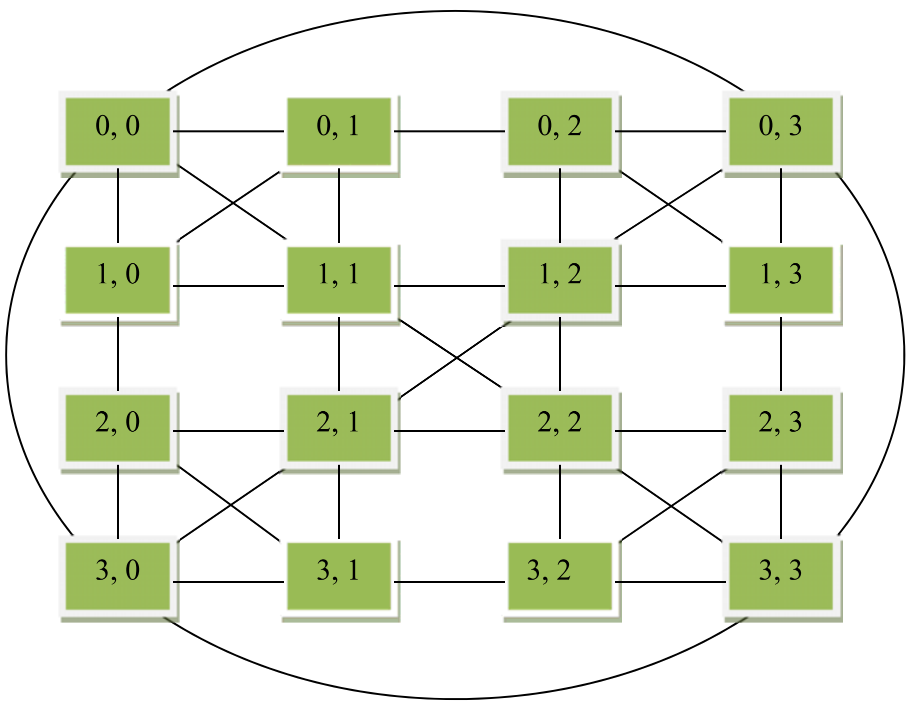

Figure 3.27: DCT 4 × 4

3.5.1 Experimental Setup

3.5.1.1 Hardware Specification

The topology has been tested on a 32-bit machine with window 7 SP1. The machine is incorporated with Intel® Core™2 CPU T5200 with a clock rate of 1.60 GHz. The System uses the 2 GB of Ram and Hard Disk at the speed of 7200 RPM.

3.5.1.2 Software Specification

The DCT is tested on the discrete event simulator omnet++ which is an open source software [72, 73]. Each node is considered as the origin and destination to evaluate the performance.The packets with the different source and destination are sent at various load factors. To create the different loads the packet injection rate of the source is varied from 163.84µs to 4.09µs.

The parameters used for designing the topology and the results obtained for comparison are described in Table 3.8 follows:

Table 3.8 Describing the parameters of the topology in omnet++

Sno.	Parameter Name	Value
1	Rows	4
2	Columns	4
3	Packet size	1024 bytes
4	Data rate	1Gbps
5	Simulation time	10 ms
6	Warm up time	0.5 ms
7	Number of runs	5

3.5.2 Results and Discussion

The three traffic patterns are used to analyse the performance of the netwok. The performance factors that are used to evaluate the performance are given by Average latency, Average Sink bandwidth. The hop count analysis of the topology is also performed using the specific source and destination.

3.5.2.1 Uniform traffic

In case of Uniform traffic node, the packets are send to each node with equal probability. Uniform traffic is considered as a e basic traffic while the performance of the mesh topologies is found to be better on this type of traffic. On the uniform traffic, the DCT is seen to perform better in comparison to the other topologies. From the Figure 3.28, it can be identified that

DCT topology has a lower latency than the other two topologies. The performance improvement is noticed at the lower inter packet arrival delay of 8.19 μs and lower. A similar trend can be observed in the case of sink bandwidth from the Figure 3.29.

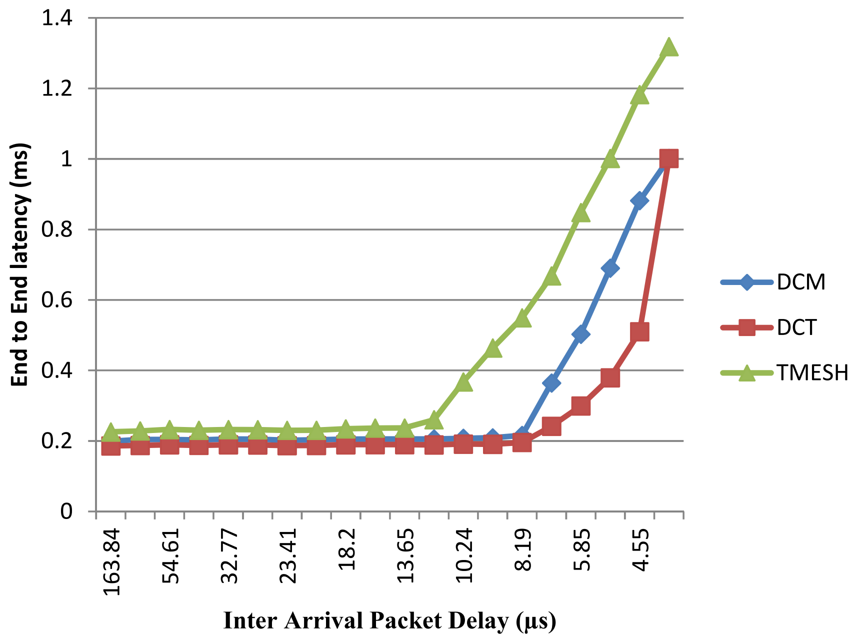

Figure 3.28: Average End to End Latency with Uniform Traffic

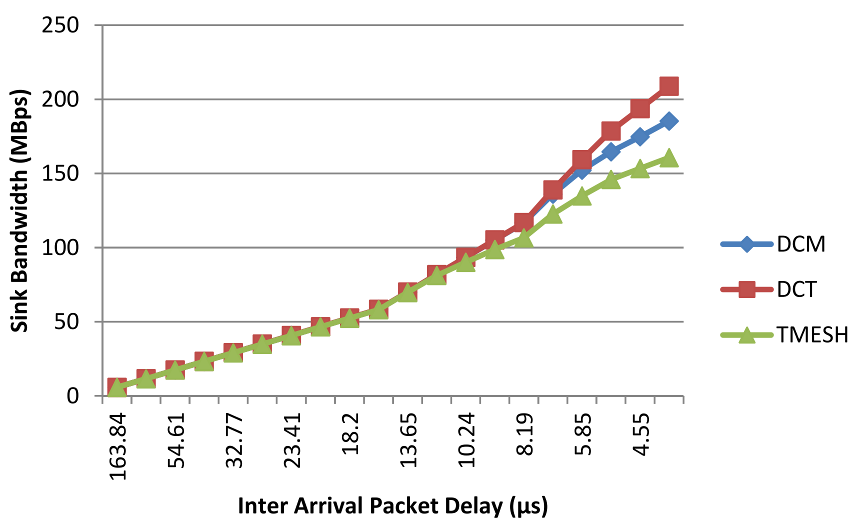

Figure 3.29: Sink Bandwidth with Uniform Traffic

3.5.2.2 Bit Complement traffic

Bit Complement traffic is represented by the equation 3.15. Consequently, the destination address is found to be the complement of the source address. For the simplicity the following relation is used.

$$D = N - S \tag{3.12}$$

Where D is the destination node address and S is the source node address in the above. The value of N is maximum Id assigned to the nodes. For the bit complement traffic, the destination address is computed based on the source address. The bit complement hold true if number of nodes are exactly in the range of 2^K. This concept is generalized by numbering the nodes from 0 to K – 1, while the complement is obtained by the actual subraction of the 2^N – 1 with source id. The same can be generalize to N – S.

Further, from the Figure 3.30and Figure 3.31, it can be observed that the performance of the DCT is better than the DCM and T Mesh topologies. The performance of the Tmesh and DCM was found to saturate at minimal load.

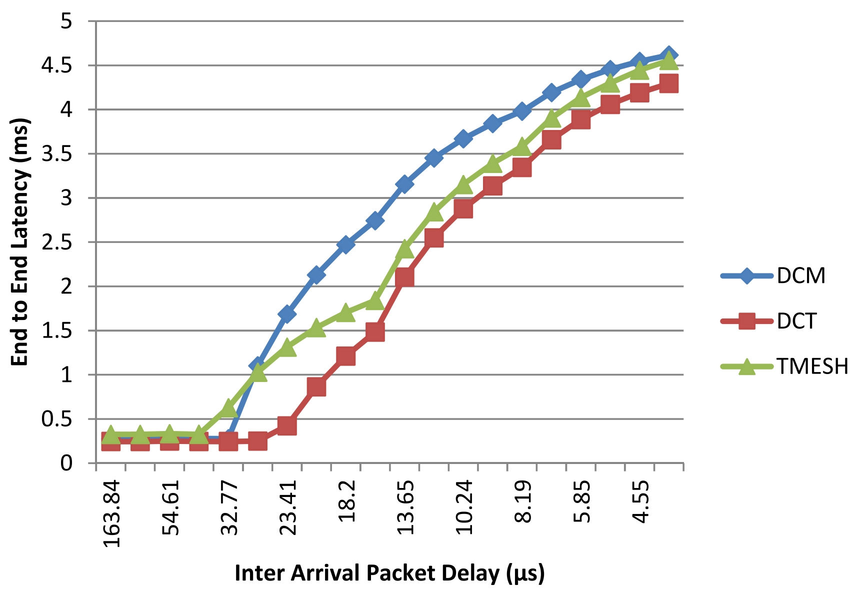

Figure 3.30: Average End to End Latency with Bit Complement traffic

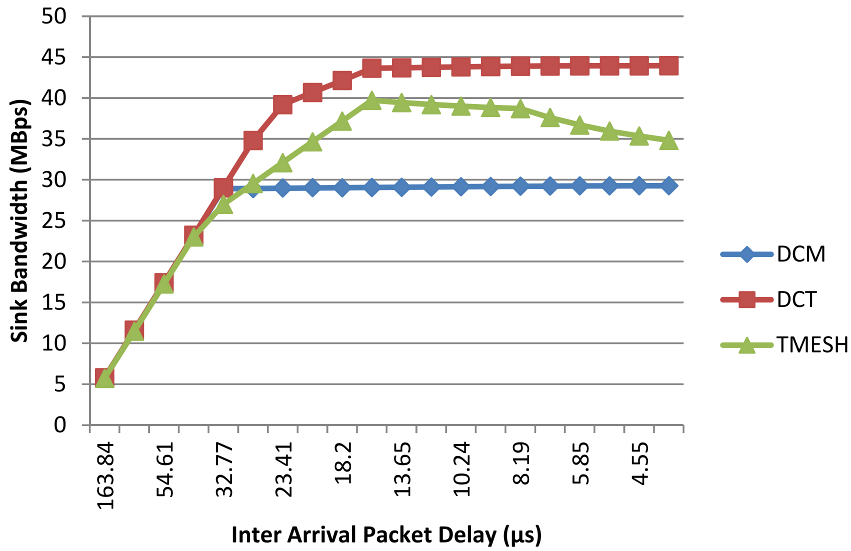

Figure 3.31: Sink Bandwidth with Bit Complement Traffic

3.5.2.3 Tornado Traffic

In the case of the Tornado traffic, the performance of the DCT topology is found to be identical to that of the DCM topology. From the Figure 3.32 and Figure 3.33, it can be observed that the DCT has performed better than Tmesh, but it is similar to the DCM.

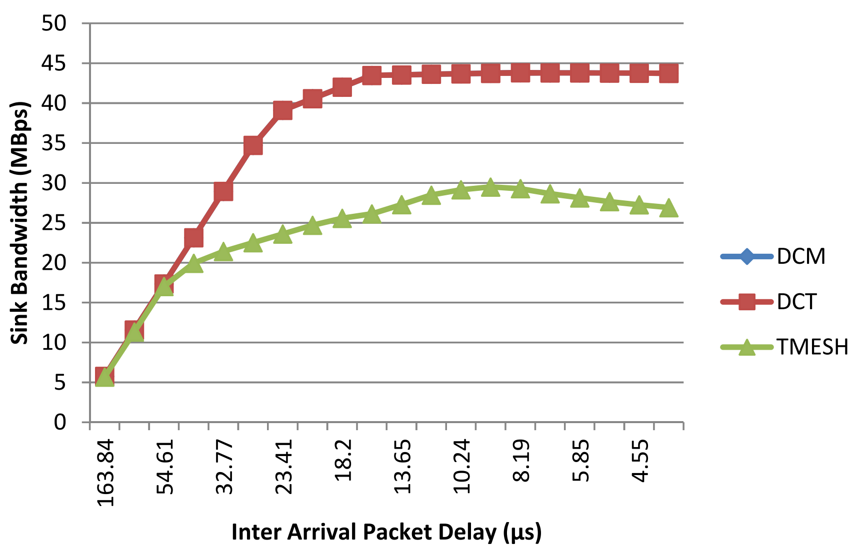

Figure 3.32: Average End to End Latency with Tornado Traffic

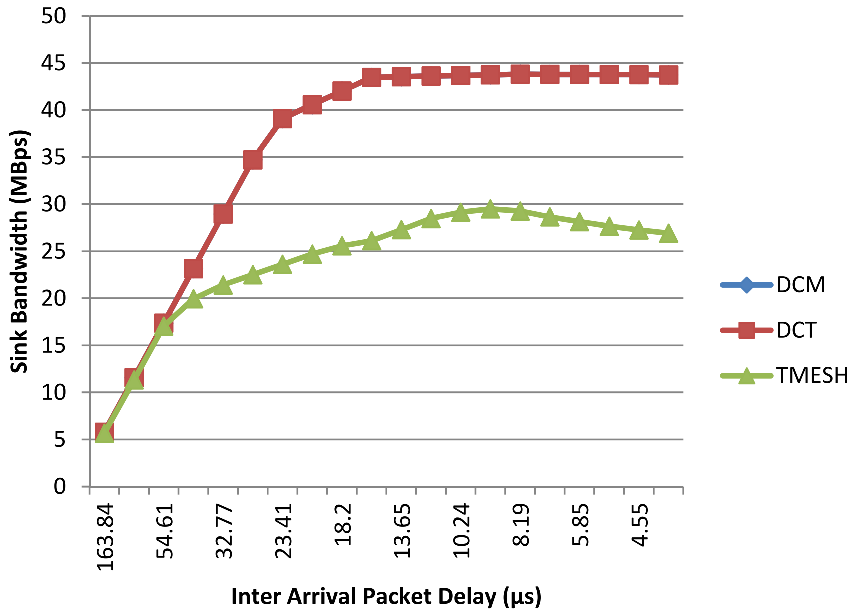

Figure 3.33: Sink Bandwidth with Tornado Traffic

3.5.2.4 Hop Count Analysis

3.5.2.4.1 Single Source Analysis

For the single source hop count analysis, packets from the fixed source are sent to the all other nodes, and their hop counts are recorded. The analsis is performed by considering Node 1 as the source. From the Figure 3.34 it is observed that the number of hops required by DCT is less when compared with the other two topologies. Based on the results shown in Figure 3.34 it can be inferred that 33% of the routes of DCM has become shorter for the source node 1.

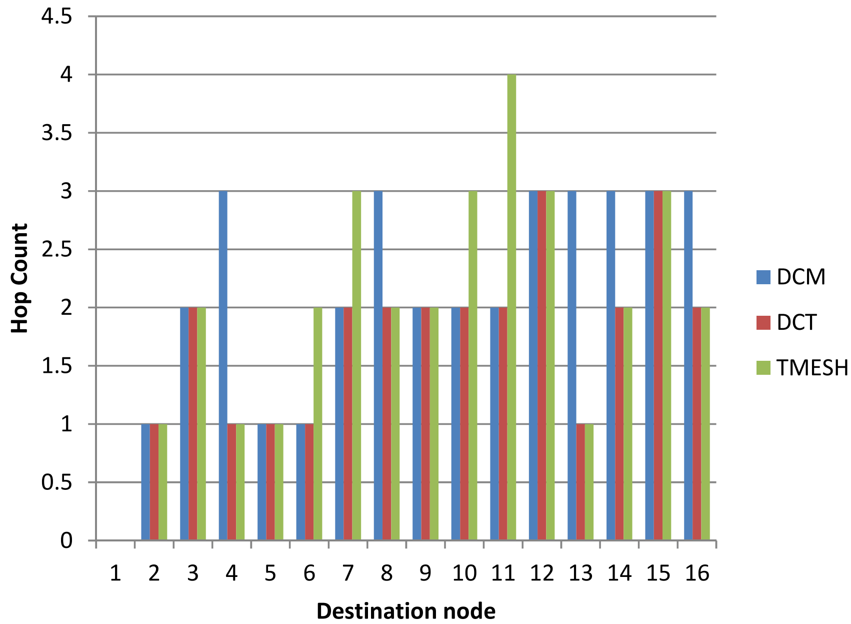

Figure 3.34: Hop Count to All Other Nodes Keeping Source as Node 1

3.5.2.4.2 Average Hop Count

Average hop count for the node is calculated by considering the mean value of all the hop counts for the single source. The Figure 3.35 describes the average value by incorporating different nodes as the node of origin. From the results, it has been observed that the DCT has smaller average hop count compared to the other two topologies.

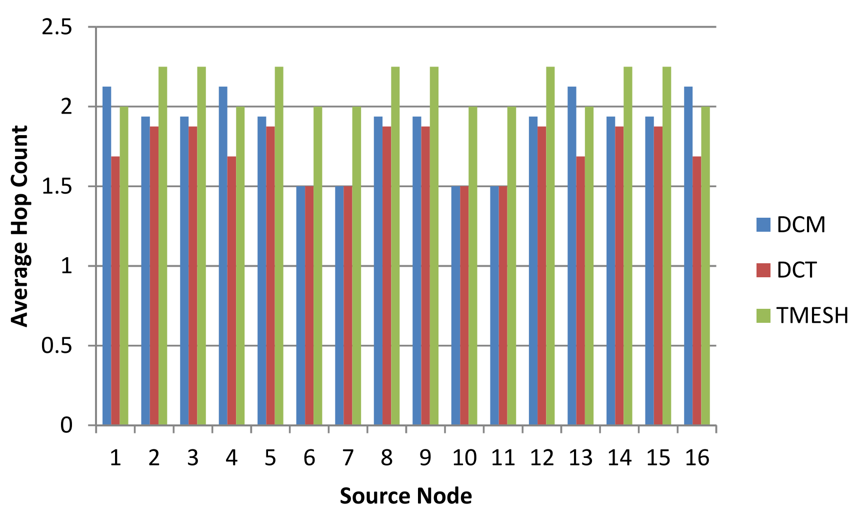

Figure 3.35: Average Hop Count of Each Node to Other Node in the Topologies

CHAPTER 4

OPTIMAL EXTRA LINKS PLACEMENT IN MESH INTERCONNECTION NETWORKS USING IMPROVED ENVIRONMENTAL ADAPTATION METHOD

4.1 Introduction

The mesh topology is applied in various supercomputers [8, 74] and parallel computers[75]. Recently, several variants of mesh have been suggested by the researchers with an increase in the high popularity of the topology. The most popular topologies are found to be torus, xtorus, xx torus, X torus, centre concentrated mesh and T Mesh, EMC^2 Mesh [35, 39, 41, 42, 45, 63, 65, 76–78] are the developed by adding the extra links to the mesh topology. These variants were designed with an objective to reduce the diameter or the average internode distance of the topology. The links introduced are based on the human perception, while there is a large scope in identifying the optimal link. The identification of optimal link is found to be complex with the iterative search being time-consuming. Therefore, heuristic search technique is considered as it is the most popular heuristic search like a genetic algorithm, particle swarm optimisation .However, from the recent research it was noticed that the Improved Environmental Adaptation Method (IEAM) performs better than these approaches[79].

4.2 Improved Environmental Adaptation Method

The genetic algorithm is considered as the most popular heuristic search algorithm which is inspired by Darwin's theory of survival of fittest. The genetic algorithm mimics the chromosomes of biological species to store the information of the individual. The genes of species are 74ategorized as genotype and phenotype. The genotype genes transfer the information from one generation to another. The algorithm focuses only on genotype genes as the offspring will represent the properties of parents that are either good or bad. The phenotype genes represent the changes that occur in species due to the environment in which they are living. The biologists consider these changes as adaptation, and the genes are found to reflect the specific properties like behaviour. IEAM uses the idea of adaptation and these genes are used to represent the spices which are adapted to the environmental conditions.

Further, it was observed that the species alters to adjust to the environmental conditions for their survival. The initialization of the IEAM is similar to genetic algorithm i.e. the random population may represent the solution to the particular problem. These solutions are either defined as good or bad. IEAM explore solutions similar to people having their role models in the specific fields. This model is found to follow them based on their role models. However, IEAM major question to answer is how the role model will behave? He has to behave on his own. The IEAM uses two operators to find the optimal solution. The first operator is the adaptation operator, and another one is the selection operator.

4.2.1 Adaptation Operator

Initially, the calculations of the upper and lower bound are performed according to the problem, which have been identified. The second most important task is to define the fitness function related to the problem. Let f_i represent the fitness of the individual in the population. Then equation (4.1) can be used to calculate the average value for the population.

$$f_{avg} = \frac{\sum_{i=1}^{n} f_i}{n} \tag{4.1}$$

In the above equation, 'n' is given as the population size or the number of individuals that makes the population. The adaptation is categorized as two types uncontrolled and controlled. The uncontrolled adaptation is for role model or leader in the population, and the controlled adaptation is considered for the followers of the leaders.

The Equation (4.2) represents the uncontrolled adaptation

$$P_{temp} = (\alpha P_i^{\phi} + \beta)\%2^L \tag{4.2}$$

Here 'P_i' accounts for the individual of the current population, 'α' and 'β' are the random numbers, 'L' represents the length of chromosome and 'ϕ' is the ratio of f_i with f_{avg}. For the uncontrolled adaptation, the population individual is represented by equation (4.3)

$$P_{temp} = (\alpha P_i^{\phi} + C\beta)\%2^L \tag{4.3}$$

Here C is the control variable and is calculated by the equation (4.4).

$$C = P_g - P_i \tag{4.4}$$

The 'P_g' in equation (4.4) accounts for the best individual, and 'P_i' accounts for the current individual. The binary chromosome is converted into the integer value, the encoding relation given by equation (4.5) can be employed.

$$Z = R_{\min} + \frac{M * (R_{\max} - R_{\min})}{2^{length}} \tag{4.5}$$

In the above equation (4.5), R_{max} and R_{min} are the maximum and minimum range of the search space in which the individual solution is to be searched. M is the integer equivalent to the binary bits and, length represents the total number of bits in the chromosome to represent the variable.

The performance of the algorithm is found to be greatly affected by the 'α' and 'β' as these parameters need to be selected carefully [79].

4.2.2 Alteration Operator

The alteration operator is similar to the single point mutation in which one bit is randomly changed from its original value. Just like a mutation operator in the genetic algorithm, the alteration operator is applied with the probability P_{alt}.

4.2.3 Selection Operator

The two populations such as current population and the population are generated after applying operator are mixed and sorted according to the fitness for the selection of the better results. The n best individuals are picked from the pool of 2n to represent next set of the individual.

4.3 Proposed Approach

Intailly, the search problem is encoded in the terms heuristic search algorithm for finding the solution to real world problem it should be.

4.3.1 Encoding

The mesh network is represented by the adjacency matrix. Several techniques have been used in the past to represent the network that requires fixed or variable length chromosome while the current objective is to identify the optimal extra links. The links can be easily represented by the two vertices. Thus, for a graph with *n* nodes, the numbers of bits required for representing a single node will be $\log_2 n$. Hence, the length of the chromosome will be given by $2\log_2 n$.

4.3.2 Fitness Function

The proper fitness function to find the optimal link in the topology is defined by an average internode distance, which has been described by equation (4.6).

$$f' = \frac{\sum_{i=1,j=1}^{i=n,j=n} D_{ij}}{n^2} \tag{4.6}$$

Here n is the total number of nodes in a mesh numbered from 1 to *n*. D_{ij} represents the distance between node I and j.

The problem is converted to minimization problem using equation (4.7) for simplicity.

$$f = D - f' \tag{4.7}$$

Here the *D* is considered as an average internode distance of mesh. For the computation of fitness value, the adjacency matrix will be updated using equation (4.8).

$$Adj_{new} = Adj_{mesh} + Adj_{update} \tag{4.8}$$

Adj_{mesh} *and* Adj_{update} represent the adjacency matrix of the mesh topology and update. The updated matrix is a zero matrix with 1 in the row and column corresponding to the link.

4.4 Proposed Algorithm

The Algorithmic steps are described below:

Link Generation ()

Input: Adj_{Mesh}, row, columns, gen_count, popsize

Output: Set of optimal Links

Step 1: Generate initial random Population of popsize

Step 2: Assign Fitness to the Population as defined

Step 3: Set Generation counter, i = 0

Step 4: Apply adaptation Operator and generate P_{temp}

Step 5: Merge P_{temp} and P_i to P

Step 6: Sort P, according to fitness value and select the top popsize element and copy them to p_{i+1}

Step 7: Increment Generation counter, i by 1

Step 8: if i<gen_count then goto step 4

Step 9: Stop

In the proposed algorithm, the first step is to encode the problem into the form of chromosome. As the problem stated is exploring the links in a topology. The pair of nodes (n1, n2) can represent the link in a topology. Each node has the unique id of type integer. For n nodes in topology these id lies in the range of zero to n – 1. In step 1, two random numbers are generated for the node 1 and node 2 within the range of zero to n-1. These values are then transformed into their binary equivalent and concatenate with each other to form of a chromosome of length $2\times\log_2(n)$. This single chromosome is going to represent a single individual in a population. To generate population of size (n) = popsize, the n individual are generated to form the population. In step 2 based on the phenotype of the chromosome the fitness value is evaluated using equation (4.6) to (4.8). In step 3 generation count is initialize to zero and will help in tracking the number of generation that have passed. As in IEAM, the first operator is Adaptation operators the random value of α, β are generated, and ϕ is estimated using from fitness of the individual and average value of fitness. Based on the fitness value a leader is selected, the leader is the best individual in the current generation. As the leader will have no role model and adapts in an uncontrolled manner. The uncontrolled adaptation is given by the equation (4.2). For the remaining population in the generation the individual will adapt according to the equation (4.3) and (4.4). The alteration operator will generate a temporary population of size equal to popsize by flipping the bit randomly and is referred as P_{temp}. In step 5 the current population is merged with temporary population and termed as P. As the P has twice the elements in comparison to popsize so in step 6, we have to select of half of the elements to do so the fitness function is evaluated first for newly generated individual and are sorted based on the fitness value. The top half of individual are marked as the P (i+1). This completes the generation of the population hence the generation counter is incremented by one in step 7. In step 8 stopping condition is tested and if it is true the algorithm will stop, as the final generation has already being generated, otherwise steps 4 to 8 repeated. After the completion of generations, the best results are picked, and the resultant topology is tested for performance on throughput and latency on various traffic patterns.

4.5 Results and Performance Evaluation

To test the performance of the system the parameters of the parameters of IEAM are selected as follows:

The value of α is adapted using the relation if $f_{avg}/f_0 >= 0.8$ then $\alpha = 0.8$ else $\alpha = 1$. Selecting the value of α using the above rules helps in maintaining the diversity in solution. The value of β is selected randomly in the range of 0 to 255. The probability of alteration is set to 0.3.As the results are compared with simple genetic algorithm. The value of P_c and P_m are selected as 0.8 and 0.3 respectively. Both the algorithms are executed with population size of 20, 30, 40, and 50. The maximum numbers of generations allowed in both algorithms are set 100.

The results are generated from both the Genetic algorithm and IEAM. Table 4.1 presents the results of IEAM for population size 30 and the generation count of six. Vertex A and Vertex B are the representing the id of nodes numbered from 1 to 16 in 4 × 4 mesh

Table 4.1: Describing IEAM results for 4 × 4 mesh

Vertex A	Vertex B	Average internode distance
5	**12**	**2.541667**
8	**9**	**2.541667**
3	**14**	**2.541667**
14	3	2.541667
2	**15**	**2.541667**
3	14	2.541667
12	5	2.541667
2	15	2.541667
1	16	2.554167

1	16	2.554167
4	13	2.554167
14	8	2.566667
3	16	2.566667
6	16	2.566667
2	13	2.566667
8	1	2.566667
3	16	2.566667
9	3	2.566667
1	8	2.566667
14	8	2.566667

The first four unique links are picked and the different links are highlighted in Table 4.1. Figure 4.1 represents the topology that is generated by adding the links to the mesh topology of 4 × 4

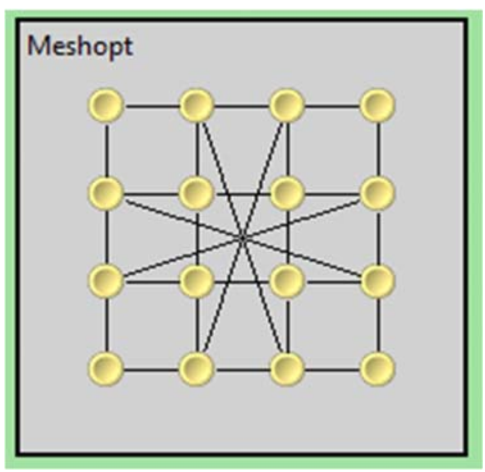

Figure 4.1: Resultant Topology 4×4 with Optimised Link

Table 4.2 accounts for the results obtained from the genetic algorithm. The genetic algorithm with the population size of 30 is found to obtain the same results after 44 iterations. To compare the search capability of both the approaches various test are performed at the different population size and generation counts. The results showing the comparison of the same has been presented in Table 4.3. Moreover, Figure 4.2 compares the results, and it has been observed that IEAM functions faster in comparison to the genetic algorithm. Figure 4.2, represents the result, which indicate the iteration at which the optimal results are obtained. The case in which, the optimal results are not obtained, then the maximum number of iterations that is 100 is recorded. From the figure, it was observed the Genetic algorithm fails to search the desired results for the population size of 20 and 40. The genetic algorithm is found to be significantly improvised at the population size of 50 as the population size covers the one-fifth of the search space, which is not an efficient approach and is adding the complexity in comparison to the iterative search technique.

To analyse the results obtained the topology is generated on the omnet++ simulator [26], [27] and the performance is compared with the existing topologies like mesh, C^2mesh and Tmesh. The parameters used for testing the topology are described in Table 4.4.

Table 4.2: Describing genetic algorithm results for 4×4 mesh

Vertex A	Vertex B	Average internode distance
12	**5**	**2.541667**
8	**9**	**2.541667**
14	**3**	**2.541667**
8	9	2.541667
2	**15**	**2.541667**
14	3	2.541667
8	9	2.541667
3	14	2.541667
15	2	2.541667
14	3	2.541667
8	9	2.541667
8	9	2.541667
8	9	2.541667
8	9	2.541667
14	3	2.541667
8	9	2.541667
2	15	2.541667
3	14	2.541667
2	15	2.541667
3	14	2.541667

Table 4.3: Describing performance of two algorithms with different population size

Population Size	IEAM	Genetic Algorithm
20	15	**100**
30	6	44
40	7	**100**
50	3	5

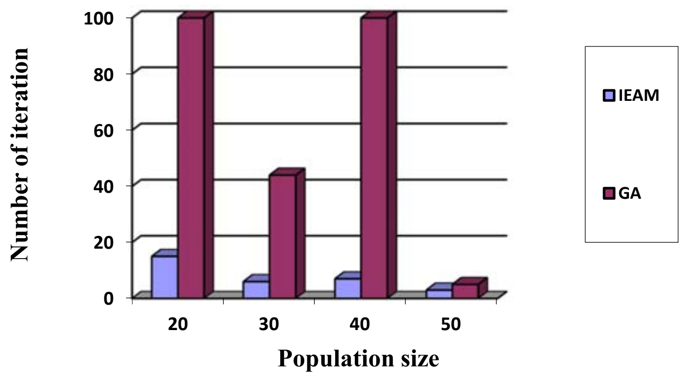

Figure 4.2: Comparison of the Performance of the Two Approaches

Table 4.4: Describing the parameters of the topology in omnet++

S.No.	Parameter Name	Value
1	Number of rows	4
2	Nodes in each row	4
3	Packet size	1024 bytes
4	Data rate	1 Gbps
5	Simulation time	10 ms
6	Warm up time	0.5 ms
7	Networks	Mesh, T Mesh C^2 Mesh IEAM Mesh
8	Inter node packet delay	163.84 µs to 4.09 µs
9	Number of runs	5

4.5.1 Performance of networks on Uniform Traffic

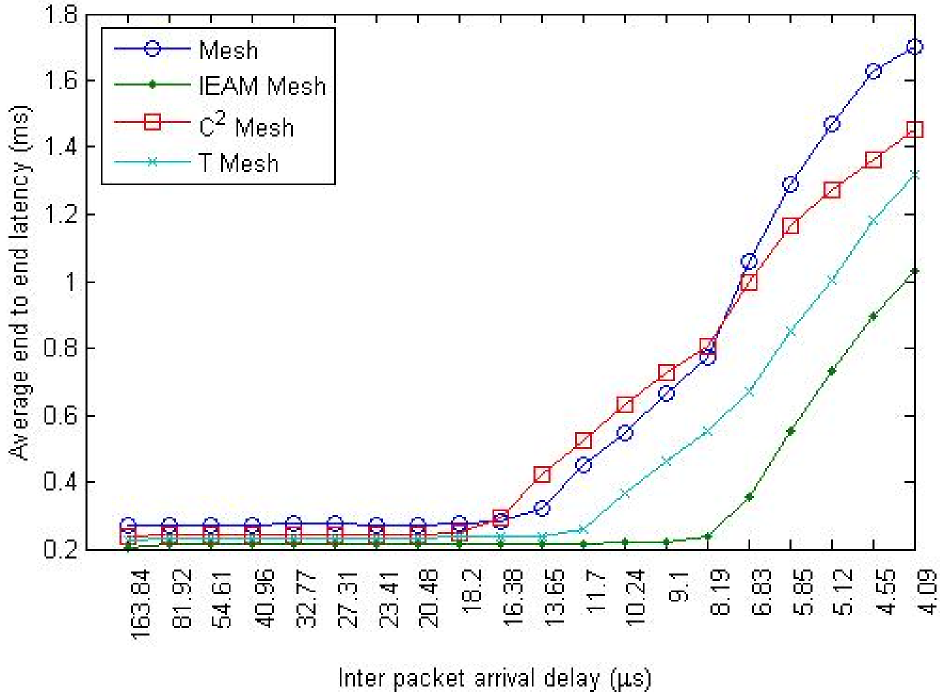

Figure 4.3: Average Latency versus Inter Packet Arrival Delay on Uniform Traffic

Figure 4.3 compares the average latency of the packets on different loads by varying the inter-packet arrival delays. From the figure, it can be observed that the IEAM has a minimum latency difference which is clearly visible at the inter arrival packet delay of 11.7 µs. Figure 4.4 also compares the throughput on the uniform traffic and IEAM Mesh that has the highest throughput among all the four topologies from the inter arrival packet delay of 11.7 µs.

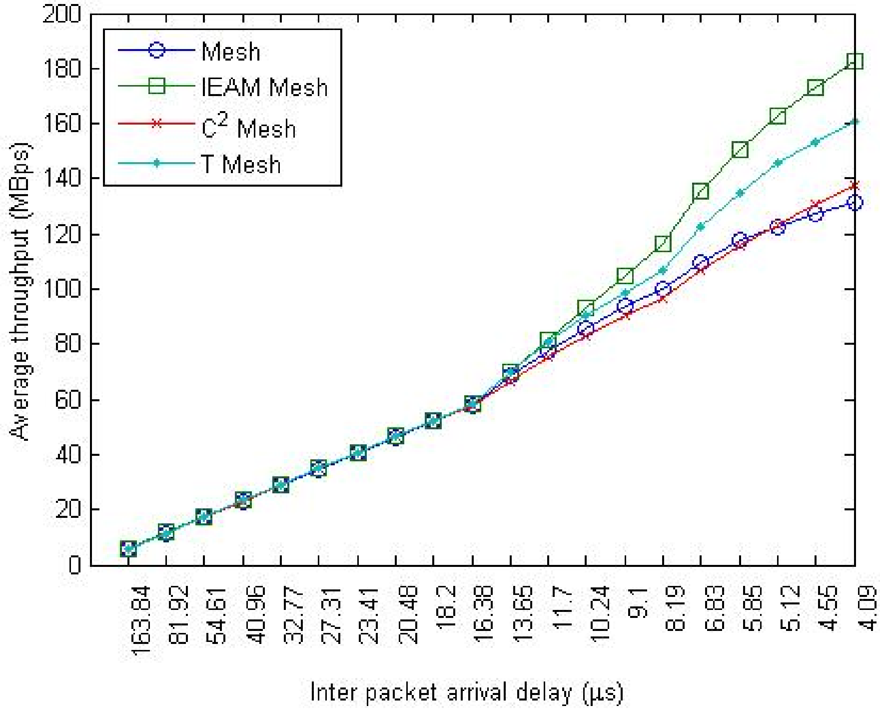

Figure 4.4: Average Throughput versus Inter Packet Arrival Delay on Uniform Traffic

4.5.2 Performance of Networks on Bit Complement Traffic

Figure 4.5 compares the average latency of all the four topologies, and the IEAM Mesh has the lowest latency on all inter-packet arrival delays. Figure 4.6 shows that very high throughput is achieved for the IEAM Mesh in comparison of the other three topologies after inter arrival packet delay of 32.77 µs.

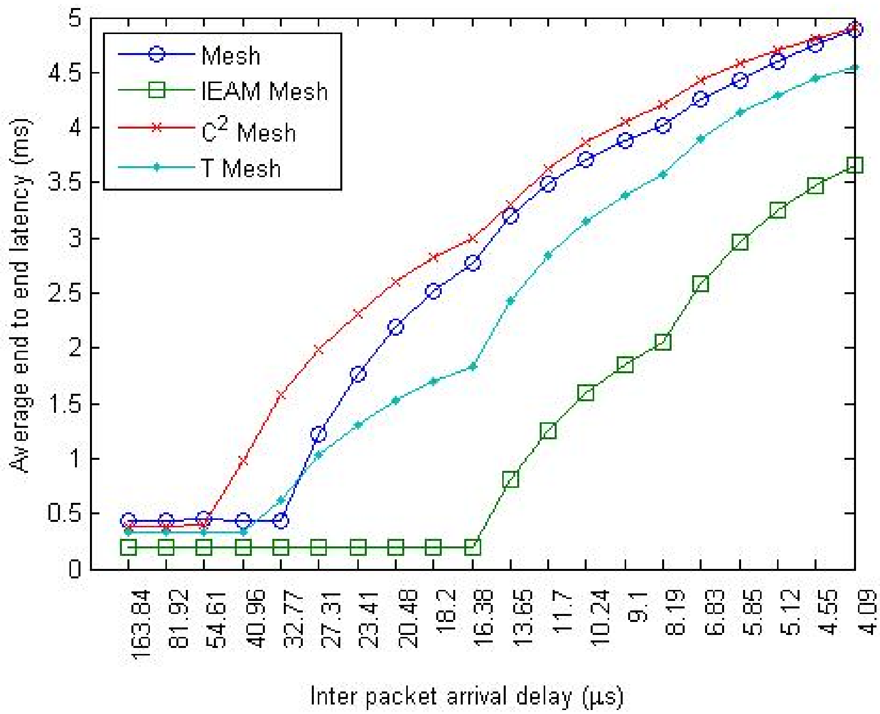

Figure 4.5: Average Latency versus Inter Packet Arrival Delay on Bit Complement Traffic

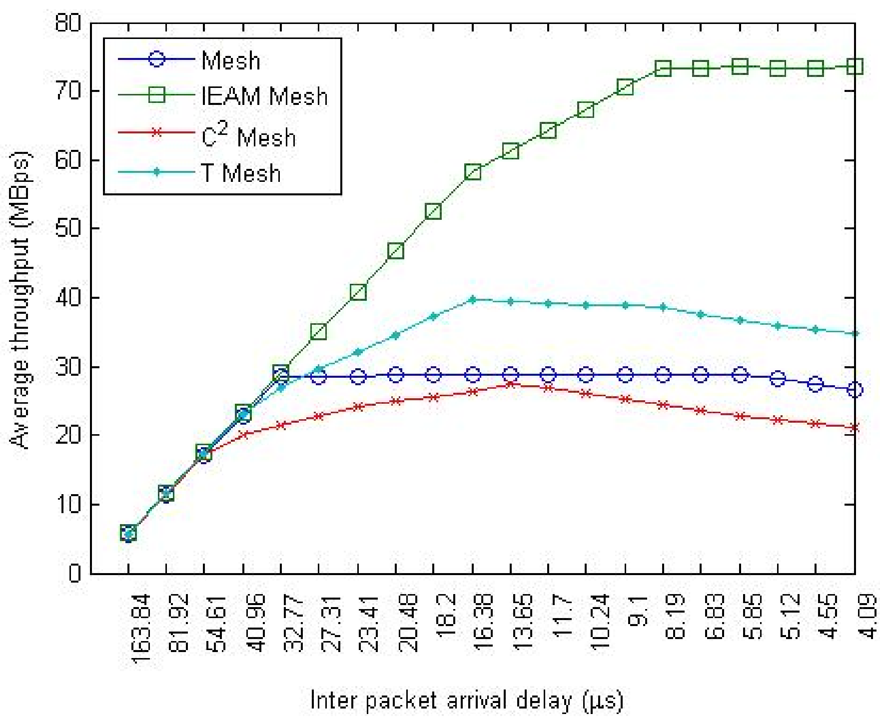

Figure 4.6: Average Throughputs versus Inter Packet Arrival Delay on Bit Complement Traffic

4.5.3 Performance of Network on Tornado traffic

In tornado traffic, the packets are sent to the destination which is present at least at half of the distance from the source. These nodes can represent either half the distance in *X* dimension or *Y* dimension, but presently it is considered both in *X* and *Y* dimension. The equation (4.9) is used to generate the tornado traffic.

$$D_{id} = \left(S_{id} + \frac{c}{2} + c * \frac{r}{2} \right) \% n \tag{4.9}$$

Here *c* and *r* represent the rows and columns respectively, and a total number of nodes in the mesh is represented by *n*.

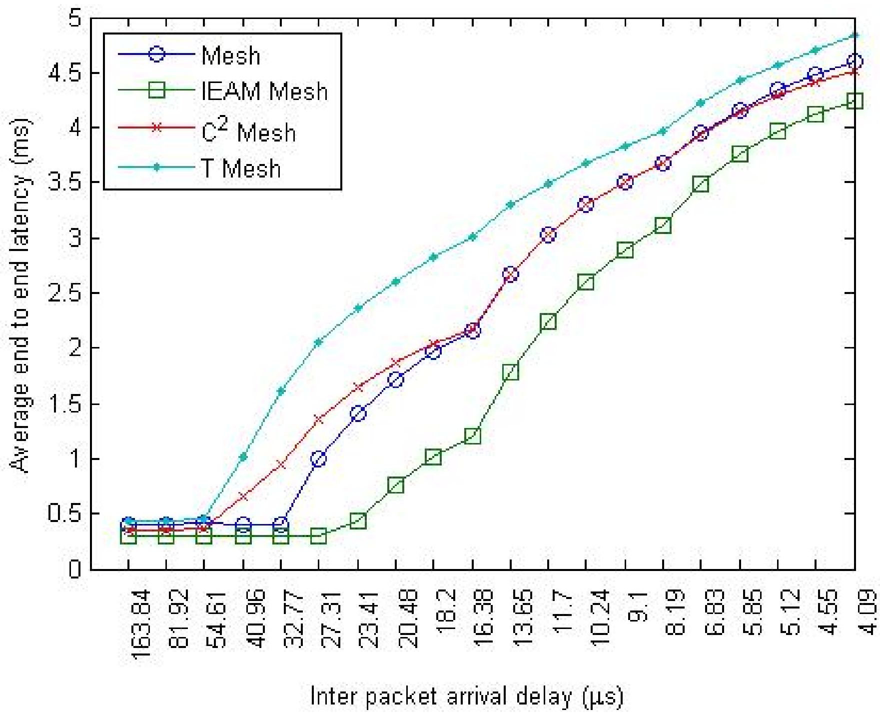

Figure 4.7: Average Latency versus Inter Packet Arrival Delay on Tornado Traffic

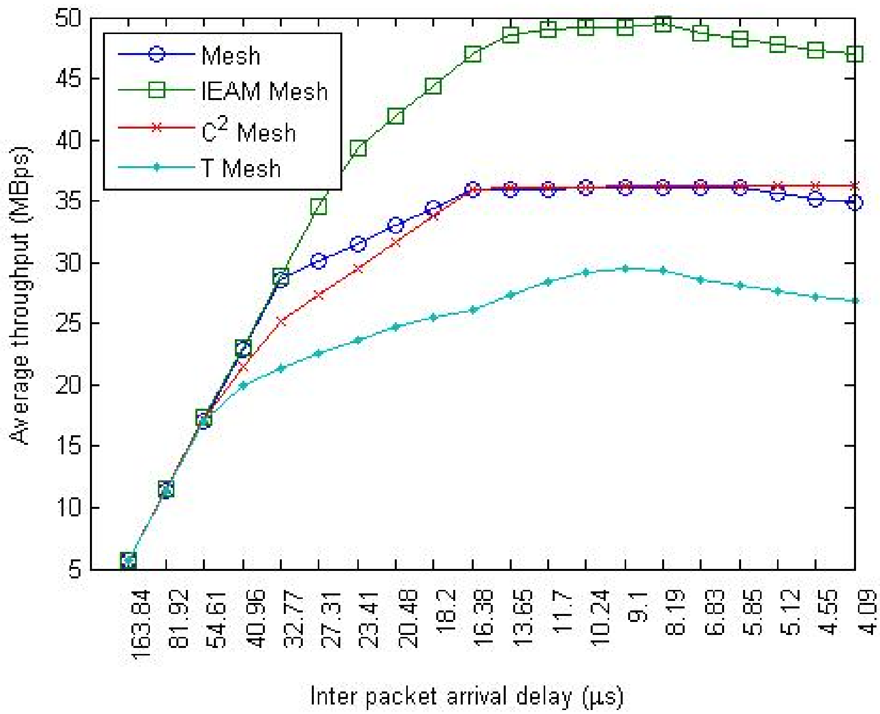

Figure 4.8: Average Throughput versus Inter Packet Arrival Delay on Tornado Traffic

Figure 4.7 shows that IEAM Mesh is having low latency at the inter packet arrival delay of 27.31 µs and Figure 4.8 again shows the performance of IEAM Mesh is higher from the inter arrival packet delay of 27.31 µs and higher.

4.5.4 Performance of Network on Neighbour traffic

To analyse the performance of the topology the diagonal neighbour is considered as a destination as compared to horizontal and vertical neighbours. Again from the Figure 4.9 it can be observed that the Average latency of IEAM Mesh is very less even at higher loads and Figure 4.10 shows that the throughput of IEAM Mesh is slightly higher than other topologies from the inter packet arrival delay of 10.24 µs and lower.

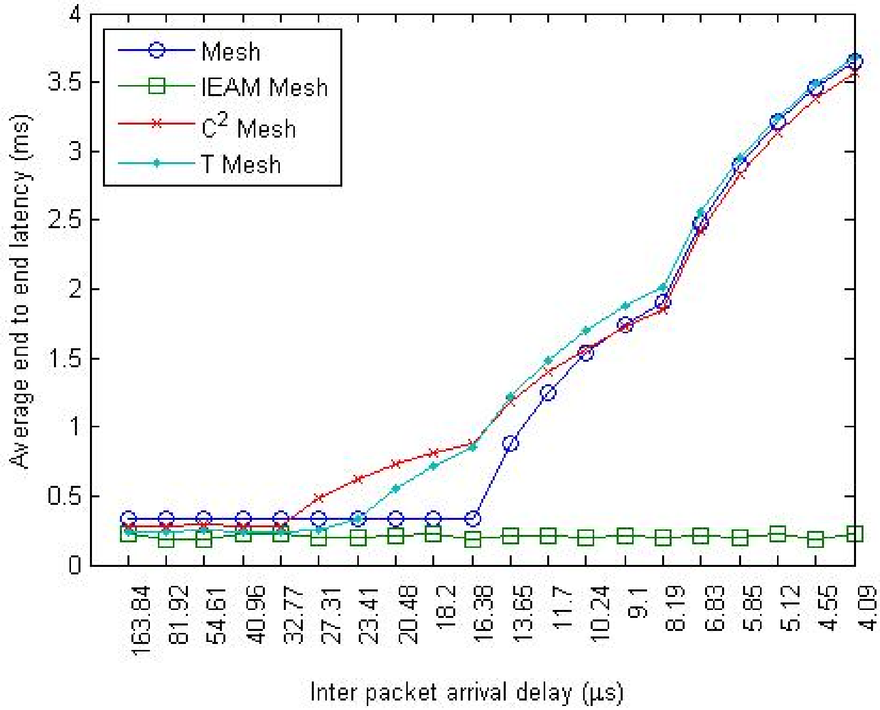

Figure 4.9: Average Latency versus Inter Packet Arrival Delay on Neighbour Traffic

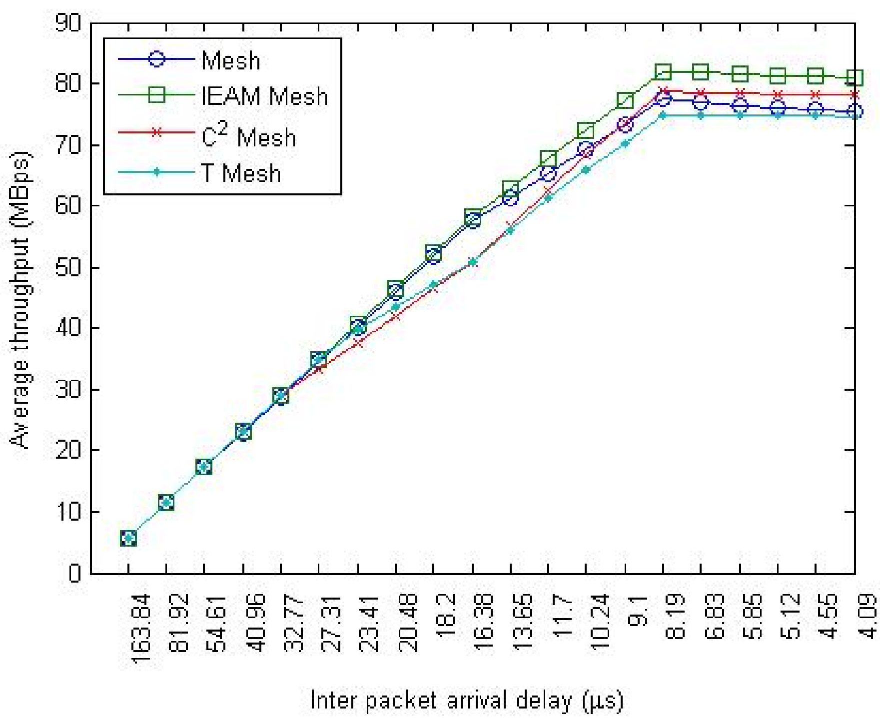

Figure 4.10: Average Throughput versus Inter Packet Arrival Delay on Neighbour Traffic

4.5.5 Performance Metric

The performance metrics or power of a network is defined as the ratio of average throughput, and average latency of the packets received [31]. Equation (4.10) represents the power of a network.

$$P = \frac{Average\ throughput}{Average\ latency} \tag{4.10}$$

Table 4.5 describes the performance metric for all networks. From Table 4.5 it can be observed that highest performance metric (power of network) is achieved by IEAM Mesh in the comparison to the other 3 topologies. The table 4.5 has the best performance metric of each topologies highlighted in bold. For the Uniform traffic the performance metric of IEAM is 468.08 in comparison to mesh with 213.34, c2 mesh with 210.04 and TMesh with 311.95.Based on the comparison between the best and second best there is an improvement of 1.5 times. Similarly, for bit complement traffic IEAM is having performance metric as 306.00 in comparison to mesh, c^2 mesh, and TMesh having the values as 64.96, 43.34, and 70.29 respectively. On comparison between the performance of IEAM and TMesh again, an improvement of 4.35 has been noted. In case of Tornado traffic the highest performance metric for mesh, IEAM mesh, c^2 mesh, and Tmesh are 69.91, 114.81, 47.30, and 38.06 respectively. The observation revels that the IEAM is having the highest performance metric and is 1.6 times better than mesh topology. For the neighborhood traffic, the performance metric of IEAM mesh is 425.36 and is to 2.5 times better than the mesh with 168.97. Another major observation that can be drawn from the table is that the performance metric of IEAM always achieved the high performance metric at the lower Inter packet arrival delay, which means that the performance of the IEAM mesh is not affected more in comparison to the other topologies. However, from the performance metric, it can be seen that the best of each topology occurs at different loads based on the inter-packet arrival delay. Now to compare the overall performance of these topologies the best value of performance metrics highlighted in the table is picked. The graph described the comparison of best performance metrics in Figure 4.11. Figure 4.11 highlights that IEAM Mesh has best performance of metric in comparison to other topologies.

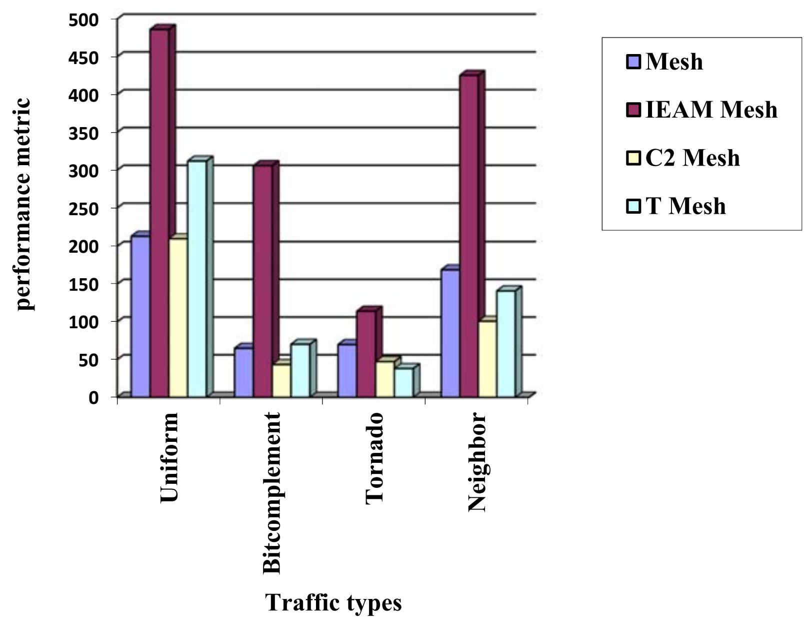

Figure 4.11: Describing the Performance Metric of the various Topologies on Different Traffics

Table 4.5: Describing the performance metric of the different topologies on various traffics

Inter-packet Arrival Delay (µs)	Uniform Traffic				Bit complement traffic				Tornado Traffic				Neighbour traffic			
	MESH	IEAM-Mesh	C2 Mesh	T MESH	MESH	IEAM-Mesh	C2 Mesh	T MESH	MESH	IEAM-Mesh	C2 Mesh	T MESH	MESH	IEAM-Mesh	C2 Mesh	T MESH
163.84	21.43	28.52	24.37	25.62	13.02	30.68	14.79	17.50	13.98	19.29	16.08	12.99	16.79	25.84	20.10	23.53
81.92	42.46	54.85	47.82	50.70	26.09	61.49	29.65	35.14	27.99	38.67	32.22	25.98	33.82	61.33	40.34	47.42
54.61	64.07	82.39	72.19	75.20	38.34	91.28	**43.34**	51.67	41.17	57.09	**47.30**	**38.06**	50.74	92.22	60.28	70.51
40.96	85.57	110.15	96.08	100.67	52.22	123.24	20.40	**70.29**	56.06	77.31	32.87	19.66	67.69	104.10	80.72	94.84
32.77	105.70	135.32	119.31	124.90	**64.96**	154.04	13.63	43.03	**69.91**	96.58	26.39	13.35	84.72	128.93	**100.80**	118.77
27.31	126.61	163.37	142.60	150.40	23.48	182.38	11.56	28.72	30.22	**114.18**	20.14	10.99	101.47	181.77	68.69	**141.04**
23.41	149.34	192.04	168.23	176.31	16.22	215.82	10.48	24.43	22.46	91.62	17.92	9.98	118.54	200.62	60.28	120.44
20.48	171.11	219.41	191.27	201.84	13.11	245.51	9.58	22.59	19.17	55.51	16.98	9.47	135.50	222.07	57.52	79.33
18.20	188.33	243.46	**210.04**	223.27	11.42	273.68	9.06	21.79	17.49	44.05	16.61	9.08	152.19	234.70	57.12	65.65
16.38	204.06	271.81	195.28	245.37	10.37	**306.00**	8.75	21.58	16.61	38.84	16.56	8.69	**168.97**	307.60	57.80	59.60
13.65	**213.34**	325.24	157.81	294.34	9.03	75.56	8.30	16.25	13.46	27.31	13.47	8.28	69.99	295.84	47.96	46.02
11.70	171.74	379.26	144.33	**311.95**	8.27	51.27	7.39	13.78	11.86	21.92	11.90	8.16	52.15	329.56	44.79	41.43
10.24	157.84	429.00	131.06	245.16	7.78	42.45	6.71	12.37	10.90	18.81	10.96	7.93	45.33	356.62	43.47	38.72
9.10	141.34	479.08	124.15	212.75	7.44	38.11	6.21	11.45	10.26	17.02	10.31	7.70	42.14	367.92	42.59	37.50
8.19	129.29	**486.08**	119.61	193.99	7.18	35.72	5.85	10.80	9.79	15.87	9.87	7.37	40.59	402.73	42.48	37.10
6.83	103.61	379.64	106.75	183.47	6.79	28.42	5.35	9.63	9.12	13.96	9.20	6.79	30.97	388.22	32.44	29.26
5.85	91.35	271.65	99.19	159.02	6.51	24.75	4.99	8.87	8.67	12.81	8.76	6.36	26.36	424.54	27.69	25.34
5.12	83.23	223.53	96.48	145.67	6.12	22.60	4.72	8.36	8.23	12.05	8.45	6.05	23.69	360.69	24.95	23.03
4.55	78.16	193.02	95.89	129.65	5.74	21.15	4.51	7.95	7.87	11.49	8.22	5.79	21.93	**425.36**	23.15	21.43
4.09	77.37	177.05	94.68	121.83	5.42	20.12	4.33	7.64	7.57	11.09	8.03	5.57	20.67	360.73	21.87	20.30

CHAPTER 5
ROUTING ALGORITHMS IN THE MESH TOPOLOGIES

5.1 Introduction

The significant challenge observed in the interconnection network is the use of routing algorithm. The performance of the network is greatly affected by the routing algorithm used. If the path selection by the routing algorithm is not the shortest path and there is no knowledge of traffic on the selected path, then the performance of the interconnection network will be questioned. The routing algorithm is found to be unsuitable on using high memory, in the router on chips. The number of comparison and the complexity of arithmetic operations is also responsible for the performance or complexity of the router. This chapter proposes an new deterministic routing algorithm which focusses on reducing the execution time and space complexity of the level based routing algorithm. Another major discussion in this chapter on the importance of selecting the shortest path to reach to the particular destination the same has been highlighted in the modified center concentrated mesh routing algorithm.

5.2 Router Architecture

The most common type of routing algorithm used in the networking makes the decision at the source, but they are not suitable for the network on chips, as the routing tables are needed to maintain at each node. This will increase the area of the router, which is a prime factor while designing the network on chip. The state-based routing algorithms are able to take the decision based on the current states, which are deterministic in nature. For designing a mesh topology, a five-port router is used. The detailed overview of the router architecture has been described in Figure 5.1. In Figure 5.1, it was identified that the router comprises of five ports out of which the four ports are connected to other routers, and one port is present at a local port that is attached to the processing element. The number of decision in the case of the router used for the mesh topology is limited to only 5 ports that are the next state of the system can be any of the five ports east, west, north, and south or to the local port. These ports can be labelled according to the various conventions either by the own labels or by the labels of that are assigned to the node of the adjacent router to which a particular node is connected. In case degree of router increases, the number of ports in the router will increase and in turn will increase the number of states in the routing algorithm.

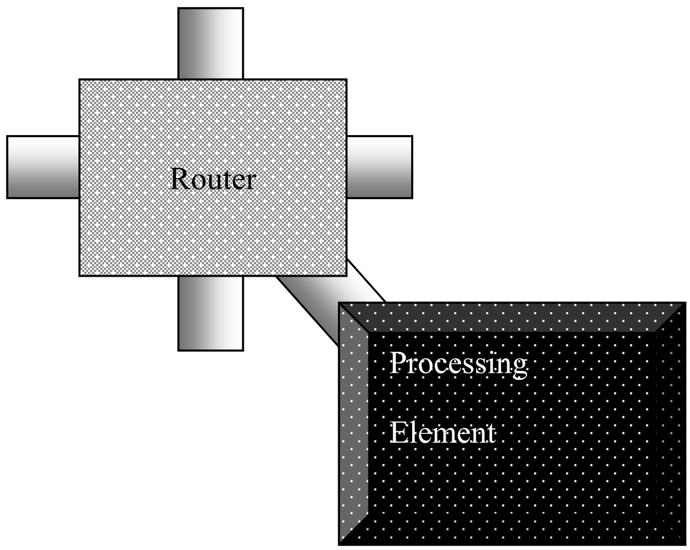

Figure 5.1: Router Architecture for the Mesh

5.3 Routing for Center Concentrated Mesh

To study the drawback of CCM[35] routing algorithm first different type of nodes are to be identified. In CMM routing algorithm the nodes can be classified into three categories:

1. Corner nodes: The four nodes that are at four corners of the topology are corner nodes.
2. Center nodes: The node that behaves as center of the topology are defined as the center nodes. The topology can have one, two, or four center nodes.
3. Elementary nodes: All the nodes that do not behave as corner or center nodes are called as elementary nodes. The relation described in equation (5.1) determines the total numbers of elementary nodes.

$$Node_{Elemetary} = TotalNodes - \left(Nodes_{Corner} + Nodes_{Center}\right) \tag{5.1}$$

The packet from source to destination will follow its journey in three parts each having sub source and sub destination. These sub-sources and destinations are

1. Source to sub-mesh center (*SC*): Initially packet is routed from the source to sub-mesh center node.
2. Source sub mesh center to the destination sub mesh center (*MM*): The packet is routed from respective sub-mesh center to the destination sub-mesh center.
3. Sub mesh center to the destination (*CD*): The packet is routed from destination sub mesh to the target node.

The total number of hops can be represented by equation (5.2).

$$H_{Total} = H_{SC} + H_{MM} + H_{CD} \tag{5.2}$$

The routing algorithms have few challenges and need to be addressed to answer:

1. Is the path selected from the source to destination is the shortest path?
2. If the path chosen to balance the load or creates a hotspot affect?

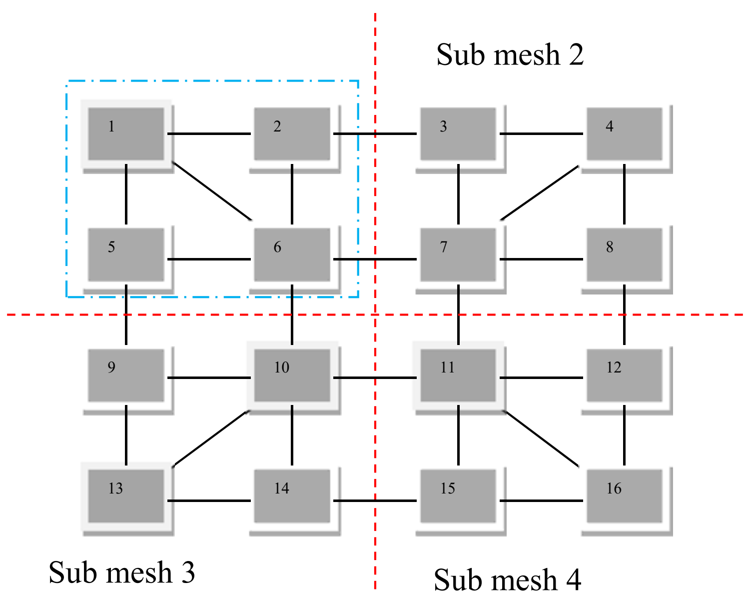

Figure 5.2: Describes the various Sub-mesh in 4 × 4 C^2 Mesh

To answer these questions, consider the center concentrated mesh described in Figure 5.2. Assuming the node 1 as source and node 2 as a destination; the will be sent to the center node which at 1 hop distance. Now as the center are same, so there is no need to change center so hop count H_{mm} is 1. To send the packet from center to the destination again 1 hop will be required. Therefore, the total hop H_{Total} = 1+0+1 = 2 as per the routing algorithm. Ideally, it can be routed directly with the link between 1 to 2 in a single hop. This approach is utilized to answer the initial question based on the selection of the route selected by the routing algorithm, which is not optimal.

As all the nodes are sending the packet to the center nodes, there is found to be a hot spot effect on the center node, and from the example it is evident, the traffic is diverted to center node even though there exists the direct path.

5.3.1 Modified Center Concentrated Routing (MCCM) Algorithm

The proposed routing algorithm assumes the router is simple and has limited memory and compute capabilities.

Algorithm 5.1: Modified Center concentrated Mesh Routing (MCCM) Algorithm

INPUT:	Coordinates of Source and Destination (S,D)
Output:	Port Number for Next destination
Step 1:	Evaluate Center Node Coordinates
	If n is even **then**
	C1← (n/2 -1, n/2-1)
	C2← (n/2, n/2-1)
	C3← (n/2 -1, n/2)
	C4← (n/2 , n/2)
	else
	C1 ←C2←C3←C4←(n/2 , n/2) // There will be only single center node
	End if
Step 2:	E1←(0,0)
	E2← (0,n-1)
	E3←(n-1,0)
	E4←(n-1,n-1)
Step 3:	Get mesh id of source and destination say s,d

Step 4:	Dxy (sx,sy,dx,dy) (dx-sx)+(dy-sy) // macros definition
	Dis← Dxy(Sx,Sy,Dx,Dy)
	DCMM ← Min(Dxy (S,Cs), Dxy (S,Es)+1) +Dxy(Cs, Cd)+ Min(Dxy(D, Cd), Dxy (S,Ed)+1)
Step 5:	**If**(Dxy < = DCCM)
	Port =XY(S,D)
	Else
	Port =CCM(S,D)
	End If
Step 6:	**END**

Initially, the center coordinates *C1, C2, C3, C4* and corner coordinates E1- E4 of the mesh are evaluated in Step 1 and Step2. In Step 3 mesh id is assessed using algorithm 2. In Step 4 distance based on XY routing algorithm and CCM routing algorithm is computed. In step 5 the shortest path is selected, and port is assigned. The exact algorithm can be identified from the various sources as the *XY* routing is considered as the most popular routing algorithm [8]. The *CCM* routing has already been proposed by the [66, p. 2]. If the port number is in the range of 0-4, then the packet is routed to the core or in four directions, but if the port number is 5, then the packet should be routed to the center or to the corner node based on the position of the current node.

Algorithm 5.2: For Sub-Mesh identification for the Specific node

INPUT:	Coordinates of node(x,y)
Output:	Port Number for Next destination
Step 1:	C← N/2
Step 2:	If (x>0 and y>0 and x<=C-1 and y<=C-1 then
	ID ←1
	End if
Step 3:	If (x>C-1 and y>0 and x<=N-1 and y<=C-1 then
	ID ←2
	End if

Step 4: If (x>0 and y>C-1 and x<=C-1 and y<=N-1 then

ID ←3

End if

Step 5: If (x>C-1 and y>C-1 and x<=N-1 and y<=N-1 then

ID ←4

End if

Step 6: END

5.3.2 Testbed for Testing the Modified Center Concentrated Mesh Routing Algorithm

The Routing Algorithm has been tested using the Omnet++ simulator[54,55] and using HNOCS[83] the parameters utilized for the testing of the routing algorithm on the center concentrated mesh are described in Table 5.1

Table 5.1: Describes various parameters used during the experimental setup

S. No.	Parameter Name	Parameter value
1	Rows	4
2	Columns	4
3	Simulation time	2ms
4	Warm up time	240 ns
5	Message length	4 Packets
6	Packet length	8 Flits
7	Flit size	4 Bytes
8	Traffic type	Uniform
9	Maximum queued packet	4
10	Channel Bandwidth	8 Gbps
11	Number of runs	5

5.3.3 Results and Discussion

The results have been classified into the four categories.

5.3.3.1 End to End Latency

Figure 5.3 compares the four topologies at load factors ranging from 0.1 to 1.0. Till the load factor of 0.5 the performance of the MCCM and CCM has the small difference in latencies but at higher loads, the CCM routing algorithms have a sharp increase in the latency. Further, the odd-even routing is found to have higher latency than the MCCM routing algorithm. The table-based routing algorithm has the same latency to MCCM, but the memory required by table-based routing algorithm is high which makes its practical application infeasible.

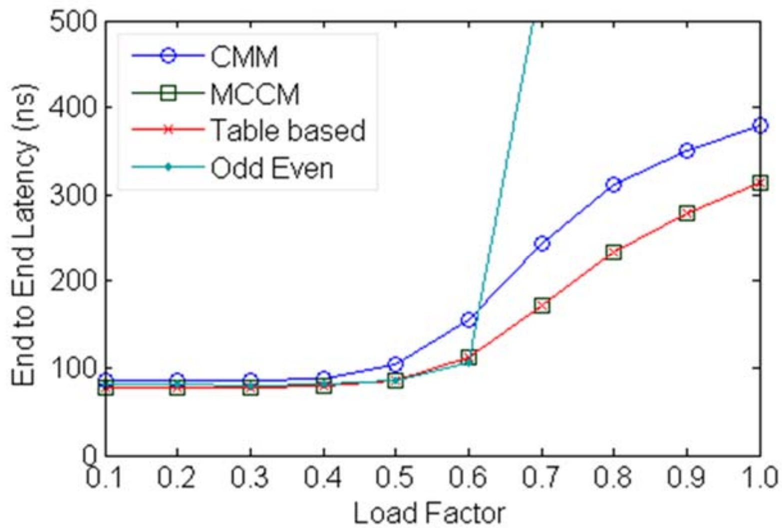

Figure 5.3: Describes the End to End Latencies at different Load factors

5.3.3.2 Bandwidth

Figure 5.4 compares the average sink bandwidth of the routing algorithm on c^2 mesh topology. Again, from the graph, it can be observed that the sink bandwidth of all the four routing algorithms is almost similar to the load factor of 0.6. The odd-even routing algorithm has shown the slight higher sink bandwidth at the load factor 0.7, but later MCCM and Table based routing algorithm have shown the improvement at the load factor of 0.7 to 1.0.

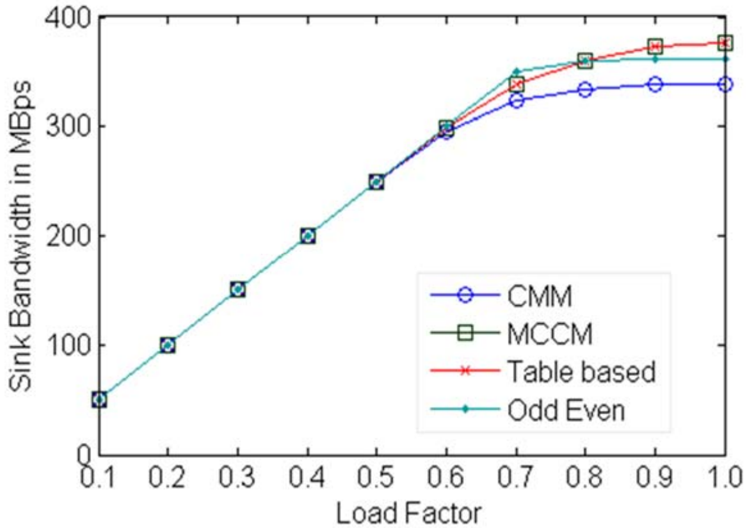

Figure 5.4: Sink Bandwidth at various Load Factors

5.3.3.3 Loss Probability

Loss Probability is the ratio of total number of packets lost to the total number of packets generated[84].

$$P_{loss} = 1 - \frac{\left(P_{queued} + P_{received}\right)}{P_{gen}} \tag{5.3}$$

Figure 5.5 compares the buffer loss probability of the various routing algorithms. From the figure, it can be observed that the loss probability of the all the routing algorithm is same as the initial loads the network is not loaded so. As the load increases, the network starts to get loaded the loss probability of the CCM routing algorithm and odd even routing algorithm has grown at a faster rate in comparison to the MCCM and table-based routing algorithm. The loss probability of the odd-even routing algorithm is higher than all the other routing algorithms after the load factor of 0.6.

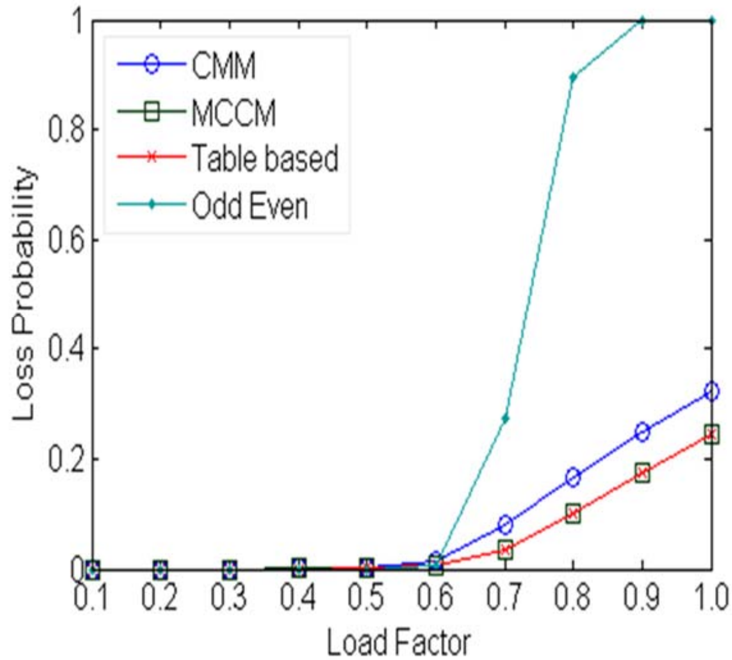

Figure 5.5: Describes the Loss Probability at the various Load Factors

5.3.3.4 Hop Count Analysis

Hop count analysis is the number of hops that are used to reach from the source to destination. The estimation of the hop count can be easily done with the help of Dijkstra's Algorithm[85] but in the case of routing algorithm hop count analysis is done on the basis of the routing algorithm. From Figure 5.6 it can be observed that the hop count required for CCM routing algorithm are more in comparison to other routing algorithms. The average hop count analysis for all the four routing algorithms on every node is performed as the routing algorithm will select the different paths and Figure 5.7 shows that CCM routing algorithm has large hop count in comparison to other three routing algorithms. The table-based routing algorithm has high memory consumption of O(n^2) which is constant in the case of the MCCM routing algorithm.

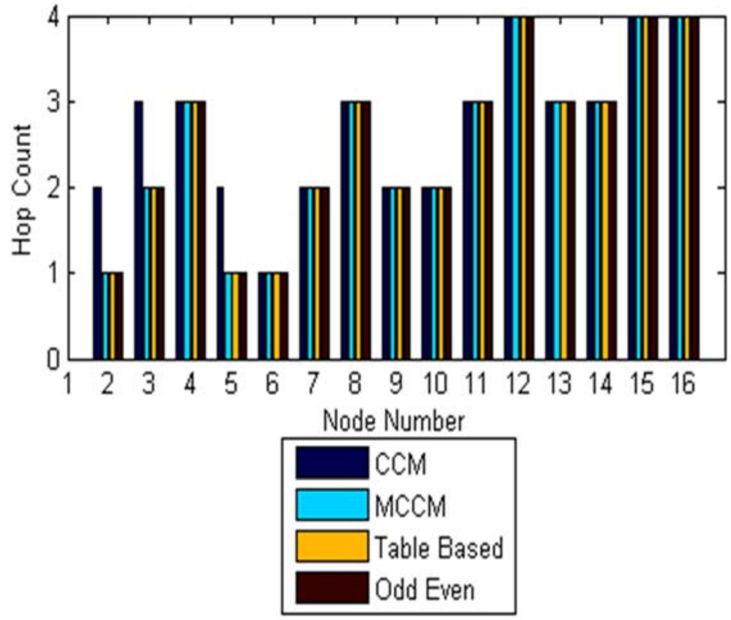

Figure 5.6: Describing the Hop Count from Source Node 1 to various Other Nodes

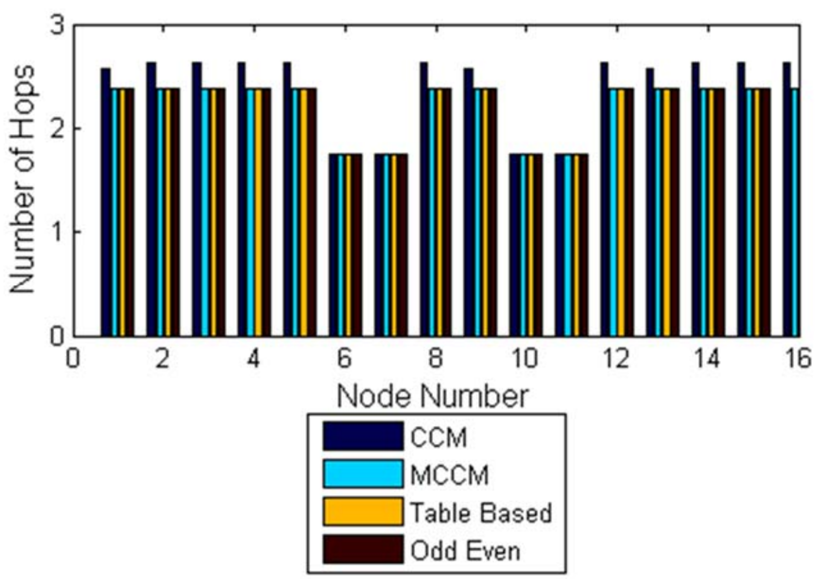

Figure 5.7: Describing the Average Hop Count by Considering Each Node as Source to Other Nodes

5.4 Level based Routing Algorithm using Dynamic Programming

The Level Based routing using Dynamic Programming (LBDP) algorithm uses the idea of reducing the execution cost by reducing the redundant computations that have been computed earlier. In the level-based routing algorithm, computations of current node level and destination nodes levels are required. The calculation of current node level can be calculated only once and used again for all the incoming packets as it depends on the router coordinates not on the packet. This will reduce the time consumed by the routing algorithm. This has been done by replacing current node level with a constant at the time of initialization of the router. It has been observed that in level-based routing algorithm uses the long node addresses and information about the local ports are missing these can be reduced by replacing them with short addresses.

Algorithm 5.5: Level based routing using dynamic programming

Input: Source ID, destination ID and level assigned at the time initialization		
Output: Destination Port		
	Cost	**Time**
int LBDP_routing(int curr_node,int dest_node)		
{		
Step 1. int level2=dest_node/n;	C_6	1
Step 2. **if**(level1==level2)	C_2	1
{		
Step 3. **if**(curr_node<dest_node)	C_2	1
return East;	C_3	1
Step 4. **if**(curr_node>dest_node)	C_2	1
return West;	C_3	1
else		
return local;	C_3	1
}		
Step 5. **if**(level1<level2)	C_2	1
return South;	C_3	1
else		
Step 6. return North;	C_3	1
}		

5.4.1 Proof of correctness

To study the correctness of algorithm, the working of the algorithm can be broadly categorized into two cases. The details of the two cases are as follows.

Case 1:

When current and destination addresses are at the same level, we can have three sub-cases

1. The current address is greater than destination address: The packet is directed to the west port according to the topology. Besides, the nodes on the left of the current node are found to have the address less than to the current node.

2. The current address is less than destination address: The packet is directed to the east port of the node and to the right of the current node at a particular level is found to be always greater.

3. The current address is equal to destination address: As the current address and destination address are same, this implies that packet is at the destination node and should be routed to the local port.

Case 2:

When current address and destination address are not at the same level, we have two sub - cases

1. Current node level is less than that of destination level node: The packet is directed to the south port as the nodes in the mesh in the downward direction are always at a higher level than nodes at the lower level.

2. The current node is at a greater level than the destination router: The packet has to be routed to the northward node as node above the given node will always at the lower level.

As both the cases are working properly so it can be inferred that the routing algorithm is correct.

5.4.2 Performance Analysis

The performance of any algorithm depends on the factors such as the execution time and Memory consumption of the algorithm.

5.4.2.1 **Comparison based on the execution time**

The execution time of the Level Based (LB) routing algorithm is compared with LBDP routing algorithm. The routing algorithm is called N number of times to analyse the performance of the routing algorithm. This is performed to identify the difference in the two routing algorithms because the single execution of the algorithm has very small execution time and is difficult to record on a machine. The results are obtained by the hardware configuration as described in

Table 5.2 below.

Table 5.2: Describing the various hardware used for obtaining the results

Sno.	**Parameter**	**Specification**
1.	Processor	Intel® Core™ 2 CPU T5200@ 1.6 GHz
2.	RAM	2 GB

The results obtained have been reported in microseconds and are described in Table 5.3 and compared in Table 5.3.

Table 5.3: Comparison of Execution time of LB Routing and LBDP Routing Algorithm

N (Number of Nodes)	**Execution time (µs)**	
	LB	**LBDP (µs)**
10000	424	169
40000	1205	626
90000	3020	1197
160000	4352	3046
250000	10643	4451
360000	16252	6395
490000	21753	9465
640000	29725	13566
810000	38090	16713
1000000	48529	20739

From the Figure 5.8, this can have inferred that the LBDP routing algorithm on in comparison to that of existing LB routing algorithm is found to be always fast. The result obtained reveals that the improvement of two times at higher number of executions.

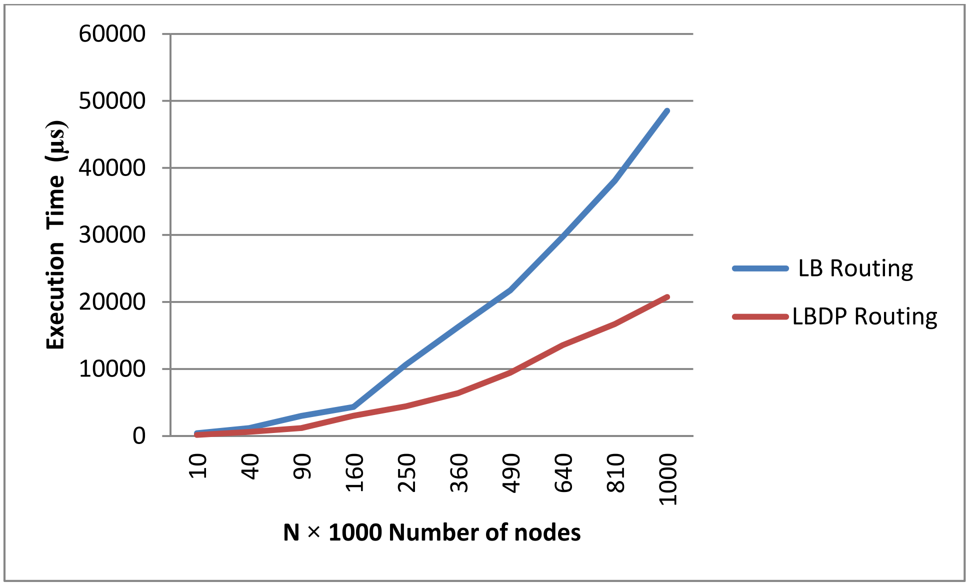

Figure 5.8: Execution Time of the Routing Algorithms

5.4.2.2 Comparison based on the Space complexity

The amount of the memory required by the particular algorithm is the significant parameter used to analyse an algorithm is. In the present routing algorithms, our main attention is on the number of bits needed to store the port number onto which the packet is to be sent.

In the case of LB routing algorithm the port numbers are $m - 1$, $m + 1$, $m - k$ *and* $m + k$, m for the five ports of the router. Here 'm' is the id of the current node and 'k' is a number of the node at a particular level. Now as the routing algorithm should be same for all the routers in the topology so the maximum port address generated by the LB routing algorithm will be N-1 as it is assumed that the topology is having N nodes and are numbered from 0 to N-1. So, the number of bits required will be given by the equation (5.4).

$$b = \log_2 N \tag{5.4}$$

In the case of LBDP routing algorithm, the maximum number of bits required will be equal to 3 as there will be maximum five ports in any router. The detailed comparison of a number of bits required is shown in Figure 5.9 and

Table 5.4. Figure 5.9 shows that the bits required to store the port address is fixed in the case of LBDP routing algorithm but increases in the case of LB routing algorithm.

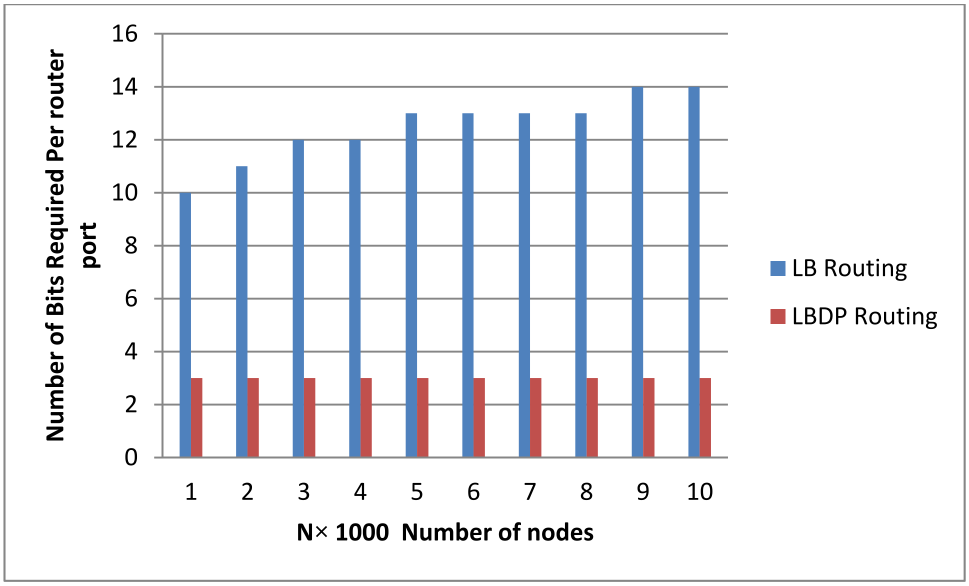

Figure 5.9: Bits Required for Representing the Port Number

Table 5.4: Describing the number of bits required per port per router

Number of nodes N	Number of bits required per router per port address	
	LB Routing	**LBDP Routing**
1000	10	3
2000	11	3
3000	12	3
4000	12	3
5000	13	3
6000	13	3
7000	13	3
8000	13	3
9000	14	3
10000	14	3

CHAPTER 6
ADAPTIVE ROUTING USING SIMPLE ROUTERS

6.1 Introduction

The adaptive routing algorithms require the dedicated links and controller to monitor the stress condition of the network. In this chapter, A routing algorithm based on the theory of maximum entropy model has been proposed[86, 87]. In the past, topology designing was performed by employing the information coding theory based on the entropy [88]. Therefore, the ideas of entropy in maintaining the equilibrium on the router is implemented to maintain the record of packets sent in each direction.

The Entropy based XY (E-XY) Routing algorithm can be implemented using a simple hardware same as that used for XY routing algorithm. The merits of proposed idea are

1. Area required to design the router will be less as congestion controlling hardware and space required to communicated stress signals are removed which are needed by most of the adaptive routing algorithms.
2. As the controllers and links involved in the communication are less so, this will reduce the heat dissipation.
3. The overall cost of the NOC with the adaptive routing will be reduced to equivalent to the simple XY routing algorithm.

6.2 Maximum Entropy Model

Entropy is the measure of information that is required to describe the value of a random variable. For a random variable *X*, let the probability of value *x* be *p(x)*. Then the entropy *H(x)* can be given by the equation (6.1).

$$H(x) = -\sum p(x)\log_2(x) \tag{6.1}$$

The entropy in the case of information is estimated using the number of binary bits.

The system is found to be in the state of equilibrium when the system has the highest entropy based on the theory of maximum entropy. In details, if *X* is the Random variable $x_0, x_1, .., x_n$ be the values that can exist for random variable *X*. However, the probabilities of occurrence of the values are unknown for a particular random variable *X*. These data may be constrained to function say *f*. *f* can be used to decide the value of probability such that the condition laid down by *f* are satisfied. In starting, the probability is unknown so they can be selected randomly. The Equation (6.2) estimates the likelihood of occurrence of a particular, where t_i is representing the total occurrence of particular values and t_{all} is representing the total samples under consideration.

$$P_i = \frac{t_i}{t_{all}} \tag{6.2}$$

The maximum entropy method presents the additional constraint: one that does not introduce any data or values of its own. This approach utilizes the parameters that is given and does not make any assumptions regarding the missing information. The non-addition of additional assumptions is the major advantage of the maximum entropy method.

6.3 Entropy Based XY Routing Algorithm (E-XY)

The entropy-based routing algorithm is designed considering the processing element and the router is the system boundaries. Besides, any parameters outside these limits are considered to be the external environment for the system. The four ports on the router are behaving as an inlet or outlet of the system. Our objective is to route the packets in order to achieve the condition of equilibrium, which is also the condition of maximum entropy. The exact algorithm 6.1 describes the entire working of the routing algorithm. The algorithm is using the extra variables like T_{pkt} to store the total number of packets, n_b, s_b, w_b, e_b represents the packets sent in each direction, P_w, P_e, P_n, P_s represent the probability in each direction. The X_{offset} and Y_{offset} are calculated to store the difference in of source and destination in both the dimensions. Port is the variable used to store the value of port to record the decision port on which the packet is sent.

Algorithm 6.1: Entropy Based XY Routing Algorithm for 2D mesh

INPUT: Coordinates of current node Cx, Cy and destination node Dx, Dy

OUTPUT: Destination port.

1: SET $T_{pkt} = n_b + s_b + w_b + e_b$

2: SET $P_w = w_b/T_{pkt}$

SET $P_e = e_b/T_{pkt}$

SET $P_n = n_b/T_{pkt}$

SET $P_s = s_b/T_{pkt}$

CALCULATE $X_{offset} = D_x - C_x$

CALCULATE $Y_{offset} = D_y - C_y$

If $X_{offset} > 0$ **and** $Y_{offset} > 0$ **Then**

  If $Pw < Ps$ **Then**

  Port = West

  Increment w_b

  Else

  Port = South

  Increment s_b

  End If

End If

If $X_{offset} < 0$ **and** $Y_{offset} > 0$ **Then**

  If $P_e < P_s$ **Then**

  Port = East

  Increment e_b

  Else

  Port = South

  Increment s_b

  End If

End If

If $X_{offset} < 0$ **and** $Y_{offset} < 0$ **Then**

  If $Pe < Pn$ **Then**

  Port = East

  Increment e_b

  else

  Port = North

  Increment n_b

  End If

```
End If
If Xoffset > 0 and Yoffset < 0 Then
    if Pw < Pn then
    Port = West
     Increment wb
    else
    Port = North
    Increment nb
     End If
End If
If Xoffset > 0 and Yoffset = 0 Then
    Port = West
    Increment wb
End If
If Xoffset < 0 and Yoffset = 0 Then
    Port = East
    Increment eb
End If
If Xoffset = 0 and Yoffset > 0 Then
    Port = North
    Increment nb
End If
If Xoffset = 0 and Yoffset < 0 Then
    Port = South
    Increment sb
End If
If Xoffset = 0 and Yoffset = 0 Then
     Port = Core
End If
```

To understand working of the E-XY routing algorithm let us consider a mesh topology of 3 × 3 as described in Figure 6.1.

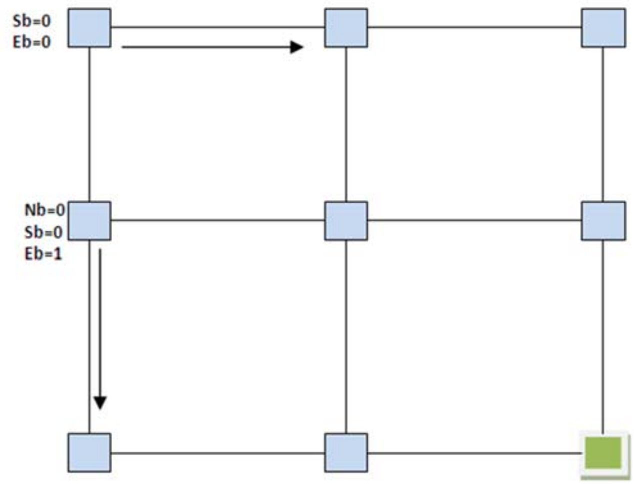

Figure 6.1: 3 × 3 Mesh with Entropy-Based XY Routing Algorithm

As each router has to maintain additional parameters for the number of packets routed in particular direction. As the node in a mesh has maximum of 4 neighbours located in east, west, north and south of the node so these parameter are named are named eb, wb ,nb and sb. During the initialization phase, no packet has been routed to none of the neighbour hence these parameters are initialized to zero.. Two diverse scenarios are used to understand the complete routing process of the algorithm.

In the first scenario, the router is in the position of equilibrium and in the second scenario; the router is not at the position of equilibrium. At equilibrium, the probability of the occurrence of each of the event is same.

In scenario 1, the router will be in the state of equilibrium that means that packet can move either in the *X* or in *Y* direction with equal probability. Now, according to XY routing, the *X* direction will be followed, if it is the part of the shortest path, otherwise we can move in the *Y* direction. The Figure 6.1 describes that the first node is sending the data to the destination in the green color as s_b and e_b are zero, so they will be equiprobable. Therefore it is the state of equilibrium, and the router uses the XY routing that is the *X* direction first.

In scenario 2, the calculation of the probability for sending of each port is done such that the router is not in an equilibrium position. The selection of *X* and *Y* ports are made according to the shortest path to the destination. From the list of the selected ports, a single port is selected based on the probability value to maximise the entropy by making the entropy of the state of equilibrium. We can observe the value of e_b is one which means there is a greater probability of sending packets in e_b The same can be inferred from Figure 6.1 if the packet is sent from the first node of the second row,. Now, according to the XY routing packet can move in both *X* and *Y* direction in south and east, moving in the south direction the system can reach the state of equilibrium, so the packet is routed in the south direction.

6.3.1 Cost Effectiveness

The E-XY routing algorithm is cost effective because there is no need to employ the extra hardware that will increase the cost as used in congestion control routing algorithms like Dxy[89] and EDxy[90]. The additional hardware are state communicating wires and various control units in the routers. This will not only reduce the cost of building the adaptive routers but will also reduce the running cost of the hardware.

6.3.2 Deadlock Free

As the XY routing is deadlock free because it does not allow 180 degree turns, which are the source of deadlocks. E-XY routing algorithm is based on XY routing, which means no deadlock creating routes are used. Only the decision to select the *X* and *Y* direction is made by entropy. Hence the E-XY routing algorithm is also deadlock free.

6.4 Experimental Setup

To test the performance of the E-XY routing algorithm a mesh topology of 8×8 which has been simulated using OmNet++ 4.4.1 using HNOCS version[5]. Table 6.1 describes the detailed configuration for testing the topology.

Table 6.1: Various parameters used in the testing of entropy-based routing algorithm

Parameters	Value
Rows	8
Columns	8
Channel Width	16 Gbps
Number of Virtual Channels	2
Flit Size	4 Bytes
Message Length	2 packets per message
Packet Length	8 Flits

Flit Arrival delay	Varying according to load factor
Warm-Up Period	20 ns
Simulation Time	200 ms
Maximum Queued Packet	16
Traffic Types	Uniform traffic, Bit Complement traffic, Neighbour traffic, Tornado traffic
Number of runs	5

6.5 Results and Discussions

The simulation on the 8 × 8 mesh is performed to study the performance of E-XY routing and the comparison of the results is made with various well-known routing algorithms. Figure 6.2 describes the throughput on the uniform traffic which shows that throughput is almost same to that of XY routing but is better than the odd even routing algorithm and IX/Y routing algorithm. From Figure 6.3 it can be observed that the latency of the E-XY routing algorithm is less in comparison to that of XY routing and IX/Y routing algorithm at single data point. The combined view of both can be visualised using the performance metric or power of the network as described in Figure 6.4. The performance metrics or power of the network is defined as the ratio of average throughput and average latency of the packets received [15]. The equation (6.3) can describe it.

$$Performance\,metrics = \frac{Average\ throughput}{Average\,latency} \tag{6.3}$$

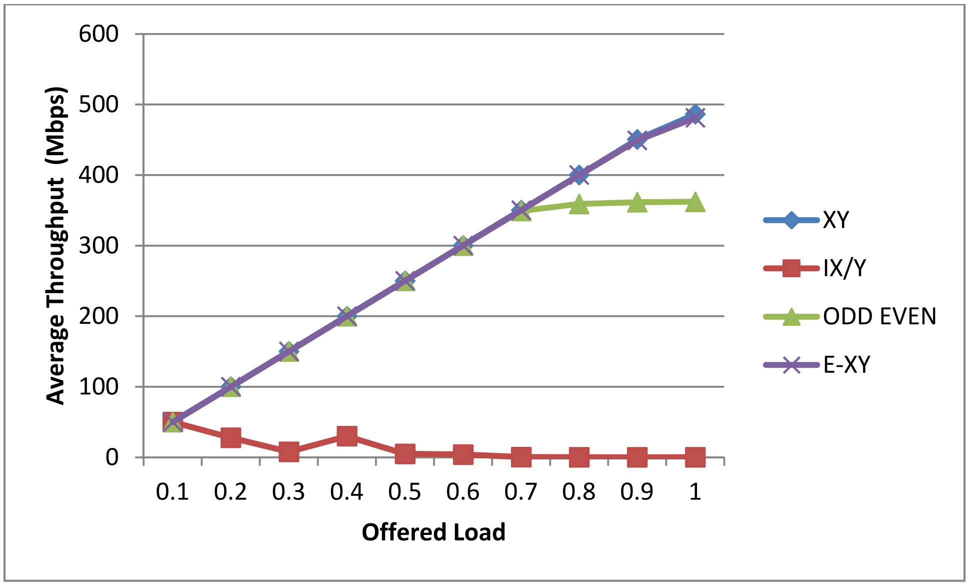

Figure 6.2: Average Throughput on Uniform Traffic

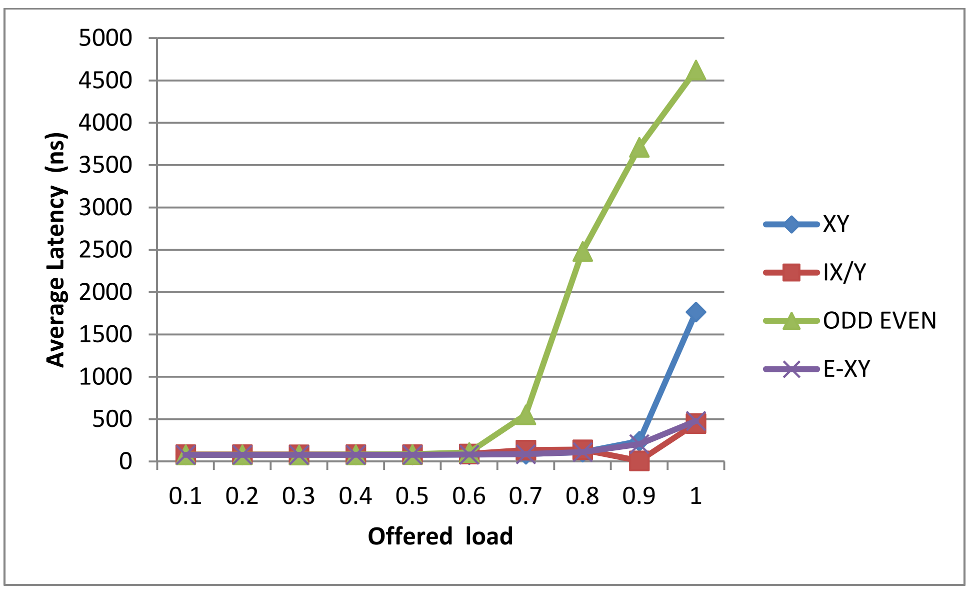

Figure 6.3: Average Latency on Uniform Traffic

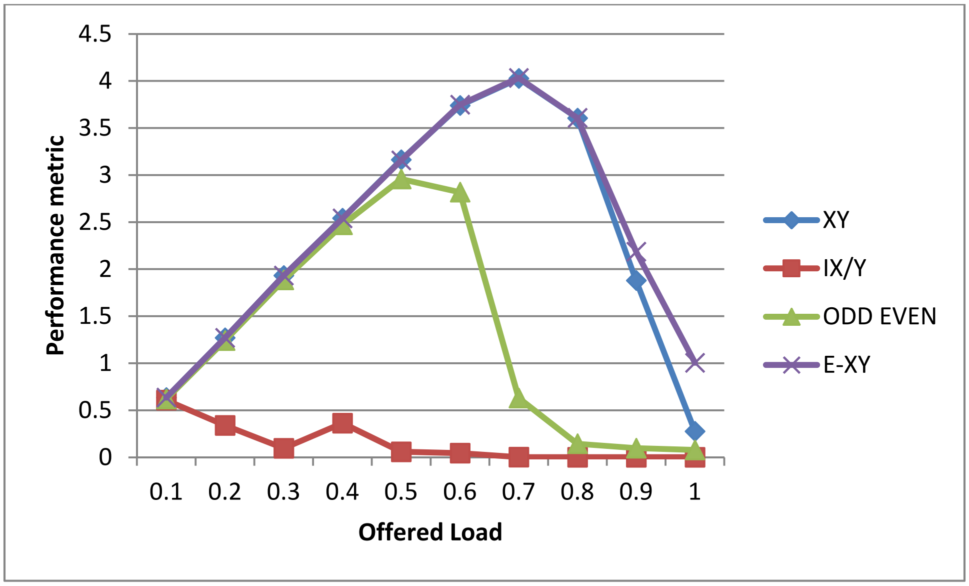

Figure 6.4: Power of Network on Uniform Traffic

From Figure 6.4 we can see that E-XY algorithm is having the marginal improvement in the performance on the higher load at 0.9 at 1.0.

In the case of neighbour traffic the throughput of XY routing was almost the same for the XY routing and E-XY routing algorithm as described in Figure 6.5, In Figure 6.6 the latency was also the same as that of XY routing algorithm except at the load factor of 0.8 to 1.0. However, we can see the combined effect of both the throughput and latency in the performance metric or power of network described in Figure 6.7. From this, it is ascertained that results are slightly better on higher traffic loads that are from 0.8 to 1.0.

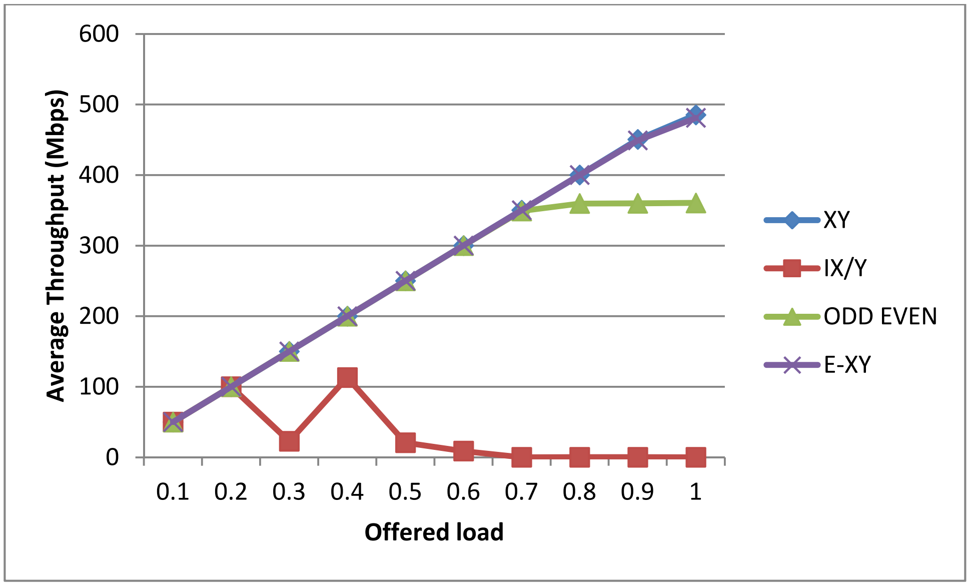

Figure 6.5: Average Throughput on Neighbour Traffic

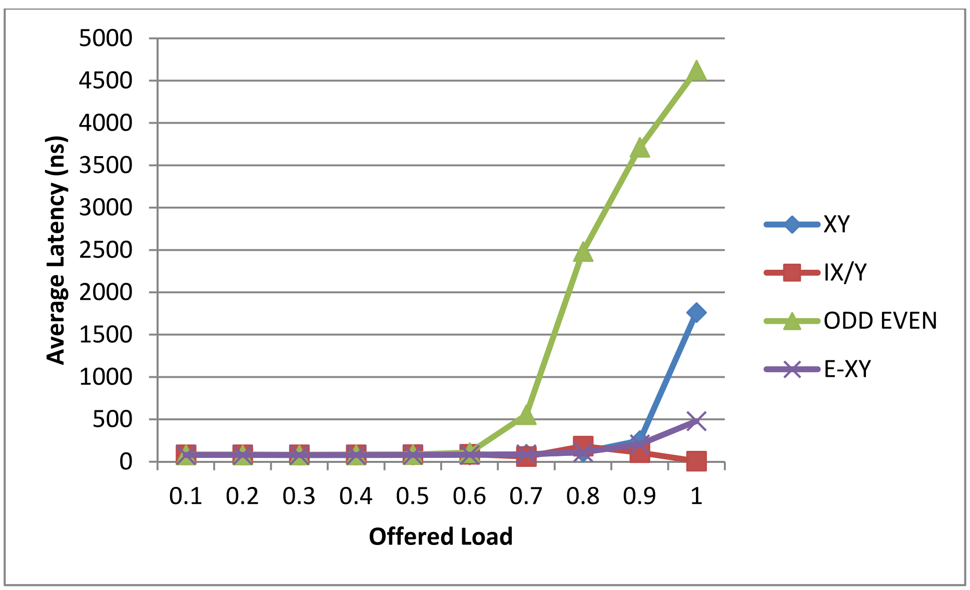

Figure 6.6: Average Latency on Neighbour Traffic

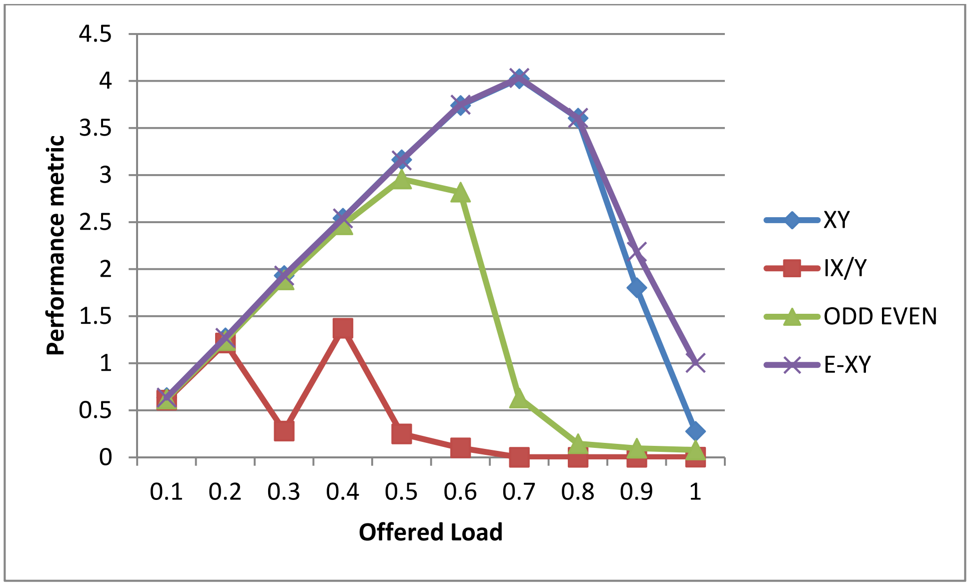

Figure 6.7: Power of Network on Neighbour Traffic

In case of bit complement traffic both Figure 6.9 shows that the latency of all the four routing algorithms is identical until 0.7 at the load factor of 0.8 IX/Y routing algorithm is slightly high in comparison to the other three routing algorithm . At the load factor of 0.9, the latency of XY Routing algorithm is slightly high in comparison to E-XY Routing algorithm. At the load factor of 1.0, the latency of XY routing algorithm is very high in comparison to other three routing algorithms... In Figure 6.10 we can see that performance metric is again good for the offered load from 0.8 to 1.0.

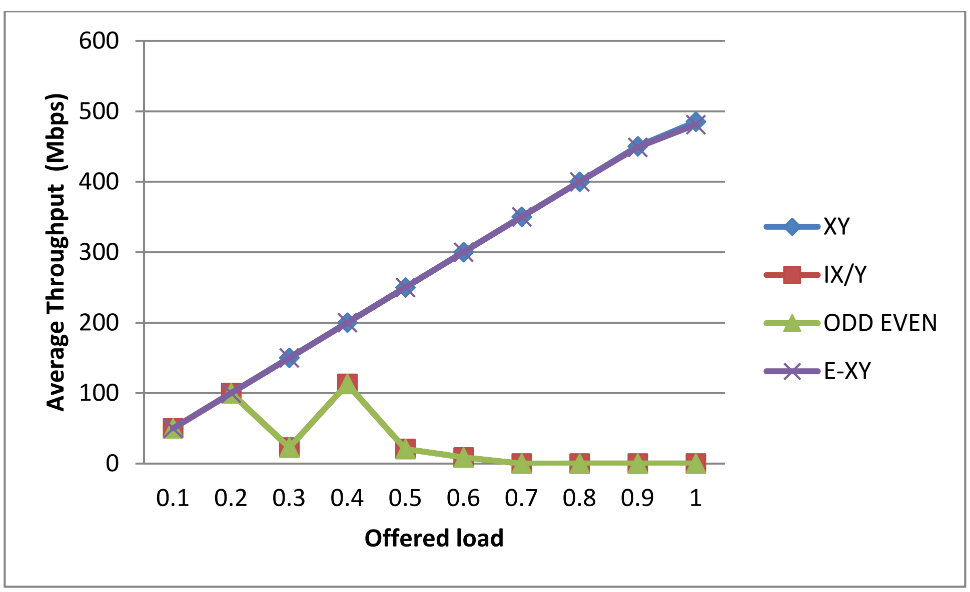

Figure 6.8: Average Throughput on Bit Complement Traffic

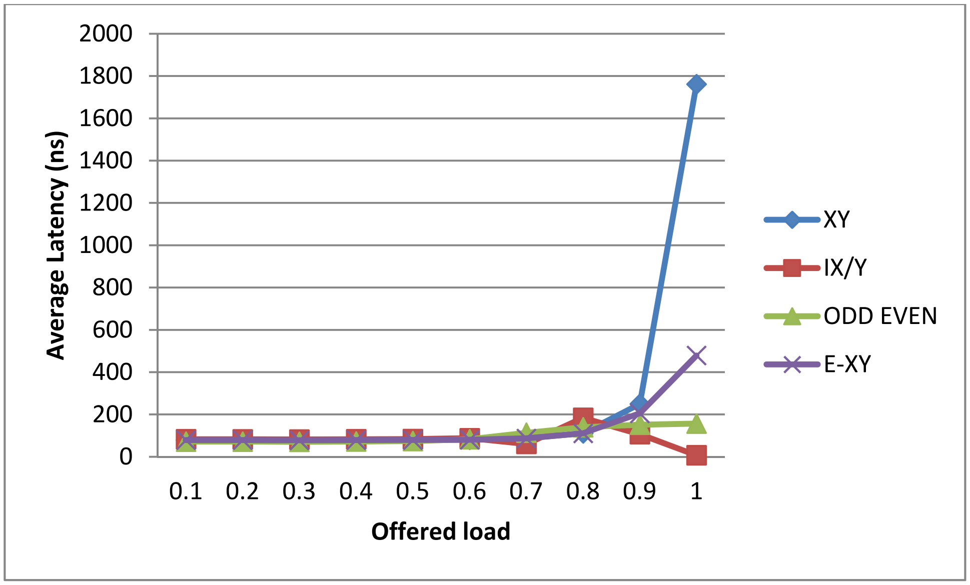

Figure 6.9: Average Latency on Bit Complement Traffic

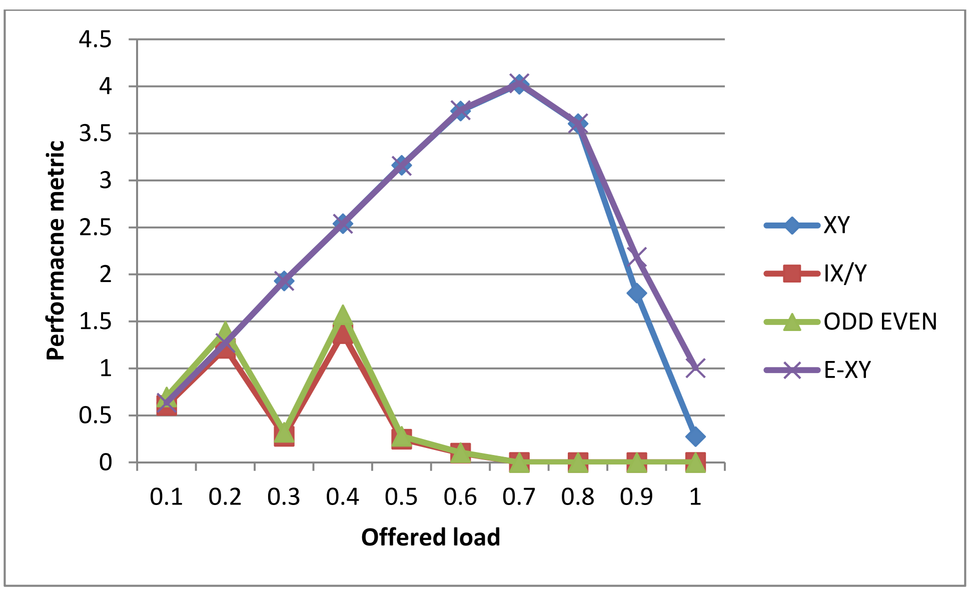

Figure 6.10: Power of the Network on Bit Complement Traffic

Figure 6.11 shows that the average throughput of E-XY is similar to XY routing algorithm. The latency of E-XY is slightly lower than XY routing algorithm between the offered loads of 0.8 to 1 the same has been represented in the Figure 6.12. Figure 6.13 shows the improvement achieved by E-XY routing algorithm in comparison to XY routing algorithm between the load factors of 0.8 to 1.0.

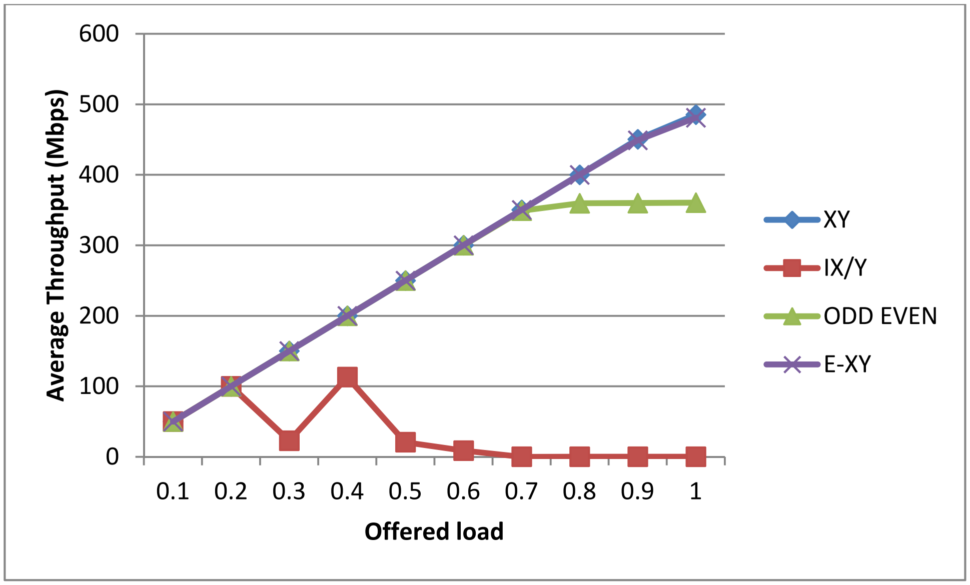

Figure 6.11: Average Throughput on Tornado Traffic

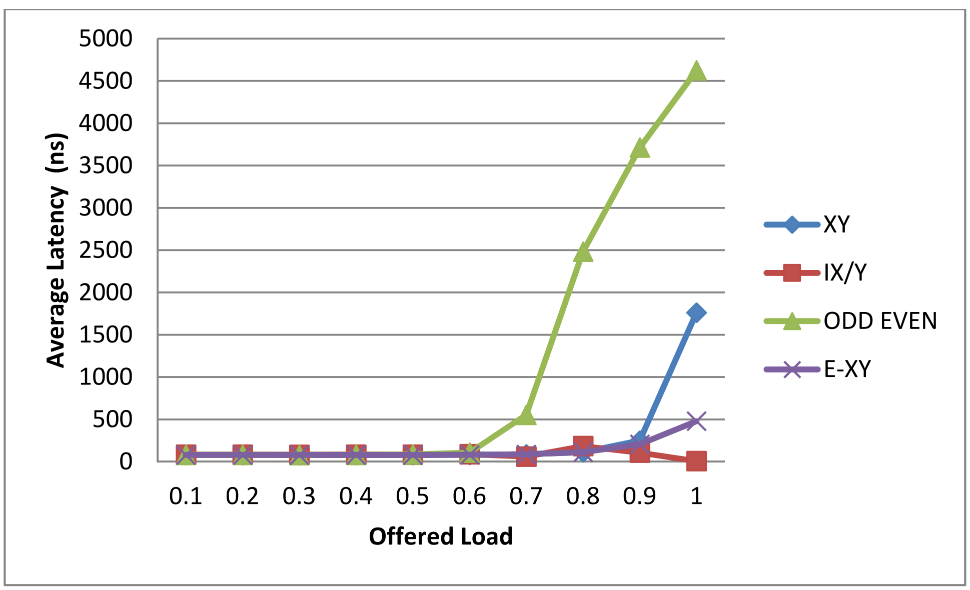

Figure 6.12: Average Latency on Tornado Traffic

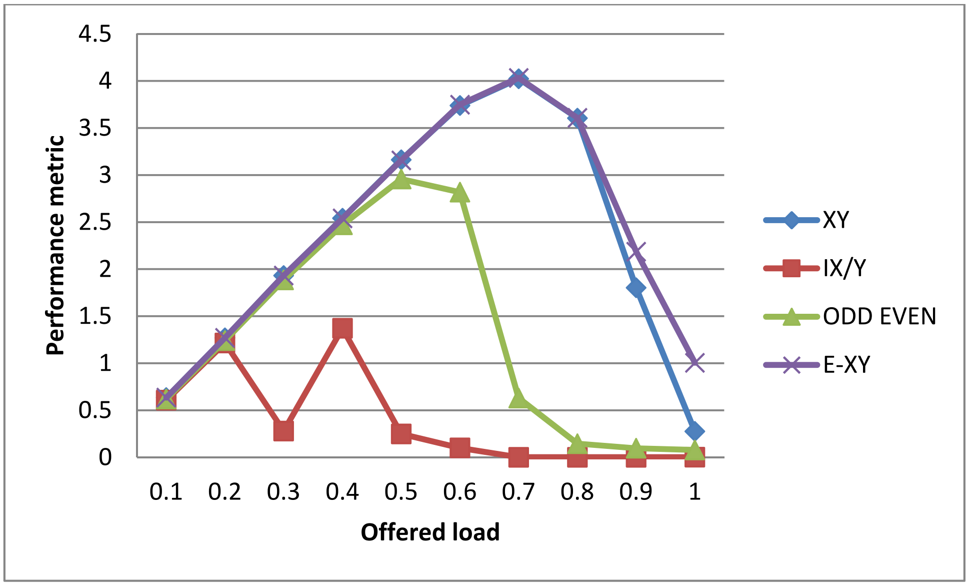

Figure 6.13: Power of the Network on Tornado Traffic

CHAPTER 7
CONCLUSIONS AND FUTURE WORK

The current thesis presents three 2-D topologies and a single 3D topology with the aim of achieving high performance. The study attempts to achieve the result in terms of throughput and latency by reducing the use of the toroidal links. The topologies have been tested on various loads and, out of all the topologies, MDMIN and MDMSEIN have performed better when compared to the other topologies on the neighbors' traffic and hotspot traffic. The 3D MDMIN has been proved better than the other 3D topologies. Moreover, the 2D torus topology has significantly achieved an improvement, due to the supportive location of the nodes. In addition, another DCT topology has performed better than DCM and TMESH. Most of the variants of mesh targets to reduce the diameter by adding links based on the human opinion, which may not be the optimal case. The search for optimal links in the topology is carried out by using the heuristic technique called as the improved environmental adaptation method. The topology with an extra link on the mesh is generated using an improved environmental adaptation method. The improved method is asserted to be better than the existing topologies with four extra links. The improvement produced by IEAM Mesh is 2.2 times in performance metric or power of network over the mesh, C^2 Mesh and T Mesh. Hence, the described technique can be beneficial in topology explorations with the given base topology and number of links to be explored. Another major challenge is designing an efficient routing algorithm for mesh. The proposed dynamic programming based routing algorithm is more efficient than its counterparts, concerning to execution time and space requirement. The routing algorithms though the application of the dynamic programming has shown the improvement by a factor of 2X. The deterministic routing algorithm must try routing the packet through the shortest distance and same path through which the MCCM algorithm highlights the same issue of selecting the shortest path in CCM routing algorithm. The MCCM has improved the bandwidth by 10%, reduced latency by 21% and hop count by 7%. Therefore, the MCCM approach should be applied in designing the topology for center concentrated topologies. The adaptive routing algorithm requires the communication of the stress value which requires extra dedicated links and special routers to handle these stress values. But, the entropy-based routing algorithm attempts to carry out the computation on the router itself and reduces the complexity of the networks. Therefore, it can be concluded that the entropy based routing algorithm are suitable for higher traffic loads, in comparison to the existing routing algorithms.

The study of mesh topologies with the implementation of the Artificial Intelligence tools can decidedly boost the performance of the mesh interconnection networks. In the current era of Big Data, and the cloud-based chips, the topology explorations based on the traffic patterns of the cloud-based architecture can significantly aid in innovations in the field of interconnection networks. Power is also a critical factor, which should be studied for the mesh topologies. Hence, there is the need for the routing algorithms to be power aware as most of the heat is inherently dissipated by the communication networks. The power can be reduced, which will directly influence the computing devices by reducing the carbon footprints generated.

REFERENCES

[1] T. William James, Dally; Brian Patrick, *Principles and Practices of Interconnection Networks*. Elsevier, 2004.

[2] Jose Duato Sudhakar Yalamanchili Lionel Ni, *Interconnection Networks*. 2003.

[3] V. Rantala, T. Lehtonen, J. Plosila, and T. T. Report, "Network on Chip Routing Algorithms," *Technology*, no. 779, 2006.

[4] C. Nicopoulos, V. Narayanan, and C. R. Das, *Network-on-Chip Architectures*, vol. 45. 2009.

[5] R. Manevich, I. Cidon, A. Kolodny, and I. Walter, "Centralized adaptive routing for NoCs," *IEEE Comput. Archit. Lett.*, vol. 9, no. 2, pp. 57–60, 2010.

[6] C. J. Glass and L. M. Ni, "The Turn Model for Adaptive Routing," *Comput. Archit. 1992. Proceedings., 19th Annu. Int. Symp.*, pp. 278–287, 1992.

[7] C. J. Glass and L. M. Ni, "The turn model for adaptive routing," *SIGARCH Comput. Arch. News*, vol. 20, pp. 278–287, 1992.

[8] Ge-Ming Chiu, "The odd-even turn model for adaptive routing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 11, no. 7, pp. 729–738, Jul. 2000.

[9] N. Gupta, M. Kumar, V. Laxmi, M. S. Gaur, and M. Zwolinski, "σlBDR: Congestion-aware logic based distributed routing for 2D NoC," *19th Int. Symp. VLSI Des. Test, VDAT 2015 - Proc.*, 2015.

[10] A. Mejia *et al.*, "Region-based routing: A mechanism to support efficient routing algorithms in NoCs," *IEEE Trans. Very Large Scale Integr. Syst.*, vol. 17, no. 3, pp. 356–369, Mar. 2009.

[11] G. Nychis, C. Fallin, T. Moscibroda, and O. Mutlu, "Next generation on-chip networks: What kind of congestion control do we need?," *Proc. 9th ACM* pp. 0–5, 2010.

[12] W. J. Dally and B. Towles, "Route Packets, Not Wires: On-Chip Interconnection Networks," in *Proceedings of the 38th conference on Design automation - DAC '01*, 2001, pp. 684–689.

[13] P. Salihundam *et al.*, "A 2 Tb/s 6X4 Mesh Network for a Single-Chip Cloud Computer With DVFS in 45 nm CMOS," *IEEE J. Solid-State Circuits*, vol. 46, no. 4, pp. 757–766,

2011.

[14] A. Punhani, P. Kumar, and Nitin, “A Modified Diagonal Mesh Interconnection Network,” in *2014 Annual IEEE India Conference (INDICON)*, 2014, pp. 1–6.

[15] A. Punhani, P. Kumar, and N. Nitin, “A Modified Diagonal Mesh Shuffle Exchange Interconnection network,” *Int. J. Electr. Comput. Eng.*, vol. 7, no. 2, pp. 1042–1050, 2017.

[16] A. Punhani, P. Kumar, and N. Nitin, “Three-dimensional topology based on modified diagonal mesh interconnection network,” *J. Telecommun. Electron. Comput. Eng.*, vol. 9, no. 3–6, 2017.

[17] A. Punhani, P. Kumar, and Nitin, “Diagonal Connected T Mesh,” *Indian J. Sci. Technol.*, vol. 9, no. 32, pp. 1–7, Aug. 2016.

[18] A. Punhani, P. Kumar, and N. Nitin, “Optimal extra links placement in mesh interconnection network using improved environmental adaptation method,” *J. Intell. Fuzzy Syst.*, vol. 32, no. 32, pp. 3285–3295, 2017.

[19] L. K. Arora, “Alternatives of XY-Routing for Mesh,” *Int. J. Comput. Appl.*, no. November, pp. 6–8, 2012.

[20] A. Punhani, P. Kumar, and N. Nitin, “Routing for Center Concentrated Mesh,” *Int. J. Intell. Eng. Syst.*, vol. 10, no. 1, pp. 86–94, Feb. 2017.

[21] A. Punhani, P. Kumar, and N. Nitin, “Level Based Routing Using Dynamic Programming for 2D Mesh,” *Cybern. Inf. Technol.*, vol. 17, no. 2, pp. 73–82, 2017.

[22] “System Architecture,” in *Computer Architecture Technology Trends*, Elsevier, 1991, pp. 23–38.

[23] R. Duncan, “Survey of parallel computer architectures,” *Computer (Long. Beach. Calif).*, vol. 23, pp. 5–16, 1990.

[24] W. J. Dally and J. W. Poulton, *Digital Systems Engineering*. Cambridge: Cambridge University Press, 1998.

[25] J. Duato, S. Yalamanchili, and L. M. Ni, *Interconnection networks: an engineering approach*. Morgan Kaufmann, 2003.

[26] M. J. QUINN, *Parallel Programming In C With Mpi And Open MP*. 2003.

[27] S. Vangal *et al.*, “An 80-Tile 1.28TFLOPS Network-on-Chip in 65nm CMOS,” in *2007 IEEE International Solid-State Circuits Conference. Digest of Technical Papers*, 2007, pp. 98–589.

[28] K. Swaminathan, S. Gopi, Rajkumar, G. Lakshminarayanan, and S.-B. Ko, “A novel hybrid topology for Network on Chip,” in *2014 IEEE 27th Canadian Conference on Electrical and Computer Engineering (CCECE)*, 2014, pp. 1–6.

[29] P. Gratz, C. Kim, R. McDonald, S. W. Keckler, and D. Burger, “Implementation and Evaluation of On-Chip Network Architectures,” in *2006 International Conference on Computer Design*, 2006, pp. 477–484.

[30] J. M. Camara *et al.*, “Twisted Torus Topologies for Enhanced Interconnection Networks,” *IEEE Trans. Parallel Distrib. Syst.*, vol. 21, no. 12, pp. 1765–1778, Dec. 2010.

[31] A. Tavakkol, R. Moraveji, and H. Sarbazi-Azad, “Mesh Connected Crossbars: A Novel NoC Topology with Scalable Communication Bandwidth,” in *2008 IEEE International Symposium on Parallel and Distributed Processing with Applications*, 2008, pp. 319–326.

[32] W. Bouknight, S. A. Denenberg, D. E. McIntyre, J. Randall, A. H. Sameh, and D. L. Slotnick, “The Illiac IV system,” in *Proceedings of the IEEE*, 1972, vol. 60, no. 4, pp. 369–388.

[33] R. Beivide, E. Herrada, J. L. Balcazar, and J. Labarta, “Optimized mesh-connected networks for SIMD and MIMD architectures,” in *Proceedings of the 14th annual international symposium on Computer architecture - ISCA ’87*, 1987, pp. 163–170.

[34] J. J. Kim and H. M. Choi, “XMESH interconnection network for massively parallel computers,” *IEE Proc. - Comput. Digit. Tech.*, vol. 143, no. 6, p. 401, 1996.

[35] L. K. Arora and Rajkumar, “C^2Mesh,” in *2013 3rd IEEE International Advance Computing Conference (IACC)*, 2013, pp. 282–286.

[36] O. Yi-ming, Z. H. U. Bing, L. Hua-guo, F. Wei, and W. Ouyang, YM and Zhu, Bing and Liang, HG and FENG, “Networks on Chip Based on Diagonal Interlinked Mesh Topology Structure,” *Comput. Eng.*, vol. 35, no. 22, pp. 100–102, 2009.

[37] Q. Yang and Z. Wu, “An improved mesh topology and its routing algorithm for NoC,” *2010 Int. Conf. Comput. Intell. Softw. Eng. CiSE 2010*, pp. 0–3, 2010.

[38] K. W. Tang and S. A. Padubidri, “Diagonal and Toroidal Mesh Networks,” *IEEE Trans. Comput.*, vol. 43, no. 7, pp. 815–826, 1994.

[39] Nitin, R. Vaish, and U. Shrivastava, “On a deadlock and performance analysis of ALBR and DAR algorithm on X-Torus topology by optimal utilization of Cross Links and minimal lookups,” *J. Supercomput.*, vol. 59, no. 3, pp. 1252–1288, 2010.

[40] H. Gu, Q. Xie, K. Wang, J. Zhang, and Y. Li, "X-Torus: A Variation of Torus Topology with Lower Diameter and Larger Bisection Width," 2006, pp. 149–157.

[41] Y. H. Liu, M. F. Zhu, J. Wang, L. M. Xiao, and T. Gong, "Xtorus: An extended torus topology for on-chip massive data communication," in *Proceedings of the 2012 IEEE 26th International Parallel and Distributed Processing Symposium Workshops, IPDPSW 2012*, 2012, pp. 2061–2068.

[42] Y. Wang, H. Du, and X. Shen, "Topological properties and routing algorithm for semi-diagonal torus networks," *J. China Univ. Posts Telecommun.*, vol. 18, no. 5, pp. 64–70, 2011.

[43] O. Yi-ming, Z. H. U. Bing, L. Hua-guo, and F. Wei, "Networks on Chip Based on Diagonal Interlinked Mesh Topology Structure," *Comput. Eng.*, vol. 35, no. 22, p. 100 102, 2009.

[44] X.-J. ZHU, "Xmesh: A Mesh-Like Topology for Network on Chip," *J. Softw.*, vol. 18, no. 9, p. 2194, 2007.

[45] S. Yadav and C. R. Krishna, "CCTorus: A New Torus Topology for Interconnection Networks," in *International Conference on Advanced Computational Technologies and Creative Media*, 2014.

[46] J. Camacho and J. Flich, "HPC-Mesh: A Homogeneous Parallel Concentrated Mesh for Fault-Tolerance and Energy Savings," in *2011 ACM/IEEE Seventh Symposium on Architectures for Networking and Communications Systems*, 2011, pp. 69–80.

[47] R. K. Saini and M. Ahmed, "2D Hexagonal Mesh Vs 3D Mesh Network on Chip : A Performance Evaluation," vol. 1, no. 1.

[48] J. Camacho, J. Flich, J. Duato, H. Eberle, and W. Olesinski, "Towards an Efficient NoC Topology through Multiple Injection Ports," in *2011 14th Euromicro Conference on Digital System Design*, 2011, pp. 165–172.

[49] A. E. Zonouz, M. Seyrafi, A. Asad, M. Soryani, M. Fathy, and R. Berangi, "A fault tolerant NoC architecture for reliability improvement and latency reduction," *12th Euromicro Conf. Digit. Syst. Des. Archit. Methods Tools, DSD 2009*, pp. 473–480, 2009.

[50] I. Stojmenovic, "Honeycomb networks: Topological properties and communication algorithms," *IEEE Trans. Parallel Distrib. Syst.*, vol. 8, no. 10, pp. 1036–1042, 1997.

[51] P. Ghosal and T. S. Das, "Network-on-chip routing using Structural Diametrical 2D mesh

architecture," in *2012 Third International Conference on Emerging Applications of Information Technology*, 2012, pp. 471–474.

[52] M. Reshadi, A. Khademzadeh, A. Reza, and M. Bahmani, "A Novel Mesh Architecture for On-Chip Networks," *https://www.design-reuse.com/articles/23347/on-chip-network.html*.

[53] Y. Liu, J. Han, and H. Du, "DL(2m): A new scalable interconnection network for system-on-chip," *J. Comput.*, vol. 4, no. 3, pp. 201–207, Mar. 2009.

[54] M. H. Furhad and J.-M. Kim, "A shortly connected mesh topology for high performance and energy efficient network-on-chip architectures," *J. Supercomput.*, vol. 69, no. 2, pp. 766–792, Aug. 2014.

[55] U. A. Gulzari, S. Anjum, and S. Agha, "Cross by Pass-Mesh Architecture for On-chip Communication," in *2015 IEEE 9th International Symposium on Embedded Multicore/Many-core Systems-on-Chip*, 2015, pp. 267–274.

[56] U. A. Gulzari, M. Sajid, S. Anjum, S. Agha, and F. S. Torres, "A New Cross-By-Pass-Torus Architecture Based on CBP-Mesh and Torus Interconnection for On-Chip Communication," *PLoS One*, vol. 11, no. 12, p. e0167590, Dec. 2016.

[57] G. B. P. Bezerra, S. Forrest, M. Forrest, A. Davis, and P. Zarkesh-Ha, "Modeling NoC traffic locality and energy consumption with rent's communication probability distribution," in *Proceedings of the 12th ACM/IEEE international workshop on System level interconnect prediction - SLIP '10*, 2010, p. 3.

[58] M. Y. Qadri and S. J. Sangwine, *Multicore Technology: Architecture, Reconfiguration, and Modeling*. .

[59] A. Mejia *et al.*, "Region-Based Routing: A Mechanism to Support Efficient Routing Algorithms in NoCs," *IEEE Trans. Very Large Scale Integr. Syst.*, vol. 17, no. 3, pp. 356–369, Mar. 2009.

[60] A. Punhani, Nitin, and P. Kumar, "A modified diagonal mesh interconnection network," in *2014 Annual IEEE India Conference (INDICON)*, 2014, pp. 1–6.

[61] A. Chauhan, A. Punhani, and Nitin, "EMC2Mesh," in *2015 Annual IEEE India Conference (INDICON)*, 2015, pp. 1–5.

[62] A. Chauhan, A. Punhani, and Nitin, "Comparative analysis of traffic patterns on centre connected topologies based on burton normal form," in *TENCON 2015 - 2015 IEEE Region 10 Conference*, 2015, pp. 1–6.

[63] W.-H. Hu *et al.*, “Xmesh: a mesh-like topology for network on chip,” *Netw. Chip Archit.*, p. 14.

[64] W. Hu, S. E. Lee, and N. Bagherzadeh, “DMesh : a Diagonally-Linked Mesh Network-on-Chip Architecture.”

[65] Q. Yang and Z. Wu, “An improved mesh topology and its routing algorithm for NoC,” in *2010 International Conference on Computational Intelligence and Software Engineering, CiSE 2010*, 2010, pp. 1–4.

[66] L. K. Arora and Rajkumar, “C2Mesh,” in *Advance Computing Conference (IACC), 2013 IEEE 3rd International*, 2013, pp. 282–286.

[67] L. Yu-hang, Z. Ming-fa, W. Jue, X. Li-min, and G. Tao, “Xtorus: An Extended Torus Topology for On-Chip Massive Data Communication,” in *Parallel and Distributed Processing Symposium Workshops & PhD Forum (IPDPSW), 2012 IEEE 26th International*, 2012, pp. 2061–2068.

[68] M. Ebrahimi, “Fully adaptive routing algorithms and region-based approaches for two-dimensional and three-dimensional networks-on-chip,” *IET Comput. Digit. Tech.*, vol. 7, no. 6, pp. 264–273, Nov. 2013.

[69] Y. Yang, A. Funahashi, A. Jouraku, H. Nishi, H. Amano, and T. Sueyoshi, “Recursive diagonal torus: An interconnection network for massively parallel computers,” *IEEE Trans. Parallel Distrib. Syst.*, vol. 12, no. 7, pp. 701–715, 2001.

[70] J. ´eDuato, S. Yalamanchili, and L. Ni, *Interconnection Networks*. MORGAN KAUFMANN PUBLIS.

[71] H. Siegel, *Interconnection networks for large-scale parallel processing: theory and case studies*. 1990.

[72] A. Varga and others, “The OMNeT++ discrete event simulation system,” in *Proceedings of the European simulation multiconference (ESM 2001)*, 2001, vol. 9, p. 65.

[73] A. Varga, “OMNeT++,” in *Modeling and Tools for Network Simulation*, Springer, 2010, pp. 35–59.

[74] D. Lenoski *et al.*, “The Stanford Dash Multiprocessor,” *Computer (Long. Beach. Calif).*, vol. 25, no. 3, pp. 63–79, 1992.

[75] N. Rakesh and Nitin, *Analysis of Multi-Sort Algorithm on Multi-Mesh of Trees (MMT) architecture*, vol. 57, no. 3. 2011.

[76] W. Hu, S. Lee, and N. Bagherzadeh, “DMesh: a diagonally-linked mesh network-on-chip architecture,” *First Int. Work. Netw. Chip Archit.*, pp. 1–7, 2008.

[77] Z. A. Khan, J. Siddiqui, and A. Samad, “Linear Crossed Cube (LCQ): A New Interconnection Network Topology for Massively Parallel System,” *Int. J. Comput. Netw. Inf. Secur.*, vol. 7, no. 3, pp. 18–25, 2015.

[78] A. Chauhan, A. Punhani, and Nitin, “EMC2Mesh,” in *2015 Annual IEEE India Conference (INDICON)*, 2015, pp. 1–5.

[79] K. K. Mishra, S. Tiwari, and A. K. Misra, “Improved environmental adaption method and its application in test case generation,” *J. Intell. Fuzzy Syst.*, vol. 27, no. 5, pp. 2305–2317, 2014.

[80] A. Varga, “Using the OMNeT++ discrete event simulation system in education,” in *the European Simulation Multiconference*, 2001.

[81] A. Varga, *Modeling and Tools for Network Simulation*. Springer Berlin Heidelberg, 2010.

[82] J. K. Singh, A. K. Swain, T. N. K. Reddy, and K. K. Mahapatra, “Performance evalulation of different routing algorithms in Network on Chip,” in *2013 IEEE Asia Pacific Conference on Postgraduate Research in Microelectronics and Electronics (PrimeAsia)*, 2013, pp. 180–185.

[83] Y. Ben-Itzhak, E. Zahavi, I. Cidon, and A. Kolodny, “HNOCS: Modular open-source simulator for Heterogeneous NoCs,” in *2012 International Conference on Embedded Computer Systems (SAMOS)*, 2012, pp. 51–57.

[84] Y. Cheng, W. Zhuang, and L. Wang, “Calculation of Loss Probability in a Finite Size Partitioned Buffer for Quantitative Assured Service,” *IEEE Trans. Commun.*, vol. 55, no. 9, pp. 1757–1771, Sep. 2007.

[85] S. Skiena, “Dijkstra’s algorithm,” *Implement. Discret. Math. Comb. Graph Theory with Math. Reading, MA Addison-Wesley*, pp. 225–227, 1990.

[86] K. Kawasaki, “Principle of Maximum Entropy and Reduced Dynamics,” *J. Stat. Phys.*, vol. 123, no. 4, pp. 711–740, May 2006.

[87] K. Kocik, “Question classification using maximum entropy models,” 2004.

[88] Y.-H. Liu, M.-F. Zhu, L.-M. Xiao, and J. Wang, “Asymmetrical topology and entropy-based heterogeneous link for many-core massive data communication,” *Cluster Comput.*, vol. 16, no. 4, pp. 679–691, Dec. 2013.

[89] Ming Li, Qing-An Zeng, and Wen-Ben Jone, "DyXY - a proximity congestion-aware deadlock-free dynamic routing method for network on chip," in *2006 43rd ACM/IEEE Design Automation Conference*, 2006, pp. 849–852.

[90] P. Lotfi-Kamran, A. M. Rahmani, M. Daneshtalab, A. Afzali-Kusha, and Z. Navabi, "EDXY – A low cost congestion-aware routing algorithm for network-on-chips," *J. Syst. Archit.*, vol. 56, no. 7, pp. 256–264, 2010.

LIST OF PUBLICATIONS

Published Journal Papers

[1] A. Punhani, P. Kumar, and Nitin, "Optimal extra links placement in mesh interconnection network using improved environmental adaptation method," *J. Intell. Fuzzy Syst.*, vol. 32, no. 5, pp. 3285–3295, April 2017. SCIE and Scopus Indexed

[2] A. Punhani, P. Kumar, and Nitin, "A Modified Diagonal Mesh Shuffle Exchange Interconnection Network," *Int. J. Electr. Comput. Eng.*, vol. 7, no. 2, pp. 1042–1050, April 2017. Scopus Indexed

[3] A. Punhani, P. Kumar, and N. Nitin, "Routing for Center Concentrated Mesh," *Int. J. Intell. Eng. Syst.*, vol. 10, no. 1, pp. 86–94, Feb 2017. Scopus Indexed

[4] A. Punhani, P. Kumar, and Nitin, "Diagonal Connected T Mesh," *Indian J. Sci. Technol.*, vol. 9, no. 32, pp. 1–7, August 2016. Scopus Indexed.

[5] A. Punhani, P. Kumar, and Nitin, "Level Based Routing Using Dynamic Programming for 2D Mesh," *Cybernetics and Information Technologies*, vol. 17, no. 2, June 2017. Major Indexing: Scopus, ESCI

[6] A. Punhani, P. Kumar, and Nitin, "Three Dimensional Topology based on Modified Diagonal Mesh Interconnection Network," *Journal of Telecommunications, Electronics and Computer Engineering*, vol. 9, no. 3-6, pp.1-6, 2017. Major Indexing: Scopus.

Published Conference Papers

[7] A. Punhani, Nitin, and P. Kumar, "A Modified Diagonal Mesh Interconnection Network," in Proc. *2014 Annual IEEE India Conference (INDICON)*, 2014, pp. 1–6. Scopus Indexed.

[8] A. Punhani, P. Kumar and Nitin, "A Horizontal Fat Mesh Interconnection Network," in IC3, JIIT Noida, 2017, pp. 1–6. Scopus Indexed.

Accepted Journal Paper

[9] A. Punhani, P. Kumar, and Nitin, “Entropy based XY routing Algorithm”, International Journal of Grid and Utility Computing, Inderscience Major Indexing: Scopus.