ON PERFORMANCE OF MODIFIED TORUS INTERCONNECTION NETWORKS

Thesis submitted in fulfillment for the requirement of the Degree of

Doctor of Philosophy

By

DINESH KUMAR

Department of Computer Science & Engineering and Information Technology
Jaypee University of Information Technology
Waknaghat, Solan-173234, Himachal Pradesh, INDIA

May 2019

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY
(Established by H.P. State Legislative Vide Act no. 14 of 2002)
Waknaghat, P.O. Waknaghat, Teh.- Kandaghat,
Distt. Solan – 173234 (H.P.) INDIA
Website: www.juit.ac.in
Phone No. (91) 01792-257999
Fax: (91) 01792 245362

CANDIDATE'S DECLARATION

I hereby declare that the work reported in the Ph.D. thesis entitled "ON PERFORMANCE OF MODIFIED TORUS INTERCONNECTION NETWORKS" submitted at Jaypee University of Information Technology, Waknaghat, Solan (HP), India is an authentic record of my work carried out under the supervision of Dr. Vivek Kumar Sehgal and Dr. Nitin. I have not submitted this work elsewhere for any other degree or diploma. I am fully responsible for the contents of my Ph.D. thesis.

Dinesh Kumar

Dinesh Kumar
(Enrollment No.: 126201)
Department of Computer Science & Engineering
Jaypee University of Information Technology, Waknaghat, Solan (HP), India
Date: 30-05-2019

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY
(Established by H.P. State Legislative Vide Act no. 14 of 2002)
Waknaghat, P.O. Waknaghat, Teh.- Kandaghat,
Distt. Solan – 173234 (H.P.) INDIA
Website: www.juit.ac.in
Phone No. (91) 01792-257999
Fax: (91) 01792 245362

SUPERVISOR'S CERTIFICATE

This is to certify that the work reported in the Ph.D. thesis entitled "ON PERFORMANCE OF MODIFIED TORUS INTERCONNECTION NETWORKS", submitted by Dinesh Kumar at Jaypee University of Information Technology, Waknaghat, Solan (HP), India is a bonafide record of his original work carried out under my supervision. This work has not been submitted elsewhere for any other degree or diploma.

Vivek Sehgal

Dr. Vivek Kumar Sehgal
Associate Professor
JUIT, Solan
Date: 30-05-2019

Nitin

Dr. Nitin
Associate Professor
Indian Institute of
Management (IIM), Shillong
Date: 30-05-2019

ACKNOWLEDGEMENT

I express my warm gratitude and profound respect to my supervisor Dr. Vivek Kumar Sehgal, Associate Professor and Ph.D. coordinator, Department of Computer Science & Engineering and IT, Jaypee University of Information Technology, Waknaghat, Solan, Himachal Pradesh, India, under whose valuable guidance the work has been completed successfully. I am also thankful to my co-supervisor, Dr. Nitin, Professor of EECS at University of Cincinnati, Cincinnati, OH, US for his active guidance. One can learn a lot from him and I found his guidance as a golden opportunity to me. The author would like to express special thanks to Dr. S.P. Ghrera, Professor and Head, Department of Computer Science & Engineering and IT, Jaypee University of Information Technology, Waknaghat, Solan, Himachal Pradesh, India, for motivating and encouraging me for every crucial moment.

I am also thankful to my DPMC members Dr. Rajesh Kumar, Assistant Professor (Senior Grade), Department of Physics and Material Science, Dr. Hemraj Saini, Associate Professor, Department of Computer Science & Engineering and IT, and Dr. Ekta Gandotra, Assistant Professor (Senior Grade), Department of Computer Science & Engineering and IT for providing their valuable suggestion.

I am also grateful to Professor (Dr.) Vinod Kumar, Vice Chancellor of Jaypee University of Information Technology and Professor (Dr.) Samir Dev Gupta, Director and academic head for providing an inert research environment.

I take the opportunity to thank my colleague Akash Punhani who helped and motivated me during my thesis work.

I am very thankful to lab staffs to provide better experimental environment and equipment which helped me to accomplish my simulation work.

I am also thankful to my parents, my elder brother Mahendra Kumar Sahni and sister Chandan Sahni for their love, care and moral support for me.

Finally, I thank to eminent reviewers of my research work and every person from whom affection, help, guidance, support, and inspiration during my research work.

TABLE OF CONTENTS

ABSTRACT

Parallel processing systems are the most powerful tools to real time applications that requires large processing of data. In recent years of advancement within the technology, a remarkable transition has been observed. Important components of these systems include; processing elements, memory modules and interconnection networks. The main purpose of interconnection network is to build up connection between these components, as it corresponds to the performance of parallel computing system. Three attributes characterize an interconnection network namely: topology, routing algorithm and flow control. On the other hand, a topology is differentiated by its topological properties such as degree, diameter, average distance, bisection width, scalability and fault tolerance. In terms of exchanging data among the components of parallel processing system, several mechanistic approaches are used. Although it is simple to use and implement, the main issue is its fault tolerance. Pertaining to the various problems and research gaps such as communication system, scalability, efficiency etc. within the interconnection network, topologies, and related factors have been discussed. Briefly, the objectives of the present thesis were to design efficient topologies with improved properties. We have proposed four topologies such as modified X-Torus, Center concentrated X-Torus, Hexagonal X-Torus and Modified Diagonal Torus in this thesis. All these topologies were tested using OMNeT++ simulator under various traffic patterns such as uniform, bit complement, neighbor, tornado and hotspot. First three topologies are compared with Mesh, Torus, and X-Torus and lastly with Mesh, Torus, D-Mesh and D-Torus. The quality of services which were obtained from simulation process were Average Throughput, Average End to End Delay and Average Hop Count.

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1
INTRODUCTION

1.1 Interconnection Networks

Multiprocessor system is comprised of several processing elements, numerous I/O modules and memory modules. Every processor can communicate with any of the I/O modules and memory modules and are connected by interconnection networks. Especially, an interconnection network plays a decisive role in exchange of the information among numerous processors in the networks. However, at present there is an immense computational task going on in various areas likely; virtual reality, scientific calculation, weather forecasting and numerical modelling that could not be effortlessly resolved in appropriate time duration with currently available computer systems. Globally, with the immense growth of the technology, the computer system has become speedy and complicated. In addition, new elements got added to the system, known as Intellectual Properties (IP) [1–3]. These are convenient functional blocks which are used in big hardware design such as processors, memory modules, Universal Serial Bus (USB) cores, Digital Signal Processing (DSP) cores, and digital processing units. Earlier systems faced memory blockage problem which was one of the main reasons of communication delay among processors. Therefore, interconnection network is one of the best possible approach to improve the overall performance of the network by increasing the computation speed via various processors working at the same time [4, 5]. It is placed between various components in the digital systems. Initially, interconnections networks were developed for multiprocessor systems. The main advantage of interconnection networks is that it focuses on shared router node rather than creating dedicated links between each terminal node. Additionally, it is also beneficial to get high fault tolerance, more scalability, and high throughput. Throughout the years, consistent efforts have been made by analysts and researchers to achieve optimal performance of the system [6–8]. Any interconnection networks depend on following key aspects [1].

- Topology
- Routing Algorithm
- Flow Control mechanism

1.1.1 Topology

Topology is one of the main aspect of interconnection networks that describes the arrangements of nodes and links in the network. Mathematically, it is represented by a graph G (V, E) where V is the collection of all vertices and E contains all edges. Particularly, there are two types of network topologies. First one is static and other one is dynamic. In static network, there is a fixed link among nodes where nodes can be memory modules, processing modules, any I/O modules, or combination of these modules. With a static network, communication links among nodes are fixed and permanent, for example, bus, ring, mesh, and torus, etc. Dynamic networks support reconfigurable links among nodes and this feature is achieved by using switch boxes in the network. The direct and indirect topology are the two main classifications of the topologies that are utilized in interconnection networks. Benes network is one of the example of dynamic network. Interconnection networks are also classified as direct topology and indirect topology. In direct topology, each node is directly connected to the adjacent nodes however, in indirect topology, nodes relate to the help of switches such as torus (direct topology) and crossbar (indirect topology). In this present study, torus and its variants belonging to direct topology have been studied and discussed.

1.1.2 Routing Algorithm

This is known that there may be single or various routes between source node and destination node in the topology. The routing algorithm decides the shortest route to send data packets from source node to destination node. Here, routes are defined as the series of communication links, through which data packet follows during traversal. It has an impact over various features of interconnection networks such as adaptivity, fault tolerance, livelock, deadlock, and connectivity.

1.1.2.1 Classification of Routing Algorithm based on the route

It is mainly classified into three groups which are as follows:

1.1.2.1.1 Deterministic Routing Algorithm

Routing algorithm selects single fixed route between every source destination nodes irrespective of the condition of the network whether it is loaded or not. It is very simple to design deterministic routing algorithm and quite easy to implement. The major advantage of deterministic routing algorithms is that it is deadlock free, although it creates load imbalance that affects the overall performance of the system.

1.1.2.1.2 Oblivious Routing Algorithm

In oblivious routing algorithm, selection of path is decided by the addresses of source node and destination node. Concurrently, it does not depend on current state of the network. Deterministic routing is unable to distribute the load which is also the subset of oblivious routing algorithm, although in oblivious routing algorithm, it is achieved by sending data packets to some other node and finally attaining destination node. Therefore, due to this situation, it can traverse some extra links to reach the destination nodes.

1.1.2.1.3 Adaptive Routing Algorithm

Adaptive routing algorithm, in contrast to oblivious routing, depends upon the network condition, which affects the overall performance of the network. The major advantage of this algorithm is that it returns multiple paths based on current state of the network and therefore reduces the packet delivery failure and improves network reliability.

1.1.2.2 Classification based on the routing algorithm

The data structure plays a substantial role in order to implement routing algorithm, therefore, considering this factor routing algorithm is further divided into two parts. First one is table based routing algorithm, while another is finite state based routing algorithm:

1.1.2.2.1 Table based Routing Algorithm

It is necessary to store the route information between nodes to transfer packet from source node to destination node. Therefore, memory is required to store it which is directly proportional to the total number of nodes in the networks. It also affects the scalability and cost of the system which are the vital components of the network.

1.1.2.2.2 Finite State based Routing Algorithm

The main objective of this algorithm is to reduce memory consumption. The routes calculation is done based on mathematical or logical calculation to send data packets from source node to destination node.

1.1.2.3 Flow Control Mechanism

Flow Control is the integral component of the interconnection networks. It provides data rate synchronization between source node and destination node. Some of the flow control mechanism are given below:

1.1.2.3.1 Bufferless Flow Control Mechanism:

It is used to save power dissipated by router buffer and achieved by dropping packets.

- Stop and Continue Mechanism: In this method receiver node sends a stop signal to the sender to prevent sending of data and transmitting of packet from sender node started when it receives continue signal.
- Credit Based Flow Control Mechanism: In this mechanism sender node can transmit the packets only if it has sufficient credits. If it does not, it will not be able to send data packets. Furthermore, the number of flits of data packets transmitted in a specific direction will be equal to credit associated with the router sending the flits.

1.2 Problem Statement

Globally, the present interconnection networks play an important role in processors of computer systems. Fundamentally, they are used to minimize communication delay rather than calculation. Therefore, accessing of network resources become faster day by day. In this present thesis work, we have found and reported that interconnection networks mainly depend on three key features. First one is topology design, second is routing algorithm and lastly flow control mechanism. Empirically, most researchers focused on either topology design or routing algorithm and some studied on flow control mechanism. Recent advancement in the processing speed of various supercomputers focused on the design of topologies. According to the November 2017 edition of the Top500 and Green500 China's super computers, like Sunway TaihuLight with processing speed 93.01 petaflops and Tianhe-2 (Milky Way-2) with processing speed 33.86 petaflops ranked at first two positions. These computers contain various processor cores which are bind up with interconnection networks. An important factor which affects the overall performance of the system depends on how these processors are connected to each other. In addition, it is well known that system performance is mostly affected by communication delay rather than computation because system buses connect the various IPs. To overcome this problem, NOCs are used to provide communication between IP cores. Therefore, topology deign becomes key factor for determining overall system performance and helps to achieve high fault tolerance, faster communication, minimal delay and maximal throughput. The literature survey reveals that every topology is affected by scalability properties to improve the performance of system as network size increases. Also, it is frequently seen that network get congested over time and probability of network failure increases. Therefore, there is a need to improve this prominent feature. and to overcome network failure issue to increase path diversity. It is also observed that network delay depends on links between sender and receiver node, therefore, addition of smart links should be focused or considered to achieve minimal delay. An efficient network requires shorter diameter [9], higher bisectional bandwidth etc. which should also be considered while designing topology.

1.3 Contributions

This present thesis contains various key contributions to study and design of smart topologies which helps to improve overall performance of network. Based on above discussion, the research problem was divided into four stages, with following objectives of the study:

Stage I

- Study of various topology of interconnection networks.
- Findings of topological properties.
- Learning of simulator to verify our results.
- Analysis of various traffic patterns to see its effects on topologies.
- Exploring the effects of routing algorithm on the performance of network topology.
- Proposed a topology which is node symmetrical and better topological properties.

Stage II

- Study of existing topology of interconnection networks.
- Findings the limitation of other topologies.
- Proposed new topology having high path diversity.
- Simulated our topology on various traffic patterns and find optimal results.

Stage III

- Identification of lack of smart edges in the existing topologies.
- Proposed new topology and try to reduce internodal distance.
- Simulated using Omnet++ simulator.

Stage IV

- Study of existing topologies and find that minimum hop is one more key issue which affects performance of network.
- We have proposed new topology which takes lesser hops to move packets to destination node.
- Simulated our work under various traffic patterns.

1.4 Outline of the Thesis

The thesis is divided into seven chapters. CHAPTER 1 presents Introduction. CHAPTER 2 presents background of torus and its variants in details. CHAPTER 3 presents detailed insight of Modified X-Torus topology to provide node symmetrical structure of the network. CHAPTER 4 presents Center Concentrated X-Torus topology which is improved version of X-Torus topology and gives better throughput and minimal delay. CHAPTER 5 presents the Hexagonal X-Torus topology which takes less hops to moves packets in the network. CHAPTER 6 presents the design and result analysis of Modified Diagonal Torus topology. Finally, CHAPTER 7 covers the conclusion of the thesis supported with the simulations result of experiments backed by the future scope of the research work.

CHAPTER 2
PRELIMINARIES AND BACKGROUND

2.1 Interconnection Networks

Interconnection networks are principally used to provide fast communication among parallel machines. There are several factors which affects the design of interconnection networks such as number of processors, scalability of network, partitionability, simplicity, reliability and nature of workload etc. In multiprocessor computer system, there are various processing elements, I/O modules and memory modules where every processor can access any of the I/O unit and memory modules. Per the Flynn's taxonomy, parallel computer systems are broadly categorized into four parts that are based on communication between processor and memory modules. An immense advancement of the technology has lead the concept of distributed parallel system into existence, where each individual processor is connected to local memory via interconnection network known as node. In view point of present time scenario and to obtain more parallelism, it is necessary to integrate more transistor on the chip in an appropriate manner [10]. Several researches and scientists proposed different topologies including mesh, torus, and x-torus etc. to get minimal communication delay in interconnection networks.
William J Dally and Brain towels suggested a new architecture, known as the tile based architecture in which a fixed rectangular area is reserved for IP cores and the rest part is for global wiring which are helpful in achieving feasible communication [11, 12]. Shared-medium, Direct and Indirect networks are the major three classifications of interconnection networks.

Definition 1: Shared-Medium Networks: In this network, communication channel is shared by all communication nodes.

Definition 2: Direct Interconnection network: In this network, each node is directly connected to subset of other nodes within the network.

Definition 3: Indirect Interconnection network: It is also called as switch-based network. In this type of network, nodes are not directly connected to each other, however, communication between them is carried by switches. Conventionally, the direct interconnection network is described by a graph G (V, E), where ‘V’ is the set of nodes and ‘E’ is the set of communication links.

2.2 Topological Properties of Interconnection Networks

A total of four fundamental network properties which certainly improve the network performances include; network degree, diameter, bisection width and bisection bandwidth, and are defined as below:

Definition 4: Network degree: It is the number of nodes which are directly connected to a node within the network. This is the measure of network cost and nodes I/O complexity.

Definition 5: Network diameter: It is the shortest distance between two nodes which are at longest position in the network. It is useful for calculating lower bounds of various algorithms such as broadcasting, and sorting etc.

Definition 6: Bisection Width: Bisection width of the network is defined as, a line drawn that divides network into almost equal two halves and numbers are linked which are cut/bisect. It is also useful in finding lower bounds of sorting algorithm.

Definition 7: Bisection Bandwidth: It is the bandwidth of the bisection width.

Topology plays a key role to improve the overall performance of the network. On demand of application, several topologies have been designed and are illustrated one by one.

2.2.1 Mesh Topology

This is the basic and simplest topology of interconnection networks. It is represented by a graph, G= (N, E), where N is the set of nodes and E is the set of links. There are few nodes which include degree 2, some others have 3 and 4, respectively. The diameter of mesh topology is 2n-2 and bisection width is n. Figure 2.1 represent simple mesh topology. Let us assume that the coordinates of any two nodes are X(i,j) and Y(p,q) such that $0 \leq i,j,p,q \leq n$ respectively. These two nodes are connected if it satisfies the following condition:

$$|p - i| + |q - j| = 1 \tag{2.1}$$

Globally, mesh topology is utilized for networking purpose because it is simple, easy to implement, cost effective and reliable. In addition, the famous laptop XO-1 which is also called One Laptop per Child (OLPC) designed by MIT's Media lab mesh networking is known for its robustness and cost-effective nature. It is also used in FabFi to provide online education, promoting social networking as well as medical solutions. Moreover, it is frequently used in IoT application and, Intel's Skylake-X and Skylake-SP processor employ mesh architecture. TRIPS is also one of the example that uses mesh networking for its chip design architecture [13], Tilera [14] etc. 2D mesh topology is used by Intel Paragon machine while J-Machine uses 3D mesh topology.

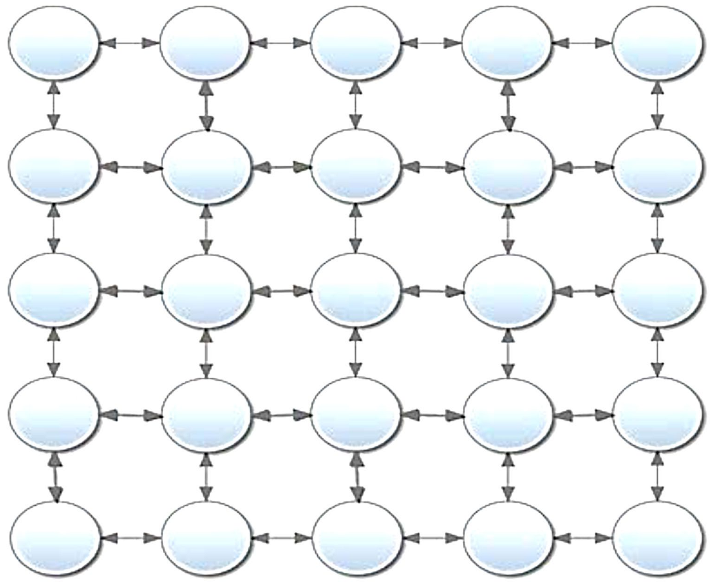

Figure 2.1 Mesh Topology

2.2.2 Torus Topology

It is an improved version of mesh topology. Comparatively, mesh topology had a problem of corner node leading to different characteristics in comparison to center node. This problem was resolved in torus topology by making extra link between corners nodes. It includes features like symmetrical in nature, lower diameter, higher bisection width and more degree in comparison to mesh topology. In torus topology, extra links are used to connect corner nodes which helps to provide better path diversity and fault tolerance. Diameter is reduced, and bisection width is increased in comparison to mesh topology. The degree of torus topology is 4 for all nodes while diameter and bisection width are n and 2n, respectively. Machines like DASH [15] and HP GS1280 [16] uses 2D torus topology while CRAY3D, CRAY T3E [17] uses 3D torus and IBM Blue Gene/Q [18] use 5D torus architecture and is presented in Figure 2.2.

Nodes X(i,j) and Y(p,q) such that $0 \leq i,j,p,q \leq n$ are connected with the following condition:

$$|p - i| + |q - j| = 1 \tag{2.2}$$

$$i = 0, j = q, p = n - 1 \tag{2.3}$$

$$i = p, j = 0, q = n - 1 \tag{2.4}$$

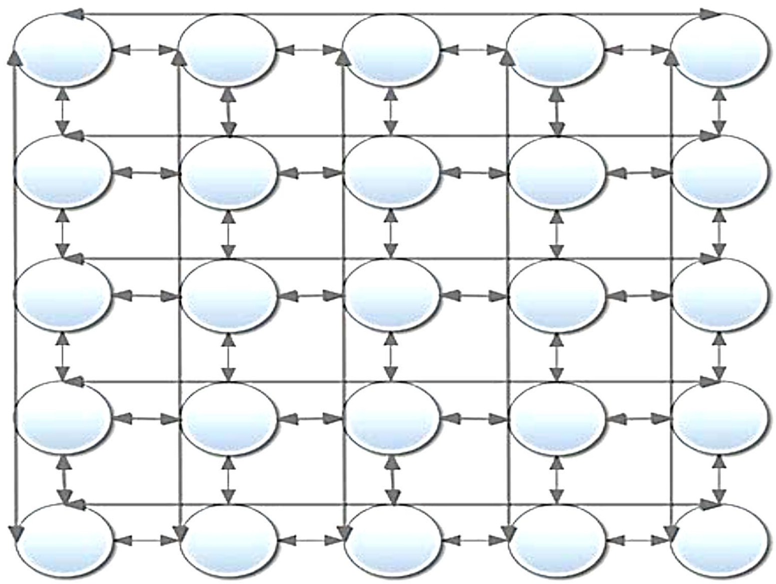

Figure 2.2 Torus Topology

2.2.3 XMESH Topology

XMESH [19] is an improved version of mesh topology which contains the horizontal adding links of TMESH. XMESH contains diagonally crossed channels instead of the vertical channels. This topology provides symmetry, regularity and high connectivity and is suitable to implement massive parallel system. The degree of XMESH topology is 6 and its diameter is 2n while bisection width is also 2n. In this present study, authors considered 3 topologies likely; XMESH, DMESH and TMESH, and simulated it and observed that XMESH minimal average delay to send the packets from source node to destination node. The mean internode distance against various number of nodes such as 10, 100, 1000 and 10000 was compared, and attributed that XMESH provides smaller distance. XMESH topology has been presented in Figure 2.3. A node represented by coordinate (x, y) is connected to other nodes having coordinates such as (g(x+1), y), (g(x- 1), y), (g(x-1), g(y-1)), (g(x+1), g(y+1)) in even parity topology. On the other hand, if the topology is odd parity, it will connect to the nodes (g(x+1), y), (g(x-1), y), (g(x-1), g(y+1)), (g(x+1), g(y-1)). The function g(x) is defined by Equation 2.5.

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

Above function represents the links which are used to make connection between nodes. Here 'k' is the parity of topology.

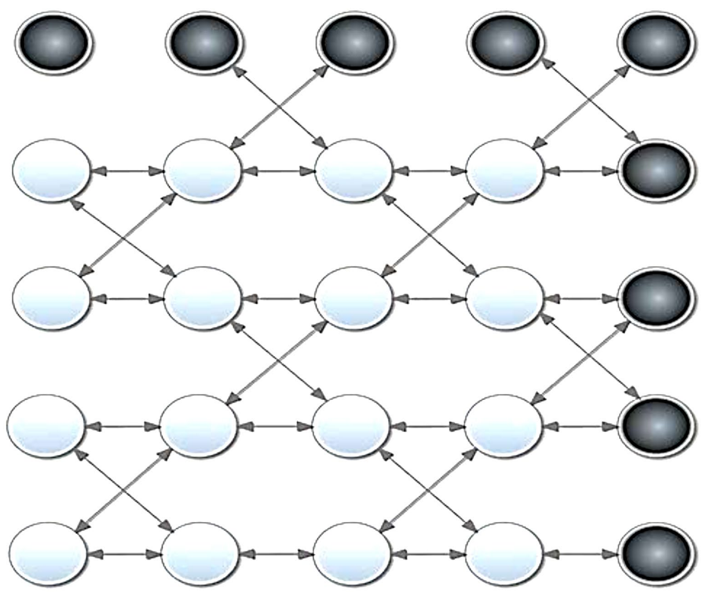

Figure 2.3 XMESH Topology

2.2.4 D-Mesh Topology

D-Mesh topology is an improved version of mesh topology and is an area efficient topology. This was designed with an aim to keep the implementation cost feasible [20]. In this topology, authors used wormhole packet switching technique, containing additional features which helps to improve network performance. Comparatively, mesh topology contains no diagonal links while in D-Mesh it exists which affects topological properties including; degree, diameter and bisection width. D-Mesh topology is presented in Figure 2.4. where degree of nodes is different. Some nodes have degree 3, and few one has 5 and remaining node have degree 8 as per their connection in the topology. Diameter of D-Mesh was calculated as n-1 with bisection width of 3n-2. In this topology, connection between any two nodes X and Y, which have node (i, j) and (p, q) were connected, satisfying Equation 2.6 and Equation 2.7:

$$|p - i| + |q - j| = 1 \tag{2.6}$$

$$|p - i| + |q - j| = 2 \tag{2.7}$$

In this study authors compared the performance with NePa network using eNoC simulator under 4 traffic patterns like Random, Bit Complement, Bit Reverse and Matrix Transpose, and revealed that D-Mesh has smaller internodal distance compared to NePa network. In addition, suggesting that this topology provides better fault tolerance facility in the network.

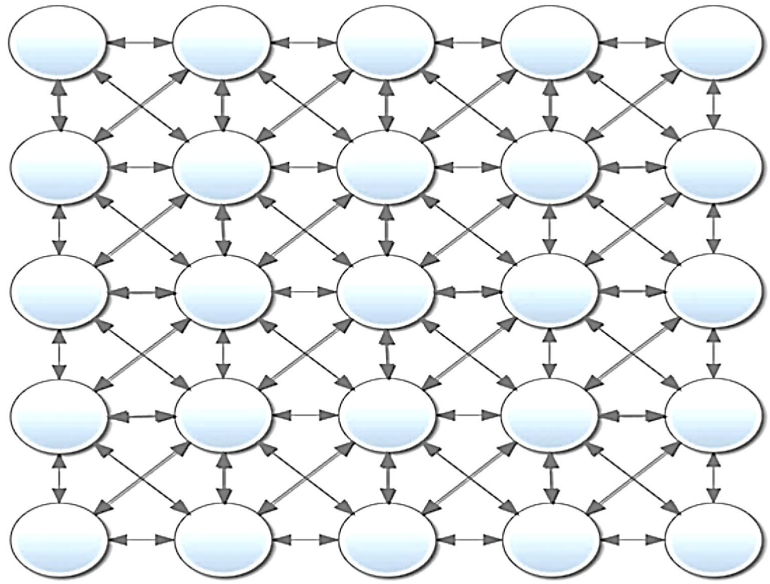

Figure 2.4 D-Mesh Topology

2.2.5 T-Mesh Topology

T-Mesh topology is defined as an extended version of Mesh topology where four additional extra smart links connect the corner nodes [21]. In comparison to Mesh topology, additional connection reduces the diameter of topology, providing improved results. Herein, the degree of nodes is 3 and 4, respectively. The diameter of this topology is found to be ‘n’ for odd parity and n-1 for even parity topology. The bisection width is also affected and found to be n+2. This study revealed that average delay and average hops in T-Mesh topology is smaller than mesh topology which is using gpNoCsim simulator and is presented in Figure 2.5.

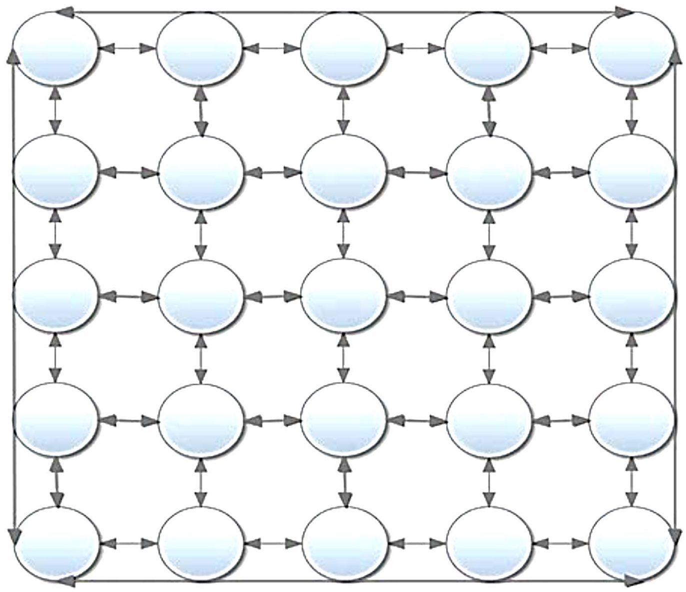

Figure 2.5 T-Mesh Topology

2.2.6 Twisted Torus Topology

Twisted torus topology is another variant of torus topology, which is the improved version of mesh topology. However, torus topology and twisted torus differ only by the arrangement of extra links [22]. Present study aimed to remove computer network bottleneck issues using twisted links. In comparison to simple torus topology, twisted torus has degree 4 and diameter n-1 has and includes the same bisection width which is 2n. The simulation was performed using INSEE network simulator and compared it using synthetic traffic patterns along with trace-driven traffics. Several traffic patterns were taken into consideration such as, Bit-Reversal, Bit-Complement, and Perfect Shuffle. Authors revealed that the twisted torus topology provides better results in comparison to mic-radix tori and provides maximum utilization of channel bandwidth. Figure 2.6 represented the layout of twisted torus topology.

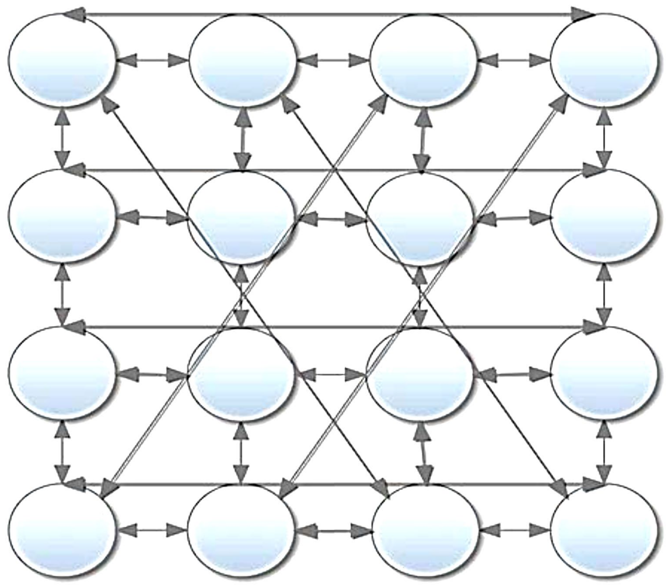

Figure 2.6 Twisted Torus Topology

2.2.7 Folded Torus Topology

The folded torus network is another variant of torus Network with a difference of using uniform links instead of non-uniform links to connect the nodes [23]. In this study, authors aimed to reduce energy consumption and improve the performance of the system. In folded torus, the degree of each node is 4 and diameter is n while bisectional width is same as torus topology i.e. 2n. For simulation purpose authors considered three other networks likely LBDR, TriBA and RAW. Experimental results revealed that Folded Torus network consumes less power per hop count when compared to another network. Present study utilizes the network size of nodes of 16, 25, 36, 49, and 64 and calculated average hop count using same network size as well as topologies, attributing that Folded Torus took lesser hop count in comparison to LBDR, TriBA and RAW networks, as shown in Figure 2.7.

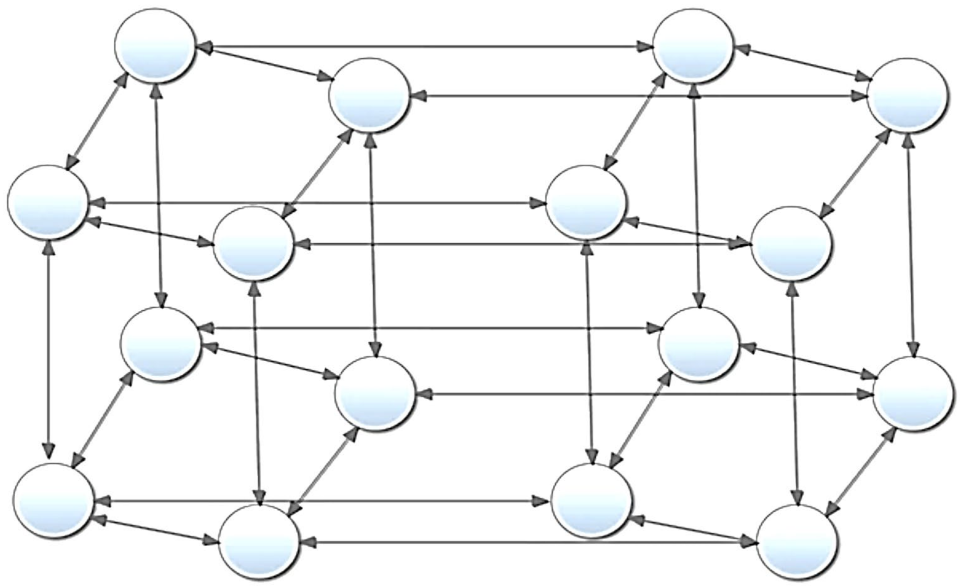

Figure 2.7 Folded Torus Topology

2.2.8 Diagonal Connected Mesh Topology

Diagonal Connected Mesh is another variant of Mesh topology which is used in Mesh connected crossbar to make links between nodes which reduces the wire density [24]. It provides better performance via extra toroidal link instead of normal links in mesh. Another advantage is greater bisectional width to overcome sooner congestion. This topology includes, nodes of degree 3, few containing 4 and rest with degree 5. Diameter of Diagonal Connected Mesh topology is found to be n-1 and bisection width is n-1. The main goal of designing topology is to achieve static topological characteristics like smaller diameter, high bisection bandwidth, minimum inter nodal distance and reliability. Since it has smaller diameter, so it could provide better performance in the overall network system and is presented in Figure 2.8.

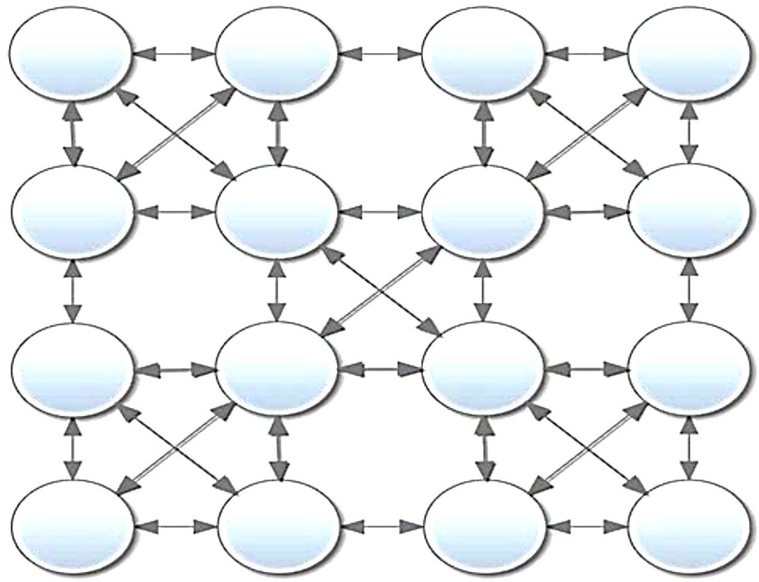

Figure 2.8 Diagonal Connected Mesh Topology

2.2.9 Xmesh Topology

Xmesh is a modification of Mesh topology [25]. In this topology, diagonally placed corner nodes are connected by two different links. This study aimed to design this topology to achieve higher bisection width and reduced average distance. This is helpful when one link is much loaded and data packets can move through another link. Herein, the degree of nodes is 3 and 4 and the diameter of this topology is n-1 while bisectional width is n+4. In this study, authors performed simulation analysis by Popnet simulator and topologies by using Torus, mesh and Xmesh. This was performed under the uniform and hotspot traffic. Simulation results revealed that Xmesh and Torus topology utilizes less average latency compared to Mesh topology. This present topology is presented in Figure 2.9.

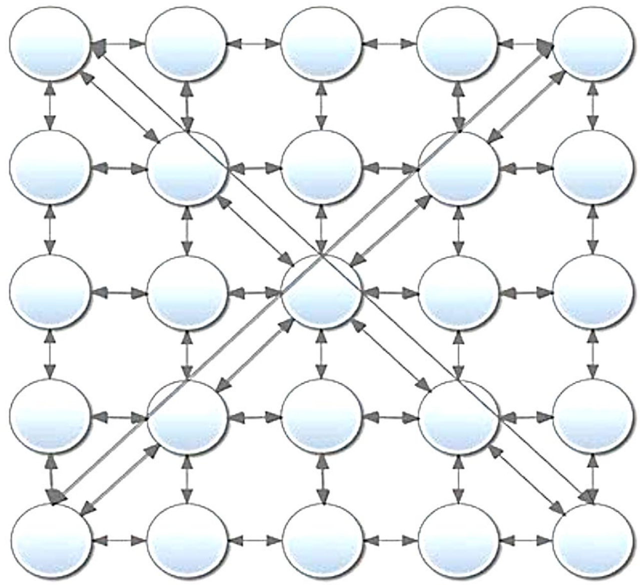

Figure 2.9 Xmesh Topology

2.2.10 C-Mesh Topology

C-Mesh is principally the acronym of Concentrated Mesh where each switch is connected to 4 processing units [26]. It helps to reduce hop count in the network. Due to high processing unit, it increases load on the communication channel and router. This topology does not handle congestion because of less path diversity. Therefore, it is considered to be less reliable in comparison to mesh and its other variants. However, it is considered for the application function which requires low traffic of data. This topology is presented in Figure 2.10.

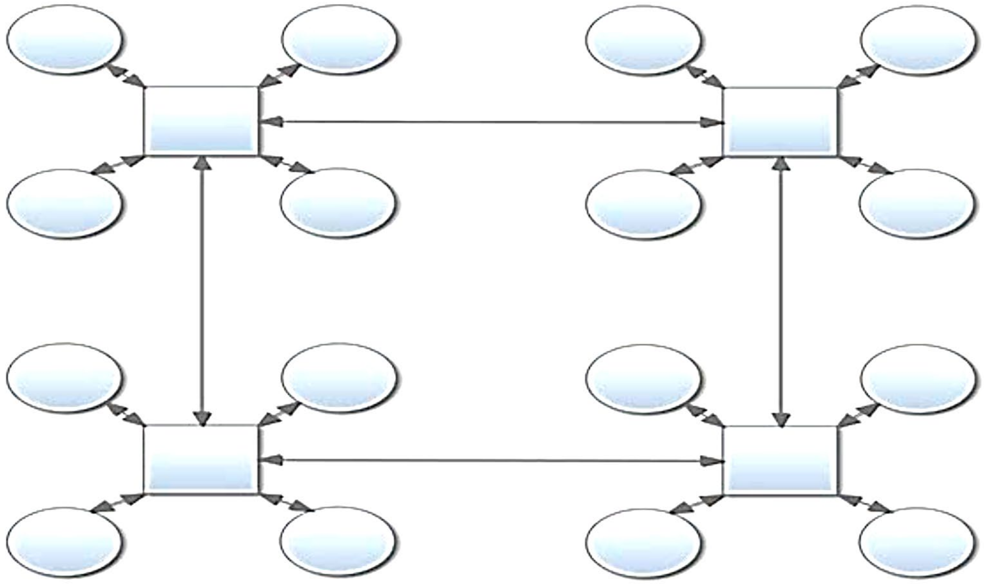

Figure 2.10 C-Mesh Topology

2.2.11 NR Mesh Topology

NR Mesh topology is an improved version of C-Mesh topology [27, 28] and also known as Nearest neighboR Mesh. In this topology, every processing unit is connected to four adjacent routers. During transmission of data, if one router gets fail then other router element attached to that processing unit can be used for data transmission. This topology provides better fault tolerance in the network and provides option to the corner nodes to send the data packet to the adjacent routers which reduces hop count and results in low latency. This study was aimed to compare NR Mesh with 2-D Mesh topology to find energy consumption and execution time using SIMICS/GEMS simulator. Herein, the degree of processing unit is 4 while the degree of router is 8. The bisection width of NR Mesh topology is 3n-1 while its diameter is 2n-4. Figure 2.11 presents the present NR Mesh topology.

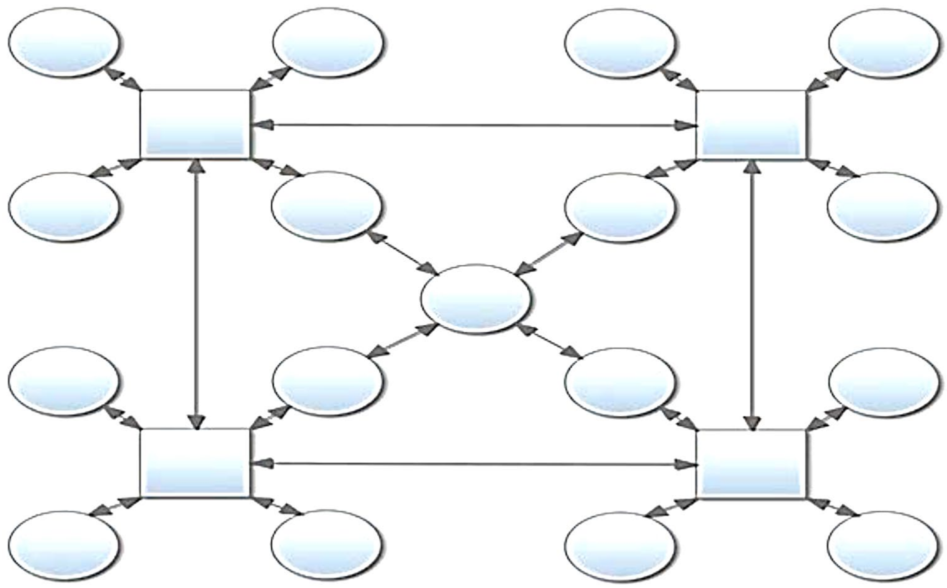

Figure 2.11 NR Mesh Topology

2.2.12 Double Loop (2m) Topology

This topology is the improved version of Ring topology where two rings are used in its designing [29]. This topology is scalable, symmetric and simple in design. Here the numbers of links for each ring is 2m and every node is connected to 3 other nodes. In addition, the degree of each node is 3. The diameter of Double Loop (2m) topology is m+1 while its bisection width is 4. In this study, authors aimed to reduce node degree, decreasing communication channel and reusing router nodes, suggesting that this topology results better performance in comparison to ring topology. Authors have simulated this topology with Ring and Mesh Topology under uniform traffic and is better for small size network. This topology is shown in Figure 2.12.

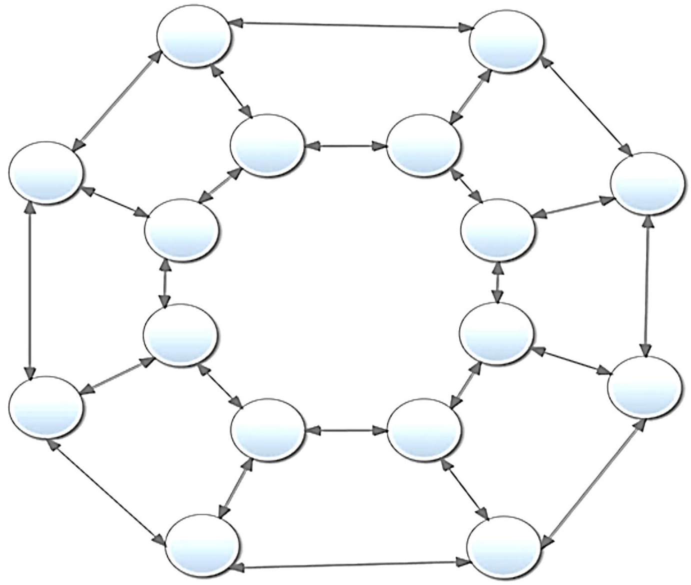

Figure 2.12 Double Loop (2m) Topology

2.2.13 Cross bypass Mesh Topology

This topology is defined as the combination of two existing topologies, containing C2 mesh and 2DDgl mesh [30]. It is well suited to design large size network with the degree of Cross Bypass Mesh Topology is 9 and diameter is n-1. Due to higher degree, it includes more path diversity which proves more reliability and fault tolerance within the network. Also, its bisection width is 2n+1 for odd parity network and 2n for even parity. Authors utilized 8 topologies to compare its cost, performance and topological properties and this topology is presented in Figure 2.13.

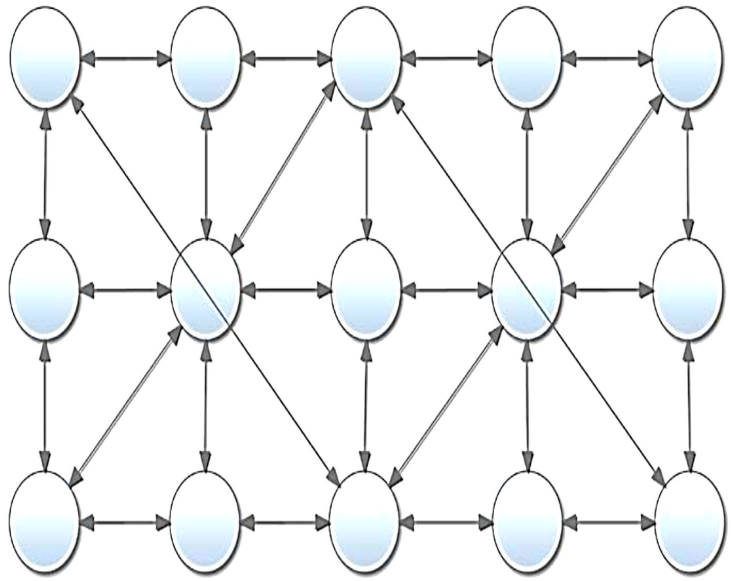

Figure 2.13 Cross bypass Mesh Topology

2.2.14 Shortly Connected Mesh Topology

This is another variant of Mesh topology where diagonals links are used by some particular nodes that are known as bridge nodes and some specific nodes, called as intermediate nodes used to make diagonal links [31]. Basically, it uses the symmetric nature of Mesh topology to design it. In this topology, the degree is 8 and, diameter and bisection width are n-1. The aim of designing this topology was to find shortest path to send packets from source node to destination node. Authors evaluated its performance using various topologies to find wiring complexity, packaging area overhead, network latency, throughput, cost. This topology is represented in Figure 2.14.

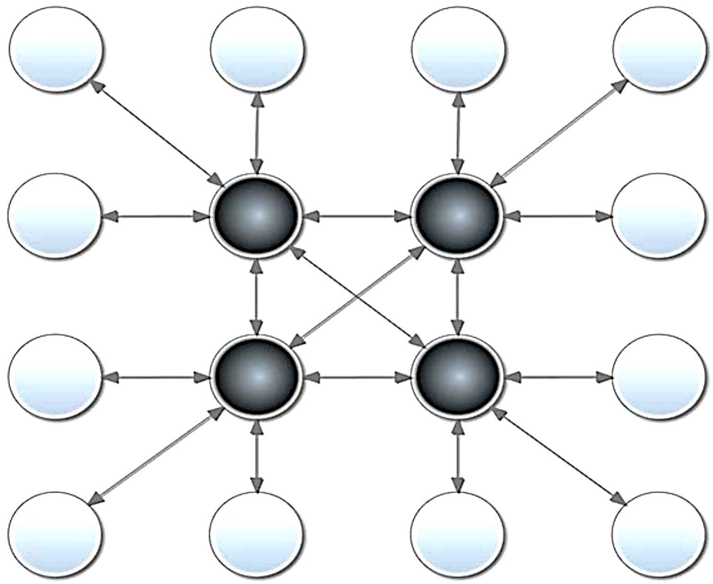

Figure 2.14 Shortly Connected Mesh Topology

2.2.15 Honeycomb Mesh Topology

It is defined as a special topology where the length of each edges is constant. The aim of designing this topology is to reduce network cost. In this topology node degree are in different pattern; some nodes have degree 2 and others have degree 3. The main feature of this topology is its planar nature. The diameter of Honeycomb Mesh topology is observed to be 1.63n which is much better for an efficient topology [32]. The other topological features of this topology are its bisection width which are 0.83n. In this study, a comparative evaluation of this study was performed with Mesh and Torus topology. The results of this study revealed that there is 40% cost reduction in compare to Mesh topology. This topology layout is shown in Figure 2.15.

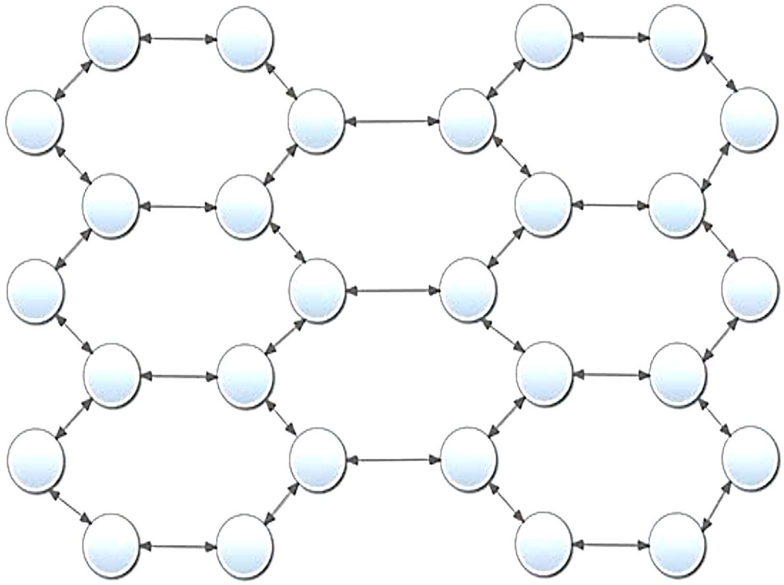

Figure 2.15 Honeycomb Mesh Topology

2.2.16 Honeycomb Rhombic Mesh Topology

Honeycomb Rhombic Mesh topology belongs to the family of Honeycomb Mesh [32]. The main similarity between both is about degree, including degree 3. The topological parameter which is affected is diameter of this topology, i.e. 2.83n. In addition, there is a disadvantage in bisection width over Honeycomb Mesh, which is 0.71n. Figure 2.16 presents the layout of this topology.

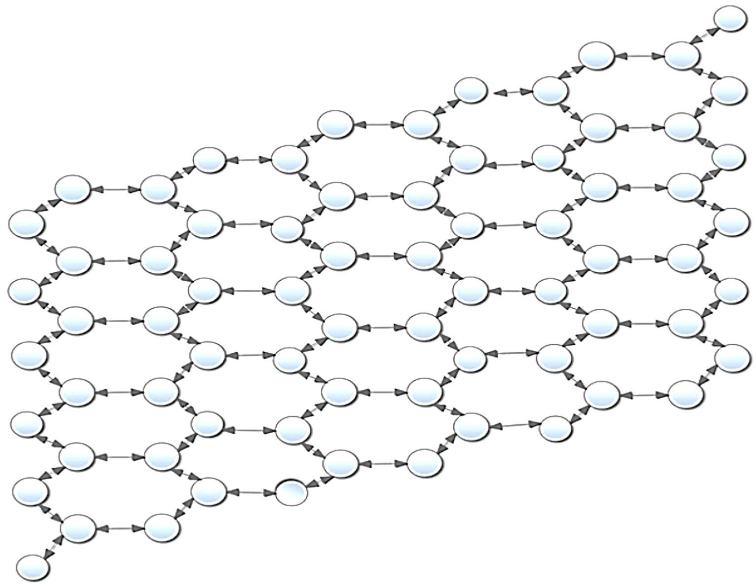

Figure 2.16 Honeycomb Rhombic Mesh Topology

2.2.17 Honeycomb Square Mesh Topology

In this topology, nodes are arranged in such a way that they look to form hexagons (Figure 2.17). The degree of each nodes is 3 and diameter is slightly larger in comparison to other variants of Honeycomb meshes [32]. The diameter and bisection width of this topology is 1.06n and 1.41n, respectively. It is also observed that the bisection width of Honeycomb Square Mesh topology is comparatively less than Rhombic and Honeycomb Mesh topology.

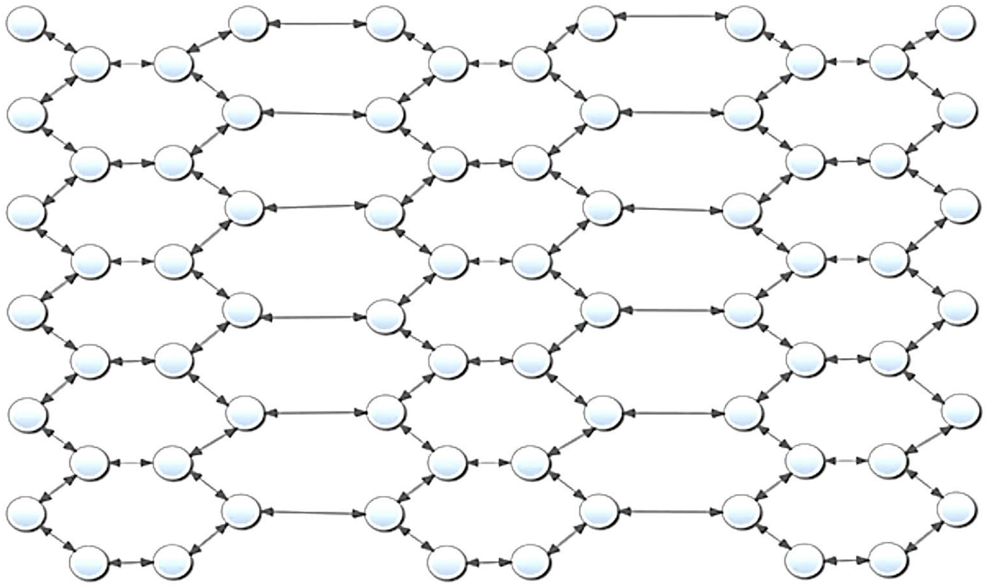

Figure 2.17 Honeycomb Square Mesh Topology

2.2.18 Hexagonal Mesh Topology

This topology is another variant of Mesh topology and resembles SD torus topology. In Hexagonal Mesh toroidal links are missing, which are present in SD torus as shown in Figure 2.18. This topology represents maximum degree of node s i.e. 6 and its diameter is 2(n-1)0.5 which is shorter in comparison to SD torus [33].

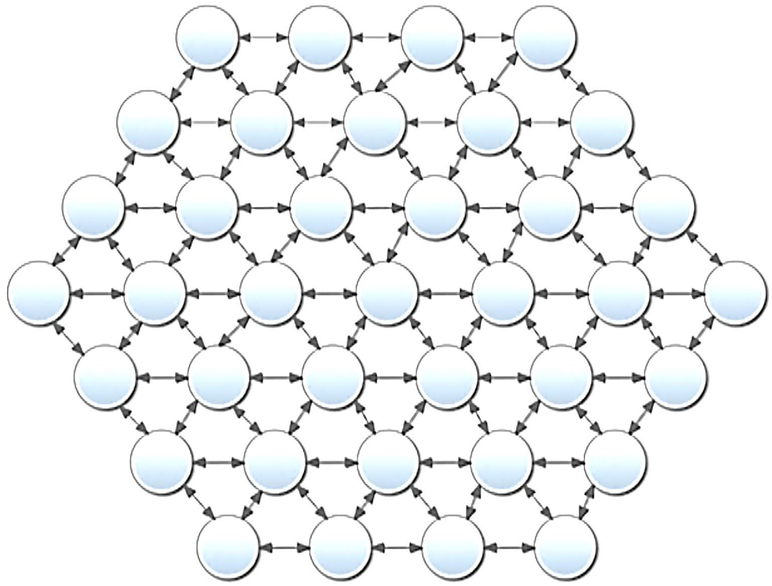

Figure 2.18 Hexagonal Mesh Topology

2.2.19 Diagonal Mesh Topology

In this study, representing Diagonal Mesh topology, authors used diagonal links instead of vertical and horizontal links to connect nodes for transmitting data packets [34, 35]. Herein, degree 4 for each node and bisection width for odd parity topology is 4n is employed. This study revealed better path diversity which is helpful for the congestion situation. The Diagonal Mesh topology is shown in Figure 2.19.

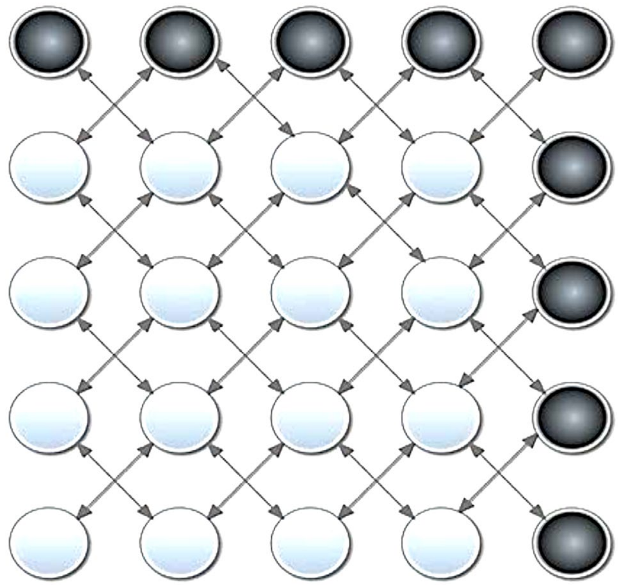

Figure 2.19 Diagonal Mesh Topology

2.2.20 Hybrid NOC Topology

Pertaining to the information provided by the authors, Hybrid NOC topology (Figure 2.20) is the combination of three different topologies i.e. Mesh, Torus and Folded Torus [14, 36]. This advantage leads to better topological properties. The toroidal links helps to achieve reduced diameter, mesh helps to make connection between adjacent nodes while Folded Torus makes extra links to join odd and even nodes together. The degree of above topology is found to be 3 and 4, and the diameter of Hybrid NOC topology is ‘n’ while its bisection width is 3n.

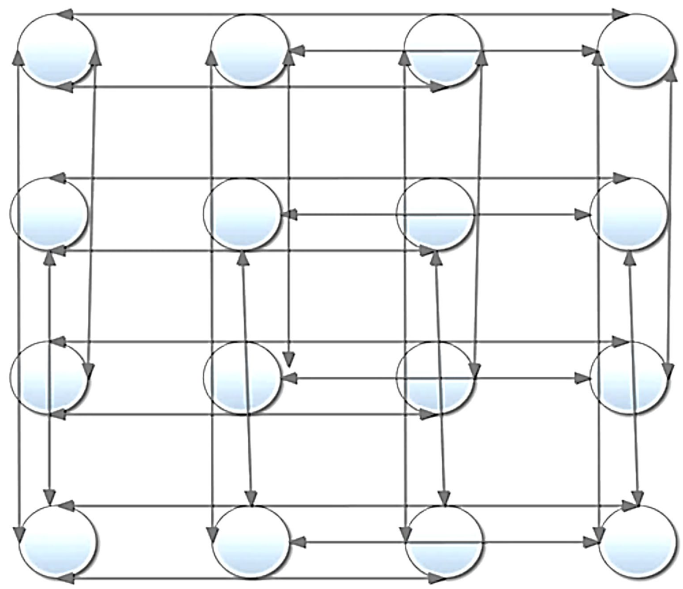

Figure 2.20 Hybrid NOC Topology

2.2.21 SD-Torus Topology

This is another variant of torus topology where minor diagonal nodes relate to extra links [37]. This is symmetrical in nature and degree of each node is uniform i.e. 6. Diameter of SD-Torus is and its bisection width is 3n. This topology takes minimal time to send data over communication links and therefore results in improvement of network performance, as shown in Figure 2.21.

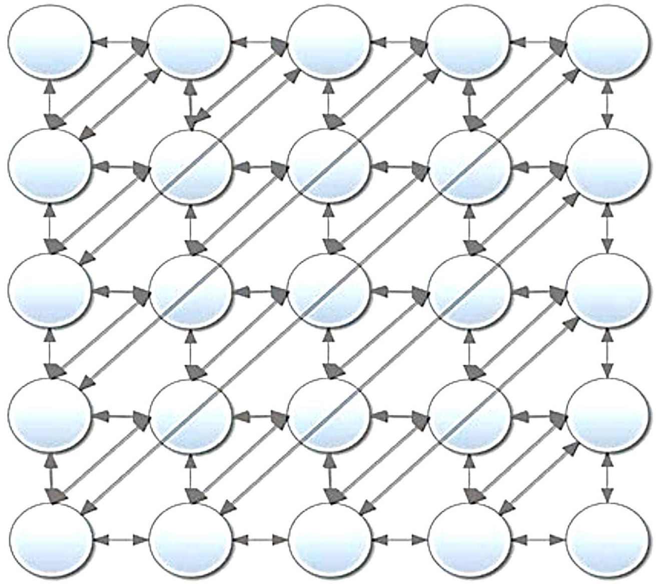

Figure 2.21 SD-Torus Topology

2.2.22 Honeycomb Torus Topology

This topology belongs to the family of Honeycomb Mesh. Honeycomb Torus topology contains three types of toroidal links [32, 38] as shown in Figure 2.22. Each node is connected to three different nodes, with degree of every node as 3. In addition, the diameter of this topology is 0.81n and bisection width is 2.04n. It is node symmetrical in nature and provides better scalability in the network.

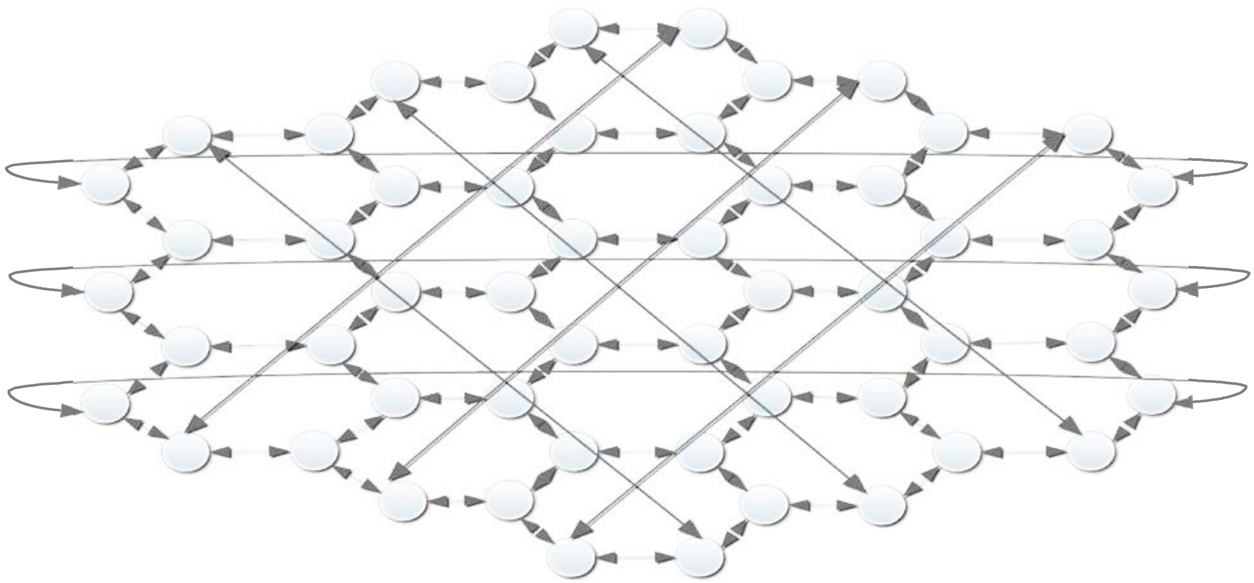

Figure 2.22 Honeycomb Torus Topology

2.2.23 Honeycomb Square Torus Topology

Honeycomb Square Torus topology is the improved version of Honeycomb Square Mesh topology and is much efficient [32]. Degree of both topologies are same which is 3. The bisection width of Honeycomb Square Torus topology is 0.5n and its diameter is 2n, representing high scalability and reliability. This topology is presented in Figure 2.23.

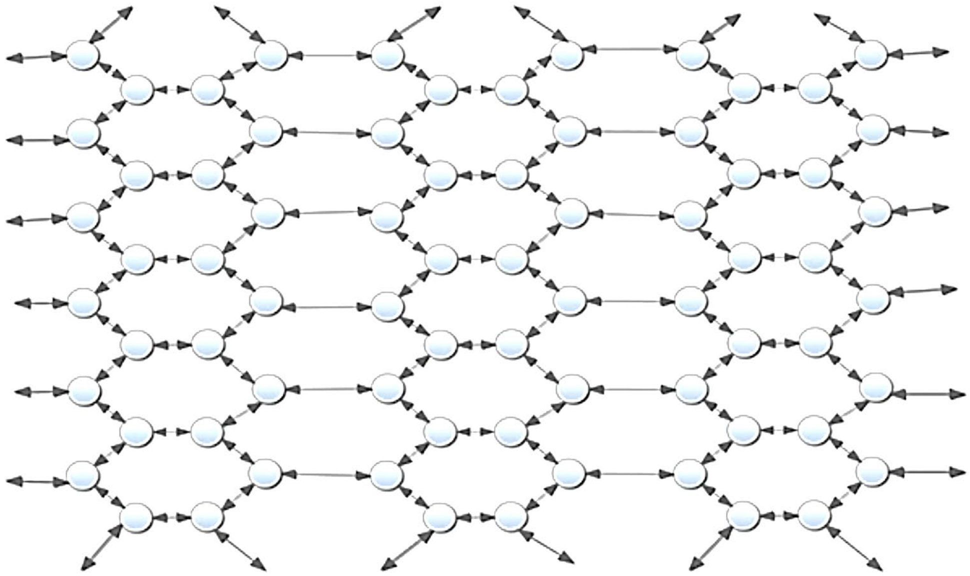

Figure 2.23 Honeycomb Square Torus Topology

2.2.24 Honeycomb Rhombic Torus Topology

Honeycomb Rhombic Torus topology is the improved and efficient version of Honeycomb Rhombic Mesh topology (Figure 2.24) [32]. There is no difference in degree for both topologies i.e. 3. In addition, the bisection width of Honeycomb Square Torus topology is 1.41n while its diameter is 1.06n. High scalability and reliability are the features of this topology.

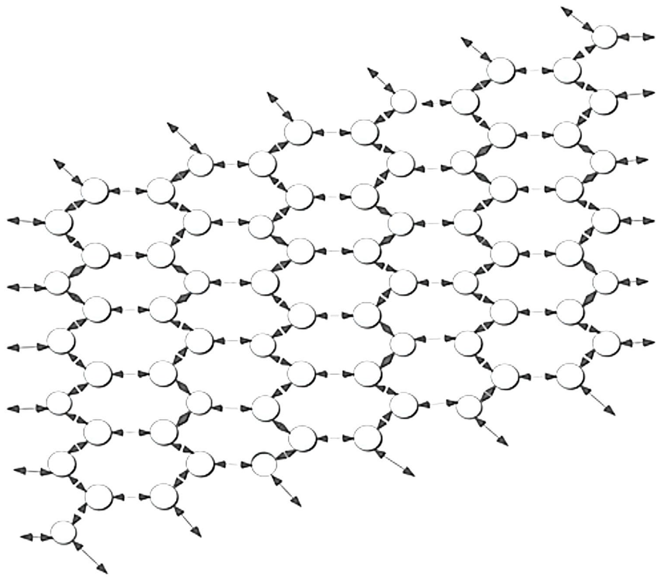

Figure 2.24 Honeycomb Rhombic Torus Topology

2.2.25 Cross bypass Torus Topology

This topology is the advanced version of cross bypass torus topology [39]. The degree of both topologies or variants are similar, with higher bisection width i.e. 3n+2 for odd parity and 3n for even parity. The study revealed that, due to sufficient bandwidth, it can handle much traffic in the network and improves network performance. Additionally, it has smaller diameter which is (3n-2)/4, resulting in lesser time to send packets over the network. This topology is presented in Figure 2.25.

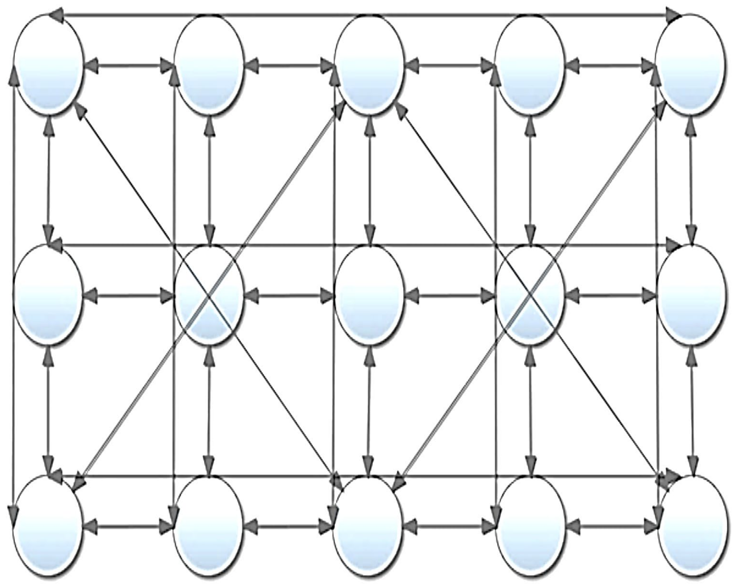

Figure 2.25 Cross bypass Torus Topology

2.2.26 CC-Torus Topology

CC-Torus topology is an extended form of C2 mesh topology [40]. In this topology, center nodes are connected to corner nodes as well as to vertical and horizontal edge nodes, as presented in Figure 2.26. Node degree for even parity CC-Torus topology are 5, 6 and 7 while for odd parity it is 4, 5 and 12. Diameter of even parity is n-1, whereas for odd parity it is n-2. The bisection width for even and odd parity are 2n and 2n+ 4, respectively.

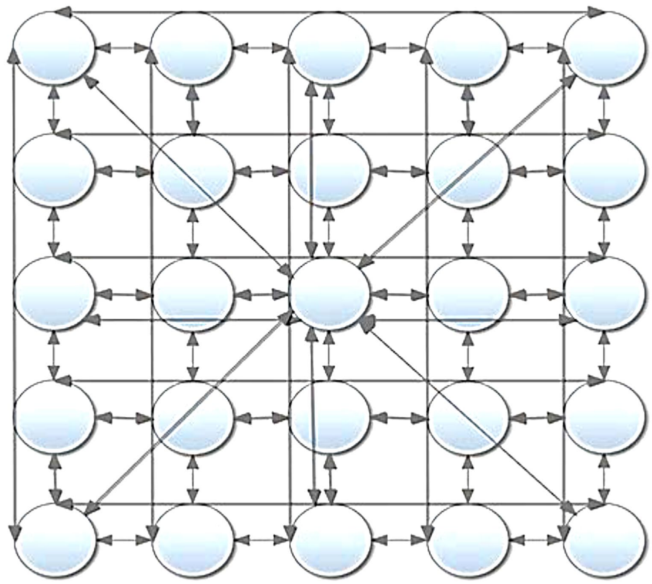

Figure 2.26 CC-Torus Topology

2.2.27 Dtorus Topology

Dtorus topology is the modification of Dmesh where some extra wraparound links are used to connect nodes to reach lesser hop count, depicting a positive impact to gain network performance [41, 42]. Improvement in bisectional width is resulted with this extra links. Degree of nodes in this topology is different which are 5, 6 and 8, respectively, whereas, the diameter of Dtorus topology is n-1 and its bisection width is 4n-2 which is much higher than Dmesh and Torus topology. The layout of this topology is presented in Figure 2.27.

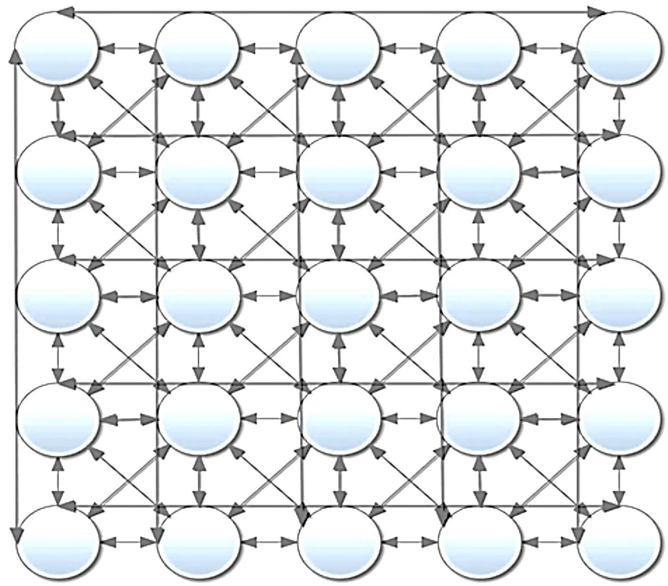

Figure 2.27 Dtorus Topology

2.2.28 xtorus Topology

This is the extended version of Torus topology where extra links are used to connect diagonally placed nodes to convert it into xtorus topology as shown in Figure 2.28 [43]. The main advantage of this topology is to reduce diameter of network which is responsible for the communication delay. Herein, the degree of nodes is 4, 5 and 6 respectively, whereas, the diameter of this topology is reduced to n-2 which is much smaller in comparison to torus topology with the bisection width of 2n+2.

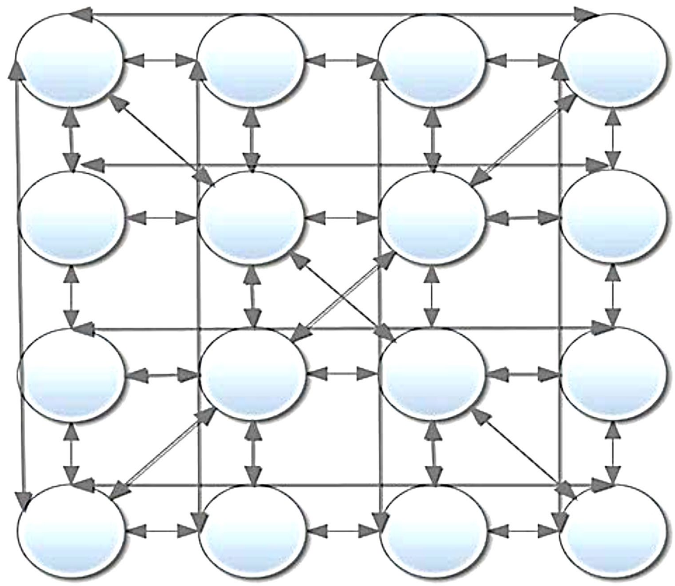

Figure 2.28 xtorus Topology

2.2.29 xxtorus Topology

This is improved version of xtorus topology where 2 extra links are used to connect diagonally corner nodes [44], as presented in Figure 2.29. This topology enhances path diversity as well as reduces communication delay by covering nodes in lesser hops. The degree of nodes is 4 and 6, with no change in diameter i.e. n-2. However, there is an impact on bisectional width which is increased to 2n+4.

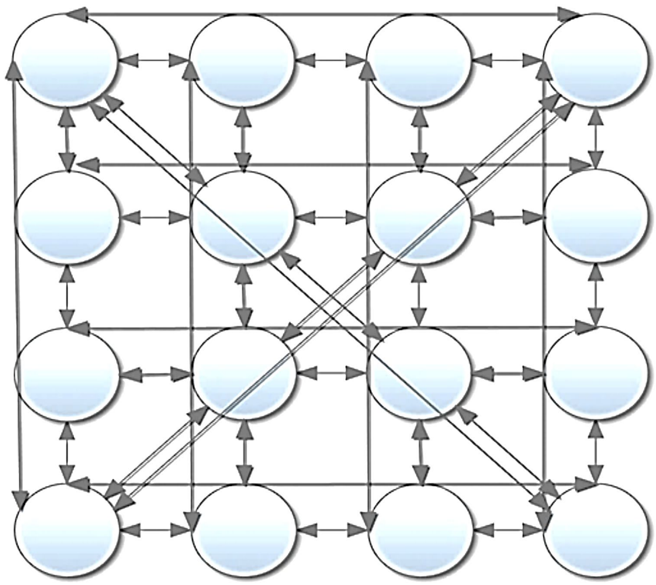

Figure 2.29 xxtorus Topology

2.2.30 X-Torus Topology

X-Torus topology is an improved version of torus topology [9, 45]. This improved version contains better bisection width, lower diameter in comparison to torus, mesh and other topologies. Degree of odd parity X-torus is found to be 5 and 6 while diameter is $\lfloor n/2 \rfloor+1$with better bisection width of n^2-n. This topology takes minimal hop counts to send data packets from source node to destination nodes. This topology is presented in Figure 2.30.

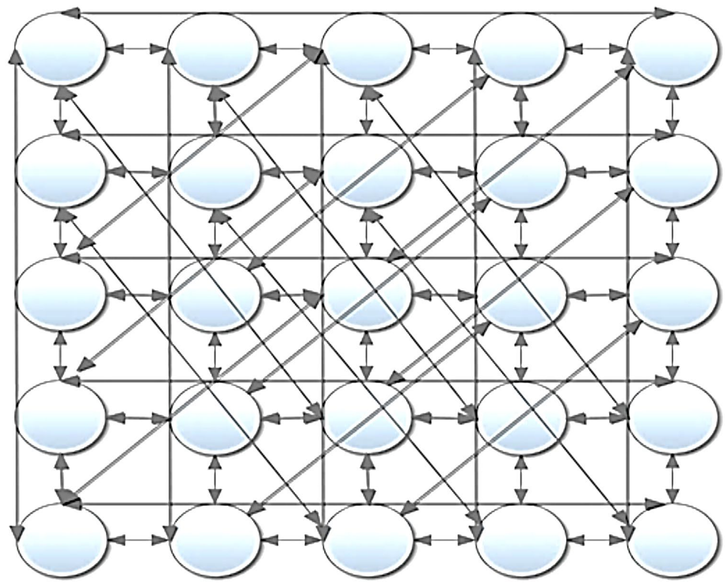

Figure 2.30 X-Torus Topology

In this chapter we have discussed about topological properties of existing networks which helped us to modify existing topology and designing efficient network to get optimal network performance in parallel computing systems.

CHAPTER 3
A MODIFIED X-TORUS TOPOLOGY FOR INTERCONNECTION NETWORKS

3.1 Introduction and Motivation

In today's era, the performance of the interconnection network topologies is an essential matter in the design of multi-processor systems [46]. It has been observed that the performance of interconnection networks is influenced by three parameters namely; topology, routing algorithm and flow control mechanism [1, 47]. Among these three parameters, network topology plays a decisive role, as it affects the key features like bisection bandwidth, node degree, diameter and average distance. The literature survey defines numerous topologies. This has been noted that it is affected by various factors, therefore researchers aim to design topologies which provides smaller diameter, maximum bisectional bandwidth, shorter average distance and higher stability [48]. In real world of technology, the processing power plays a significant role and is in absolute demand of increment. Therefore, to overcome speed related problems, high speed processors are developed. However, because of mismatch in transfer speed, the key issue involves is the data exchange between the processor and memory that results in delay and affects the performance of the system. To resolve this issue, the concept of interconnection networks come into existence which provides significant assistance to minimize the delay between the components of the computer systems. However, in previous time data between nodes was shared by bus interconnection networks [49]. It was considered simple to design and implement but performance was the main issue of concern. In this topology, at a time only one node can send the data in the channel and other must sense it before transmitting their data packet to another node in the network. There was the limitation of the nodes on the communication channel of bus topology [50]. It was security prone and reduces the performance with the increase of nodes. Also, in case the link gets damaged at any point, it resulted the failure of overall system. Furthermore, mesh topology was designed which have provided the facility to other nodes to send data packet at the same time, with better scalability in comparison to shared bus topology [51]. A better performance in comparison to existing topologies was observed.

In mesh topology, moving data between corner nodes still took much hop count [52]. To overcome this matter, torus topology was proposed [53]. In this topology corner nodes are connected by wrap around links and turn mesh in node symmetrical structure where degree of each node becomes 4, diameter is reduced to half and bisection width become double. It has better path diversity and scalability. In addition, there are many topologies that have been proposed to improve the network performance either providing less delay, or diameter etc. To gain better performance in latency and throughput, a new topology is designed which is known as X-Torus topology [9]. The X-Torus topology has been described and presented in Figure 2.30. In this topology, authors attributed that the degree of X-Torus odd parity topology is 6 and revealing that each node will use a 6-degree router. However, in most of the cases, it has not been utilized. The complete mathematical description of X-Torus topology is presented and described within the study [9, 45].

In this chapter, we have proposed a topology that can have a uniform degree by adding extra communication links without affecting the degree of the router. Detailed information of this topology is given section 3.2 and its experimental setup and test bed are described in section 3.3. Section 3.4 includes detailed discussions of the results.

3.2 Modified X-Torus Topology

This section describes the proposed topology for parallel computing system [54]. The proposed topology is known as Modified X-Torus. It is the extended version of X-torus topology which is represented by a graph. In graph, nodes represent the processing unit and edges are the bisectional communication links of the interconnection networks as shown in Figure 3.1. The proposed topology inherits the fundamental properties of torus topology like symmetry, scalability as well as fault tolerance nature [55, 56]. The mathematical representation of proposed topology of odd parity is given below [54].

$$C_x = \left\{ \left\langle (u_a, u_b), (v_a, v_b) \right\rangle \middle| \left(\left(u_a = v_a \cap u_b = [v_b \pm 1]_{k_x} \right) \cup \left(u_a = \left[v_a + \left\lceil \frac{k_x}{2} \right\rceil \right]_{k_x} \cap u_b = v_b + \left\lfloor \frac{k_x}{2} \right\rfloor_{k_x} \right) \right) \cap \left((u_a, u_b), (v_a, v_b) \right) \in N_x \right\} \tag{3.1}$$

$$C_y = \left\{ \left\langle (u_a, u_b), (v_a, v_b) \right\rangle \middle| \left(\left(u_b = v_b \cap u_a = [v_a \pm 1]_{k_y} \right) \cup \left(u_b = \left[v_b + \left\lceil \frac{k_y}{2} \right\rceil \right]_{k_y} \cap u_a = v_a + \left\lfloor \frac{k_y}{2} \right\rfloor_{k_y} \right) \right) \cap \left((u_a, u_b), (v_a, v_b) \right) \in N_y \right\} \tag{3.2}$$

$$C = C_x \cup C_y \tag{3.3}$$

In the same way, communication links for even parity topology is described by

$$C_x = \left\{ \left\langle (u_a, u_b), (v_a, v_b) \right\rangle \middle| \left(\left(u_a = v_a \cap u_b = [v_b \pm 1]_{k_x} \right) \cup \left(u_a = \left[v_a + \left\lfloor \frac{k_x}{2} \right\rfloor \right]_{k_x} \cap u_b = v_b + \left\lfloor \frac{k_x}{2} \right\rfloor_{k_x} \right) \right) \cap \left((u_a, u_b), (v_a, v_b) \right) \in N_x \right\} \tag{3.4}$$

$$C_y = \left\{ \left\langle (u_a, u_b), (v_a, v_b) \right\rangle \middle| \left(\left(u_b = v_b \cap u_a = [v_a \pm 1]_{k_y} \right) \cup \left(u_b = \left[v_b + \left\lfloor \frac{k_y}{2} \right\rfloor \right]_{k_y} \cap u_a = v_a + \left\lfloor \frac{k_y}{2} \right\rfloor_{k_y} \right) \right) \cap \left((u_a, u_b), (v_a, v_b) \right) \in N_y \right\} \tag{3.5}$$

$$C = C_x \cup C_y \tag{3.6}$$

Here, (v_a, v_b) are the coordinate of the source node and (u_a, u_b) are the coordinates of the adjacent nodes in the proposed topology. (k_x, k_y) are the number of nodes in x dimension and y dimension.

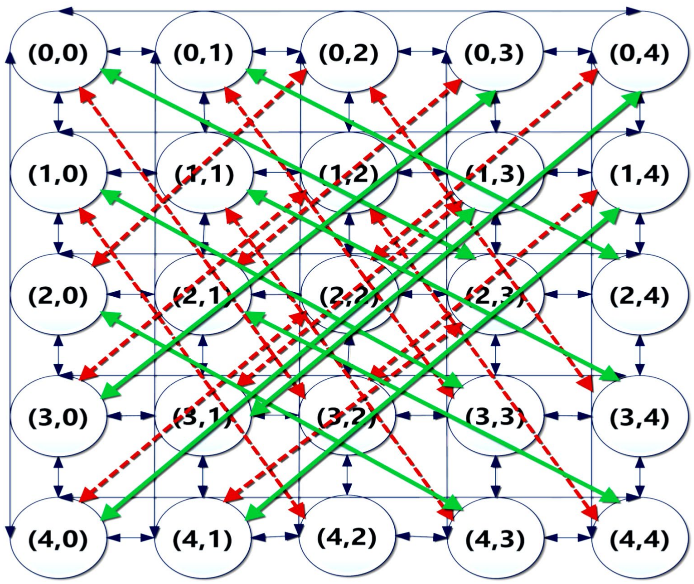

Figure 3.1 Modified X-Torus Topology

Topological properties of modified X-Torus are compared with Mesh, Torus and X-Torus in terms of number of nodes, diameter, bisection width, number of links, degree of various nodes, and path diversity [57]. This comparison is described in Table 3.1.

Table 3.1 Topological Properties

Characteristics	Mesh	Torus	X-Torus	MX-Torus
Number of Nodes	n^2	n^2	n^2	n^2
Diameter	2n-2	n-1	$\left\lfloor \frac{n}{2} \right\rfloor + 1$	$\left\lfloor \frac{n}{2} \right\rfloor + 1$
Bisection Width	n	2n	n^2 -n	n^2 +n
Number of links	$2n^2$-2n	$2n^2$	$2n^2+2\left\lfloor \frac{n}{2} \right\rfloor^2+3\left\lfloor \frac{n}{2} \right\rfloor + 1$	$2n^2+2\left\lfloor \frac{n}{2} \right\rfloor^2+3\left\lfloor \frac{n}{2} \right\rfloor + 2n + 1$
Degree of Nodes	2, 3, 4	4	5, 6	6
Path Diversity	yes	yes	yes	yes

3.3 Testbed for Testing Modified X-Torus Topology

In this section author has analyzed the performance parameters of proposed topology [54]. These parameters are average throughput, average end to end delay and average hop count. For this purpose, author employed windows 10 operating system of 32-bit, equipped intel® Core™ i3 CPU M330@2.13 GHZ with 4.00 GB and 2.99GB usable. Another key component which is used is OMNET++ simulator, a component-based C++ simulation library and framework which is both extensible and modular and is primarily used for building network simulator based on the Eclipse

IDE [58]. All the key parameters for simulation purpose are presented in Table 3.2. This table describes the dimension of topology which is order of 5. Packet size is 1024 bytes, and data rate is 1 Gbps. Warm up time and simulation time are 0.5ms and 0.5s respectively. Author considered 5 traffic patterns including; uniform, bit complement, neighbor, tornado and hotspot. The proposed topology is compared with mesh, torus and x-torus topologies. Simulation results are presented in both, table as well as graph. Table 3.3 to Table 3.7 presents Average End to End Delay under various traffic such as uniform, bit complement, neighbor, tornado and hotspot, respectively. Similarly, Table 3.8 to Table 3.12 presents Average Throughput and Table 3.13 to Table 3.17 shows Average Hop Count for the same traffic patterns. In the same way Figure 3.2 to Figure 3.16 represents the graph of Average End to End Delay, Average Throughput and Average Hop Count for the same traffic patterns which are mentioned. Results pertaining to the evaluation are discussed in section 3.4.

Table 3.2 Simulation Parameters

S.no.	Parameter Name	Value
1	Rows	5
2	Coloums	5
3	Packet size	1024 bytes
4	Data rate	1Gbps
5	Simulation time	0.5 s
6	Warm up time	0.5ms
7	Simulator	OMNeT++
8	Traffic Type	Uniform, Bit Complement, Neighbor, Tornado, Hotspot
9	Link Delay	0.1 ms
10	Routing Algorithm	Table based Shortest Path (Static)

Table 3.3 Average End to End Delay under Uniform Traffic

Interpacket Arrival Delay(μs)	Mesh	Torus	X-Torus	MX-Torus
17.07	0.00239	0.00026	0.00021	0.00020
8.53	0.04343	0.01099	0.00136	0.00028
5.69	0.07309	0.02602	0.01646	0.01514
4.27	0.08986	0.05327	0.02579	0.02291
3.41	0.09687	0.08094	0.03678	0.03000
2.84	0.10134	0.09833	0.04901	0.03614
2.44	0.10392	0.10886	0.05994	0.04496
2.13	0.10559	0.11489	0.06929	0.05443
1.90	0.10740	0.11835	0.07682	0.06144
1.71	0.10951	0.12012	0.08408	0.06676

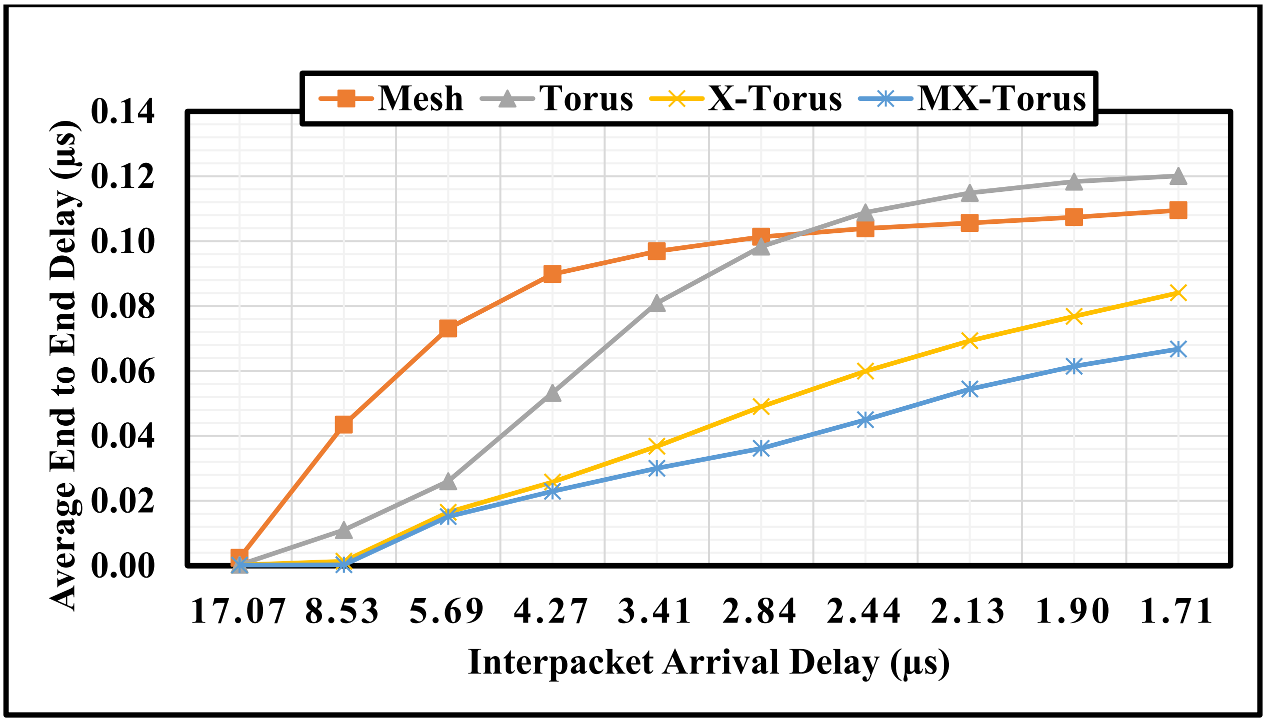

Figure 3.2 Average End to End Delay under Uniform Traffic

Figure 3.2 shows the average end to end delay in the Mesh, Torus, X-Torus and MX-Torus based on data points (Interpacket Arrival Delay) provided in Table 3.3. The end to end delay of Torus and Mesh showed a steady but significant rise over the interpacket arrival delay, while in X-torus and MX-Torus the end to end delay rose gradually. At 8.53 the end to end delay of X-Torus and MX-Torus were 0.00136 μs and 0.00028 μs respectively. Torus's delay increased sharply throughout the time, exceeding Mesh topology's delay and reaching almost 0.12012 μs at the end of the time. In the meantime, the average end to end delay of X-torus and MX-Torus grew to 0.01646 µs and 0.01514 µs at 5.69 respectively and then increased gradually until 1.71.

Table 3.4 Average End to End Delay under Bit Complement Traffic

Interpacket Arrival Delay(µs)	Mesh	Torus	X-Torus	MX-Torus
17.07	0.07701	0.00026	0.00021	0.00020
8.53	0.15696	0.07684	0.03851	0.03850
5.69	0.18478	0.12905	0.10127	0.10126
4.27	0.20007	0.15683	0.13598	0.13597
3.41	0.20916	0.17350	0.15681	0.15680
2.84	0.21510	0.18462	0.17069	0.17069
2.44	0.21933	0.19256	0.18061	0.18061
2.13	0.22249	0.19853	0.18806	0.18805
1.90	0.22499	0.20317	0.19384	0.19384
1.71	0.22706	0.20688	0.19847	0.19847

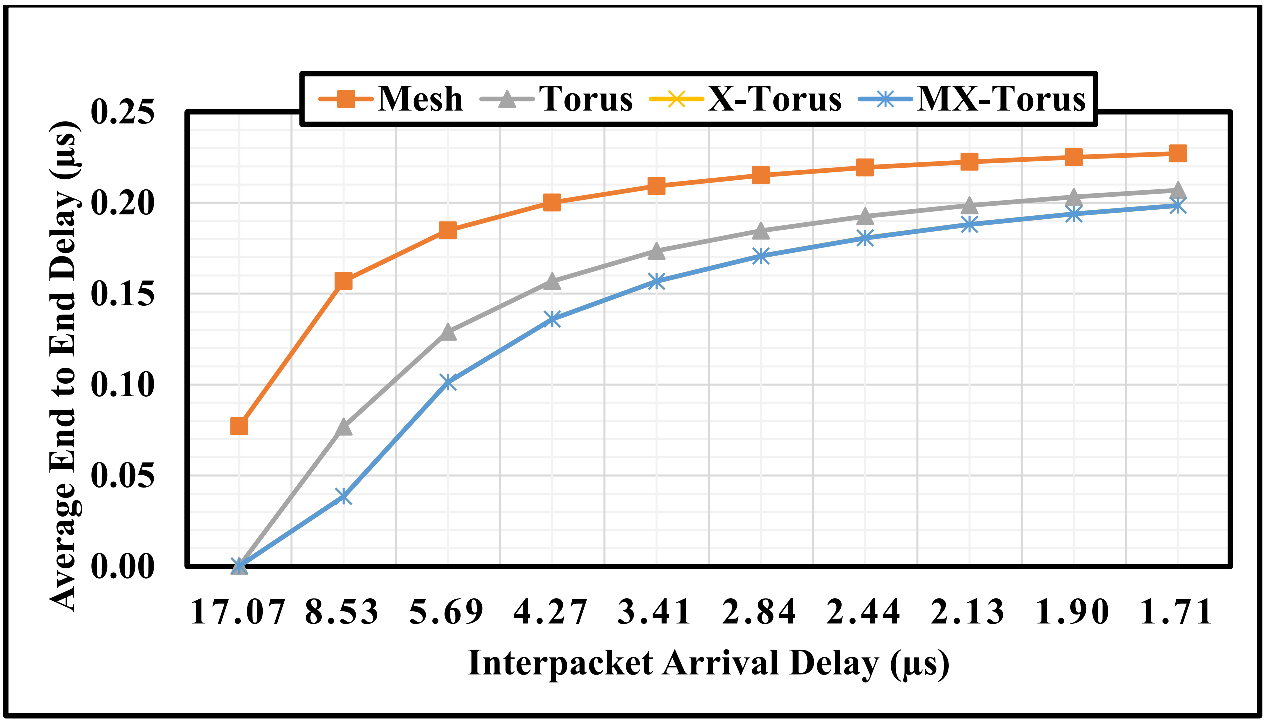

Figure 3.3 Average End to End Delay under Bit Complement Traffic

Figure 3.3 illustrates the average end to end delay in the Mesh, Torus, X-Torus and MX-Torus based on data points (Interpacket Arrival Delay) provided in Table 3.4. In starting the end to end delay of Mesh, Torus, X-Torus and MX-Torus rose significantly and then increased gently over the interpacket arrival delay. At 17.07 the end to end delay of Torus, X-Torus and MX-Torus were nearly 0.00026 μs, 0.00021 μs and 0.00020 μs respectively. Delay of X-Torus and MX-Torus are almost same. Mesh's delay increased sharply at 2.84, exceeding Torus topology's delay and reaching almost 0.22706 μs at the end of the time. In the meantime, the average end to end delay of X-Torus and MX-Torus gradually grew to 0.19847 μs at 1.71.

Table 3.5 Average End to End Delay under Neighbor Traffic

Interpacket Arrival Delay(µs)	Mesh	Torus	X-Torus	MX-Torus
17.07	0.00021	0.00013	0.00013	0.00013
8.53	0.00977	0.00971	0.00971	0.00971
5.69	0.08349	0.08344	0.08344	0.08344
4.27	0.12515	0.12509	0.12509	0.12509
3.41	0.15014	0.15009	0.15009	0.15009
2.84	0.16681	0.16675	0.16675	0.16675
2.44	0.17871	0.17865	0.17865	0.17865
2.13	0.18764	0.18758	0.18758	0.18758
1.90	0.19458	0.19453	0.19453	0.19453
1.71	0.20014	0.20008	0.20008	0.20008

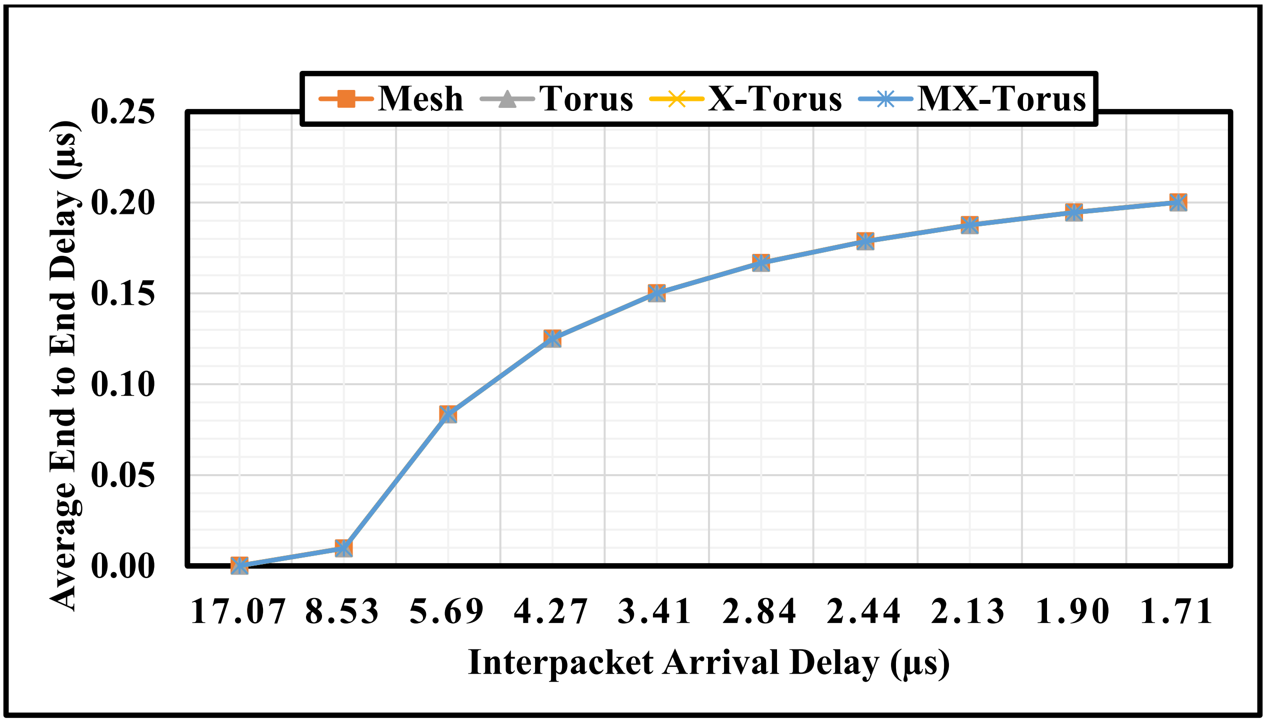

Figure 3.4 Average End to End Delay under Neighbor Traffic

Figure 3.4 gives information about the average end to end delay in the Mesh, Torus, X-Torus and MX-Torus based on data points (Interpacket Arrival Delay) provided in Table 3.5. In starting the end to end delay of Mesh, Torus, X-Torus and MX-Torus experienced a gradual rise and then increased dramatically over the interpacket arrival delay. At 17.07 the end to end delay of Torus, X-Torus and MX-Torus were 0.00013 µs. Delay of Torus, X-Torus and MX-Torus are almost same. Mesh's delay increased gently at 8.53, exceeding Torus, X-Torus, and MX-Torus topology's delay and reaching almost 0.20014 μs at the end of the time. In the meantime, the average end to end delay of Torus, X-Torus and MX-Torus gradually grew to 0.20008 μs at 1.71.

Table 3.6 Average End to End Delay under Tornado Traffic

Interpacket Arrival Delay(μs)	Mesh	Torus	X-Torus	MX-Torus
17.07	0.19439	0.22507	0.05845	0.02175
8.53	0.21023	0.23804	0.12732	0.09958
5.69	0.21678	0.24140	0.16533	0.14545
4.27	0.22135	0.24304	0.18432	0.16913
3.41	0.22482	0.24414	0.19553	0.18327
2.84	0.22755	0.24492	0.20304	0.19282
2.44	0.22975	0.24556	0.20848	0.19977
2.13	0.23157	0.24613	0.21266	0.20510
1.90	0.23310	0.24654	0.21599	0.20934
1.71	0.23440	0.24701	0.21872	0.21281

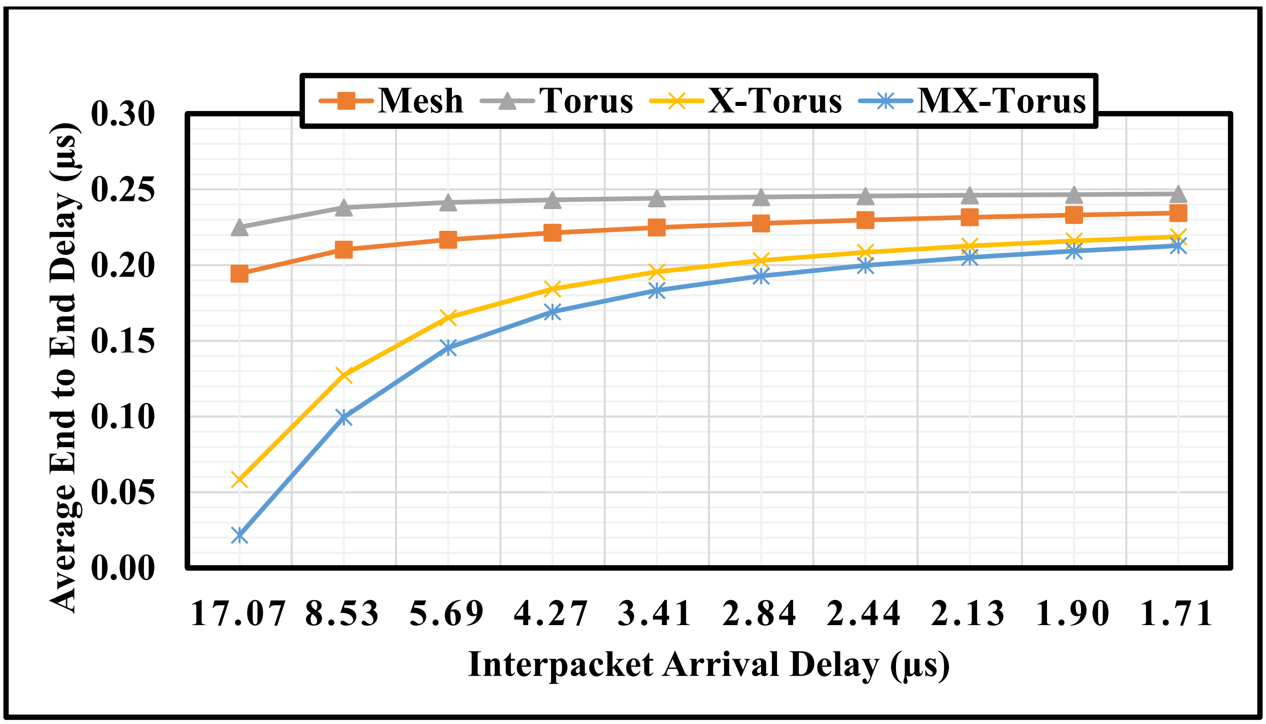

Figure 3.5 Average End to End Delay under Tornado Traffic

Figure 3.5 describes the average end to end delay in the Mesh, Torus, X-Torus and MX-Torus based on data points (Interpacket Arrival Delay) provided in Table 3.6. In starting the end to end delay of Mesh, Torus, X-Torus and MX-Torus showed a gradual increase and then remained almost steady over the interpacket arrival delay. At 5.69 the end to end delay of Mesh and Torus were 0.21678 µs and 0.24140 µs and then rose gently to 0.23440 µs and 0.24701 µs at the end of time respectively. In the meantime, the average end to end delay of X-Torus and MX-Torus increased considerably to 0.20304 µs and 0.19282 µs at 2.84 respectively and then rose gently to 0.21872 µs and 0.21281 µs at 1.71 respectively.

Table 3.7 Average End to End Delay under Hotspot Traffic

Interpacket Arrival Delay(µs)	Mesh	Torus	X-Torus	MX-Torus
17.07	0.00814	0.00227	0.00021	0.00099
8.53	0.04593	0.01680	0.00442	0.00543
5.69	0.07254	0.02580	0.01427	0.01171
4.27	0.08862	0.05244	0.02351	0.01985
3.41	0.09557	0.07955	0.03556	0.02815
2.84	0.10089	0.09610	0.04894	0.03515
2.44	0.10441	0.10782	0.06059	0.04433
2.13	0.10645	0.11513	0.07038	0.05414
1.90	0.10825	0.11958	0.07814	0.06172
1.71	0.11052	0.12225	0.08517	0.06775

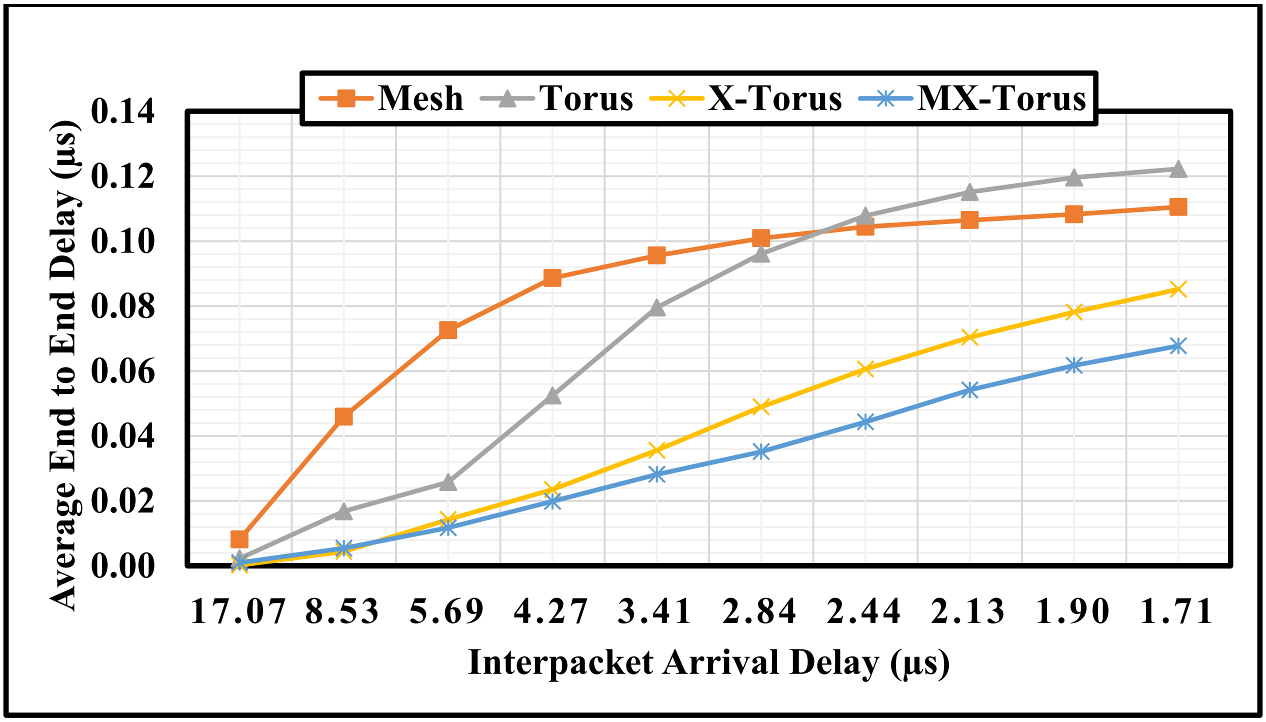

Figure 3.6 Average End to End Delay under Hotspot Traffic

Figure 3.6 indicates the average end to end delay in the Mesh, Torus, X-Torus and MX-Torus based on data points (Interpacket Arrival Delay) provided in Table 3.7. The end to end delay of Torus and Mesh increased significantly over the interpacket arrival delay, while in X-torus and MX-Torus the end to end delay rose gradually. At 8.53 the end to end delay of X-Torus and MX-Torus were 0.00442 µs and 0.00543 µs respectively. Torus's delay increased sharply throughout the time, exceeding Mesh topology's delay and reaching almost 0.12225 µs at the end of the time. In the meantime, the average end to end delay of X-torus and MX-Torus grew to 0.02351 µs and 0.01985 µs at 4.27 respectively and then experienced a significant rise until 1.71.

Table 3.8 Average Throughput under Uniform Traffic

Interpacket Arrival Delay(us)	Mesh	Torus	X-Torus	MX-Torus
17.07	58061.52	58562.48	58567.83	58569.35
8.53	90463.99	111655.91	116571.67	117109.74
5.69	103380.32	149966.08	161123.82	162392.02
4.27	109051.59	169907.41	199149.69	202271.52
3.41	114654.80	173585.03	229235.51	237510.79
2.84	119390.38	174261.33	251530.40	269531.90
2.44	124002.47	174215.65	268674.31	294646.06
2.13	128169.50	174991.33	282345.84	313587.87
1.90	131656.36	176402.24	293940.15	331074.50
1.71	134041.22	178252.85	302420.27	347746.68

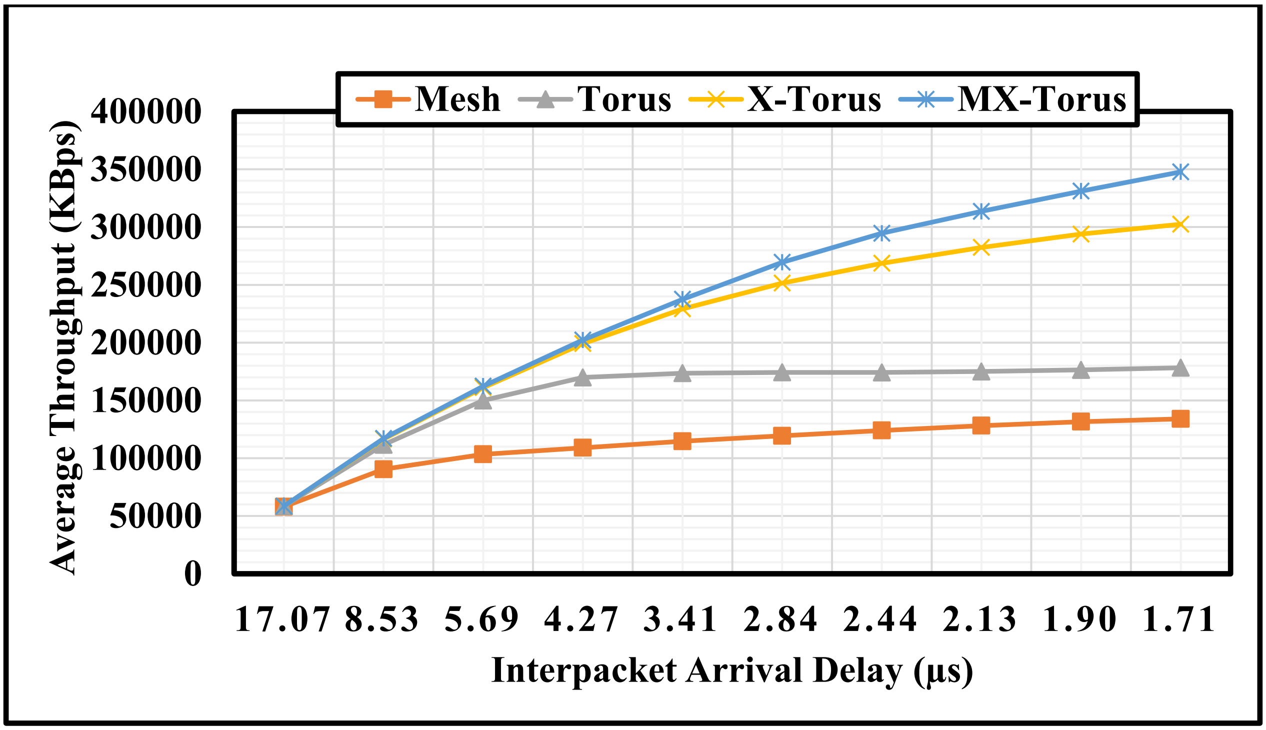

Figure 3.7 Average Throughput under Uniform Traffic

Figure 3.7 shows the average throughput in the Mesh, Torus, X-Torus and MX-Torus based on data points (Interpacket Arrival Delay) provided in Table 3.8. The average throughput of X-Torus and MX-Torus showed a steady but significant rise over the interpacket arrival delay, while in Mesh and Torus the average throughput rose gradually. At 8.53 the average throughput of Torus, X-Torus and MX-Torus were 111655.91 KBps, 116571.67 KBps and 117109.74 KBps respectively. MX-Torus' throughput increased sharply throughout the time, exceeding X-Torus, Torus and Mesh topology's throughput and reaching almost 347746.68 KBps at the end of the period. In the meantime, the average throughput of Torus and Mesh grew to 169907.41 KBps and 109051.59 KBps at 4.27 respectively and then remained steady until 1.71.

Table 3.9 Average Throughput under Bit Complement Traffic

Interpacket Arrival Delay(us)	Mesh	Torus	X-Torus	MX-Torus
17.07	40592.63	58562.07	58568.31	58569.43
8.53	43720.08	81217.67	99178.61	99180.69
5.69	46063.72	85123.30	104646.45	104648.69
4.27	47182.17	87466.04	106990.20	106992.28
3.41	48462.01	89809.23	109333.94	109336.02
2.84	49977.18	92153.01	111677.81	111679.89
2.44	51658.28	94496.26	114021.34	114023.42
2.13	53459.83	96839.91	116365.16	116367.24
1.90	55352.38	99183.41	118709.00	118711.08
1.71	57314.37	101526.67	121052.42	121054.50

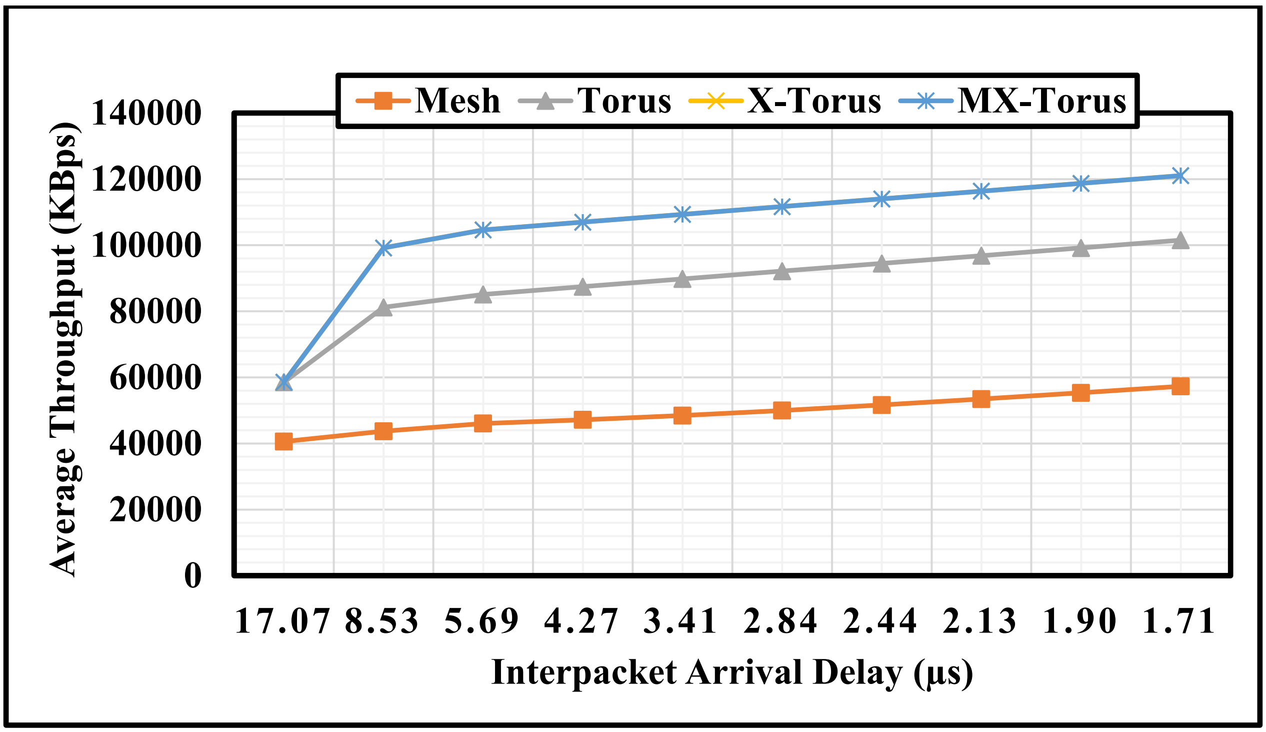

Figure 3.8 Average Throughput under Bit Complement Traffic

Figure 3.8 illustrates the average throughput in the Mesh, Torus, X-Torus and MX-Torus based on data points (Interpacket Arrival Delay) provided in Table. In starting, the average throughput of Torus, X-Torus and MX-Torus experienced a significant trend over the interpacket arrival delay and then increased, while in Mesh the average throughput rose gradually. At 8.53 the average throughput of X-Torus and MX-Torus were nearly 99178.61 KBps and 99180.69 KBps respectively. MX-Torus' throughput increased sharply throughout the time, exceeding X-Torus, Torus and Mesh topology's throughput and reaching almost 121054.50 KBps at the end of the period. In the meantime, the average throughput of Torus and Mesh grew to 81217.67 KBps and 43720.08 KBps at 8.53 respectively and then rose gradually until 1.71.

Table 3.10 Average Throughput under Neighbor Traffic

Interpacket Arrival Delay(us)	Mesh	Torus	X-Torus	MX-Torus
17.07	58568.31	58577.43	58577.43	58577.43
8.53	112654.63	112666.64	112666.64	112666.64
5.69	117145.45	117158.00	117158.00	117158.00
4.27	117145.31	117157.92	117157.92	117157.92
3.41	117145.43	117157.91	117157.91	117157.91
2.84	117145.48	117158.03	117158.03	117158.03
2.44	117145.25	117158.05	117158.05	117158.05
2.13	117145.32	117158.12	117158.12	117158.12
1.90	117145.40	117158.28	117158.28	117158.28
1.71	117145.39	117158.19	117158.19	117158.19

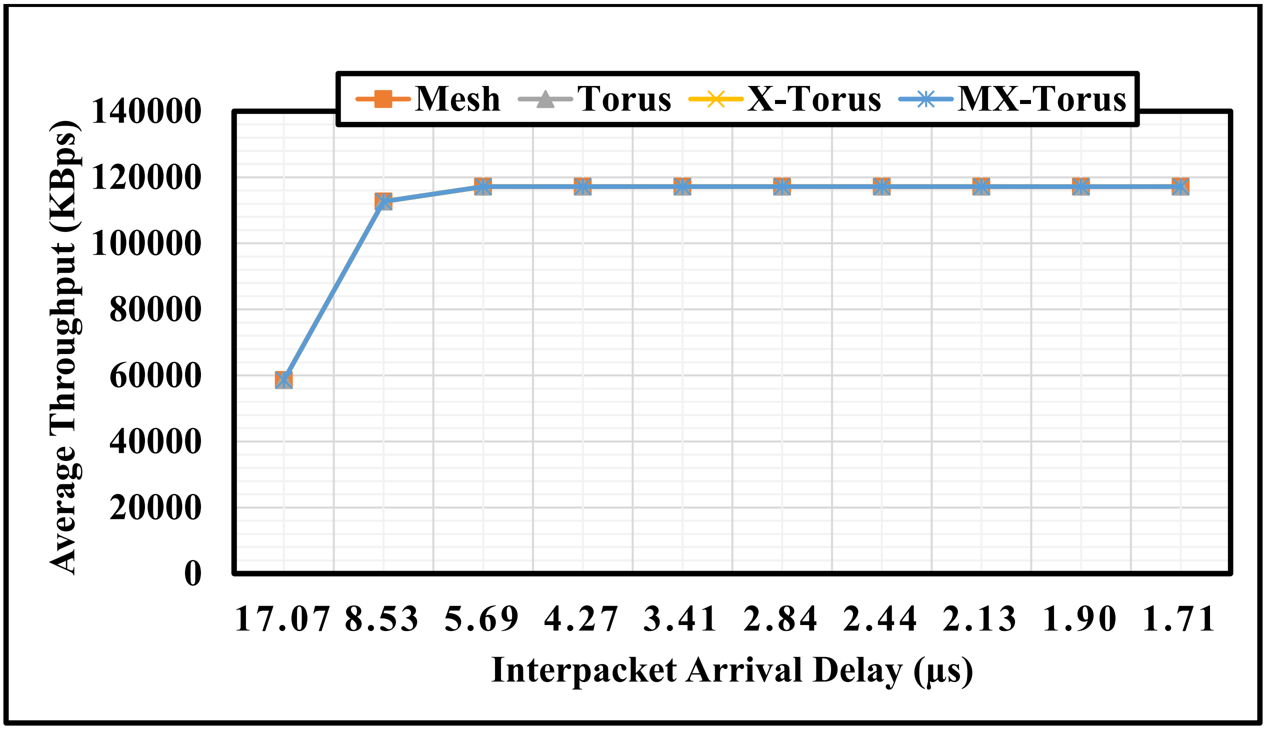

Figure 3.9 Average Throughput under Neighbor Traffic

Figure 3.9 gives information about the average throughput in the Mesh, Torus, X-Torus and MX-Torus based on data points (Interpacket Arrival Delay) provided in Table 3.10. In starting, the average throughput of Mesh, Torus, X-Torus and MX-Torus showed a significant rise over the interpacket arrival delay and then remained same. At 17.07 the average throughput of Torus, X-Torus and MX-Torus were 58577.43 KBps and then reaching 117158.19 KBps at 1.71. In the meantime, the average throughput of Mesh grew to 112654.63 KBps at 8.53 and then rose gradually to 117145.45 KBps at 5.69. However, it remained the same at end of the time.

Table 3.11 Average Throughput under Tornado Traffic

Interpacket Arrival Delay(us)	Mesh	Torus	X-Torus	MX-Torus
17.07	58638.35	29333.92	202842.77	250965.96
8.53	57048.39	25691.91	209932.06	280269.81
5.69	55897.85	23245.90	204420.76	277194.86
4.27	55028.57	21235.91	199964.78	273043.28
3.41	54348.57	19553.92	196649.58	269881.27
2.84	53802.63	18121.94	194060.99	267375.96
2.44	53354.28	16891.96	191974.76	265330.19
2.13	52979.91	15823.97	190252.08	263626.35
1.90	52661.98	14880.00	188805.14	262185.52
1.71	52389.58	17054.36	187565.91	260949.88

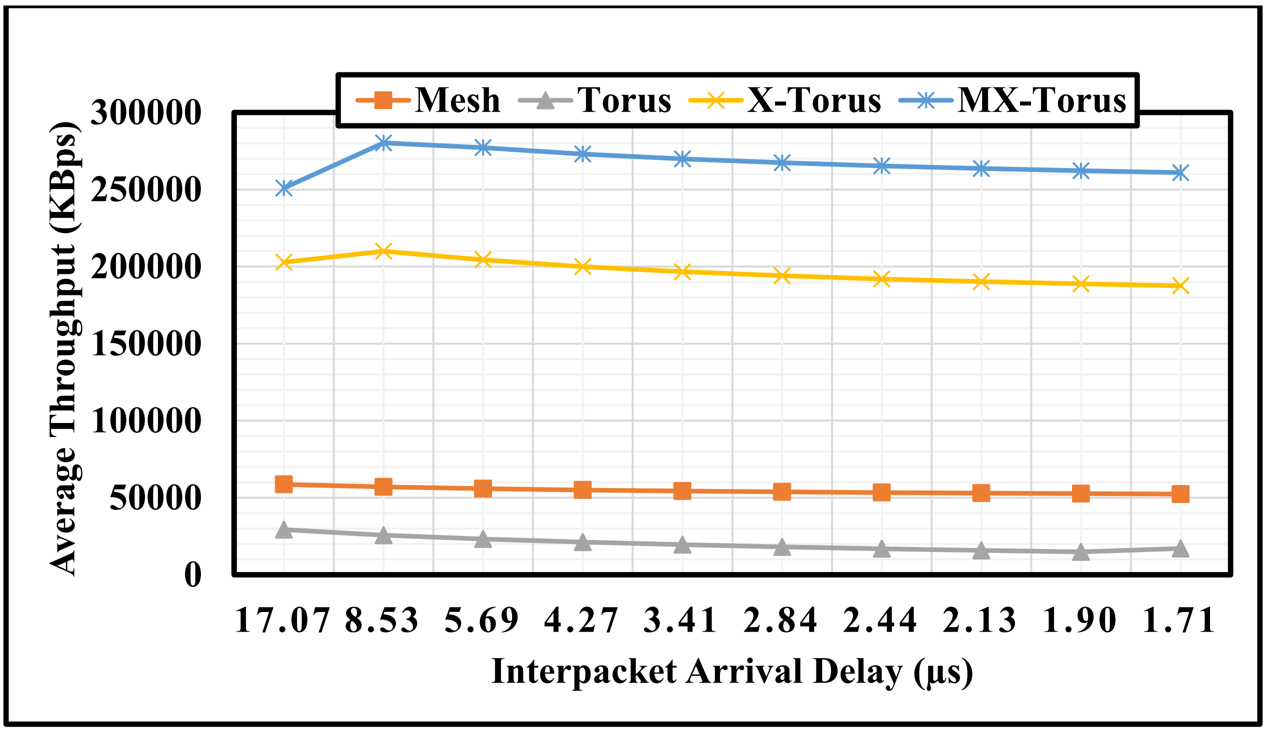

Figure 3.10 Average Throughput under Tornado Traffic

Figure 3.10 indicates the average throughput in the Mesh, Torus, X-Torus and MX-Torus based on data points (Interpacket Arrival Delay) mentioned in Table 3.11. In the beginning, the average throughput of X-Torus and MX-Torus showed a significant rise and then experienced a gradual decrease over the interpacket arrival delay, while in Mesh and Torus the average throughput fell gradually. At 8.53 the average throughput of X-Torus and MX-Torus were 209932.06 KBps, and

280269.81 KBps respectively. MX-Torus' throughput decreased gently throughout the time, exceeding X-Torus, Torus and Mesh topology's throughput and reaching almost 260949.88 KBps at the end of the period. In the meantime, the average throughput of Mesh, Torus and X-Torus declined gradually to 55897.85 KBps, 23245.90 KBps and 204420.76 KBps at 5.69 respectively and then remained almost same until 1.71.

Table 3.12 Average Throughput under Hotspot Traffic

Interpacket Arrival Delay(us)	Mesh	Torus	X-Torus	MX-Torus
17.07	55079.43	57406.63	58567.24	58011.67
8.53	84914.95	103012.32	113028.23	109962.16
5.69	99081.62	140841.36	157725.60	157209.52
4.27	105515.98	159733.65	194509.62	196669.66
3.41	111398.23	163831.59	222351.27	230556.79
2.84	115827.19	166145.93	242313.42	261076.48
2.44	120024.39	166655.99	257657.10	285297.39
2.13	124134.58	167330.93	270167.08	303840.93
1.90	127625.49	168573.44	281113.36	320803.28
1.71	129960.14	170252.29	289936.28	336661.69

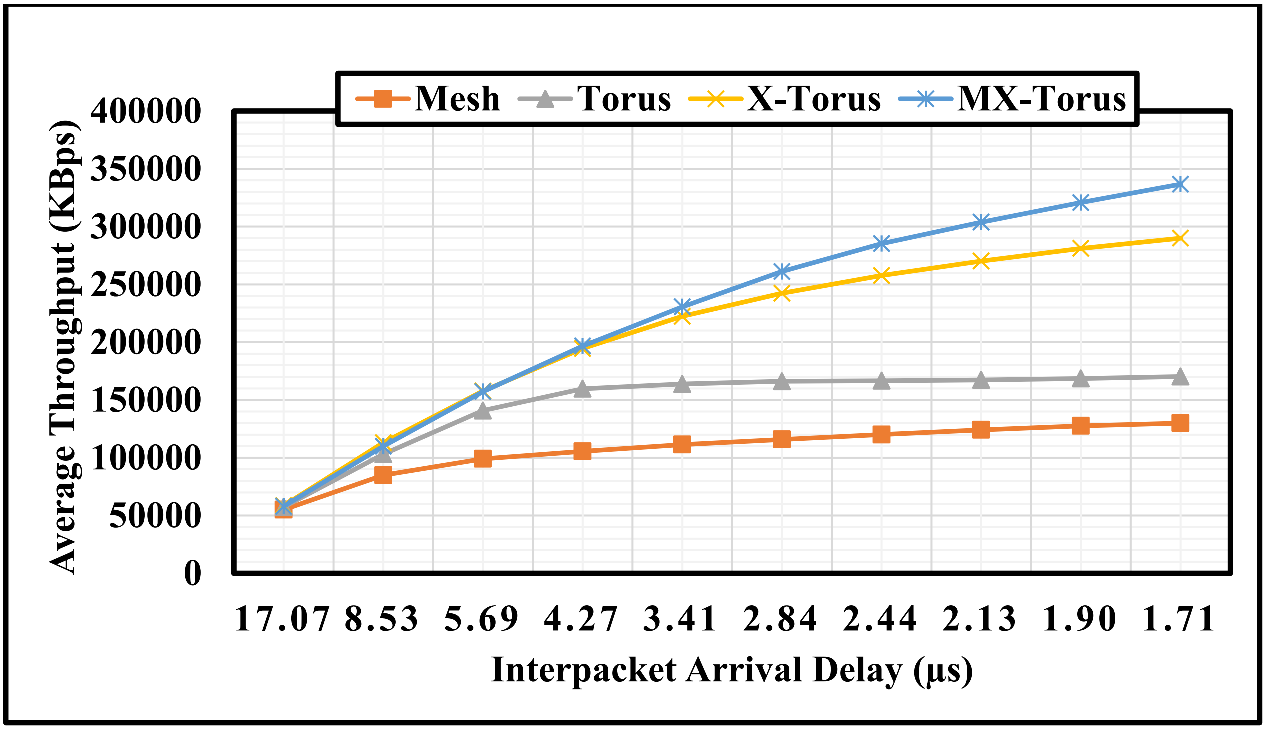

Figure 3.11 Average Throughput under Hotspot Traffic

Figure 3.11 describes the average throughput in the Mesh, Torus, X-Torus and MX-Torus based on data points (Interpacket Arrival Delay) provided in Table 3.12. The average throughput of X-Torus and MX-Torus showed a steady but significant rise over the interpacket arrival delay, while in Mesh and Torus the average throughput rose gradually. At 4.27 the average throughput of Torus, X-Torus and MX-Torus were 159733.65 KBps, 194509.62 KBps and 196669.66 KBps respectively. MX-Torus' throughput rose sharply throughout the time, exceeding X-Torus, Torus and Mesh topology's throughput and reaching almost 336661.69 KBps at the end of the period. In the meantime, the average throughput of Torus and Mesh increased until 3.41 and then remained same till 1.71.

Table 3.13 Average Hop Count under Uniform Traffic

Interpacket Arrival Delay(us)	Mesh	Torus	X-Torus	MX-Torus
17.07	3.18904	2.40158	1.95245	1.84000
8.53	2.86184	2.36803	1.94882	1.83956
5.69	2.54338	2.28315	1.89571	1.78607
4.27	2.29060	2.18022	1.84574	1.73943
3.41	2.10101	2.05743	1.79810	1.70026
2.84	1.94998	1.94852	1.75056	1.66690
2.44	1.82617	1.85090	1.70613	1.63495
2.13	1.72065	1.76578	1.66496	1.60296
1.90	1.63042	1.69134	1.62696	1.57341
1.71	1.54987	1.62522	1.59081	1.54616

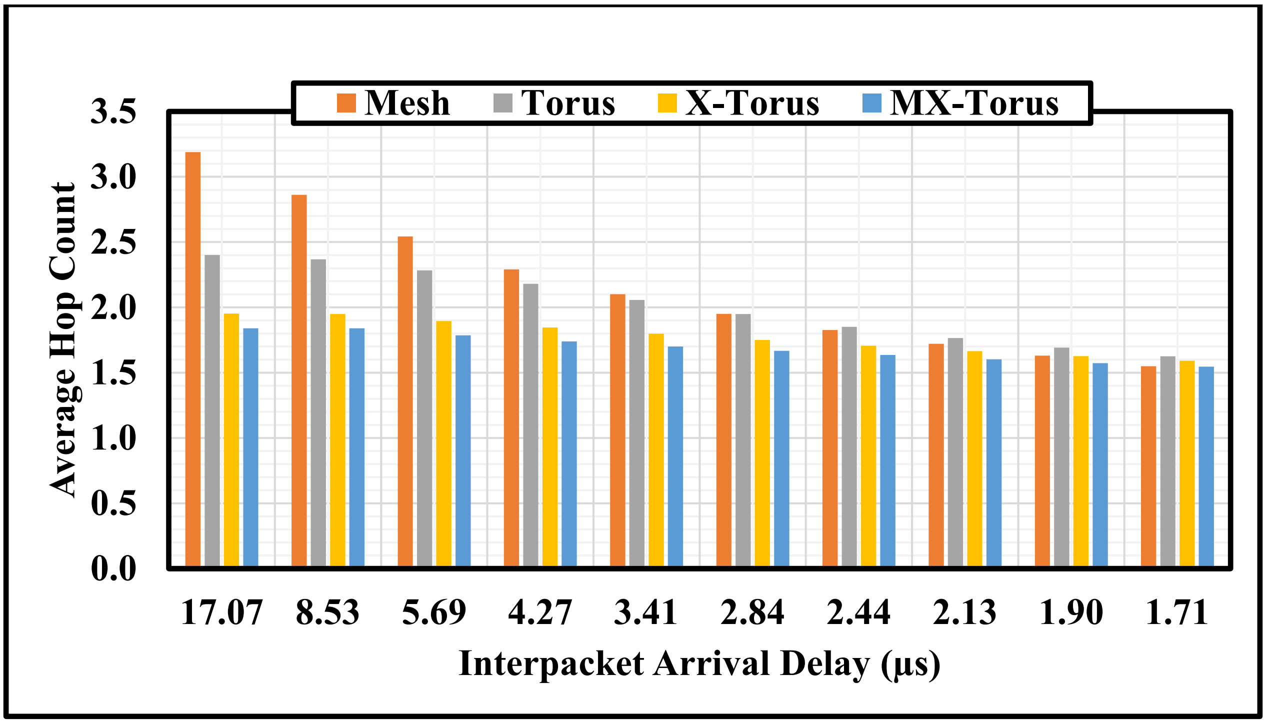

Figure 3.12 Average Hop Count under Uniform Traffic

Figure 3.12 shows the average hop count in the Mesh, Torus, X-Torus and MX-Torus based on data points (Interpacket Arrival Delay) provided in Table 3.13. Overall, Mesh, Torus, X-Torus and MX-Torus experienced a downwards trend throughout the time.MX-Torus hop count was 1.84 at 17.07 being lower than Mesh, Torus and X-Torus topology’s hop count and continued to decline steadily to the end of the time, reaching around 1.70026 at 3.41 and hitting low point of 1.54616 at

1.71 interpacket arrival delay. Same pattern is also observed for Mesh, Torus and X-Torus topologies.

Table 3.14 Average Hop Count under Bit Complement Traffic

Interpacket Arrival Delay(us)	Mesh	Torus	X-Torus	MX-Torus
17.07	4.80	2.40	1.92	1.84
8.53	4.80	2.40	1.92	1.84
5.69	4.80	2.40	1.92	1.84
4.27	4.80	2.40	1.92	1.84
3.41	4.80	2.40	1.92	1.84
2.84	4.80	2.40	1.92	1.84
2.44	4.80	2.40	1.92	1.84
2.13	4.80	2.40	1.92	1.84
1.90	4.80	2.40	1.92	1.84
1.71	4.80	2.40	1.92	1.84

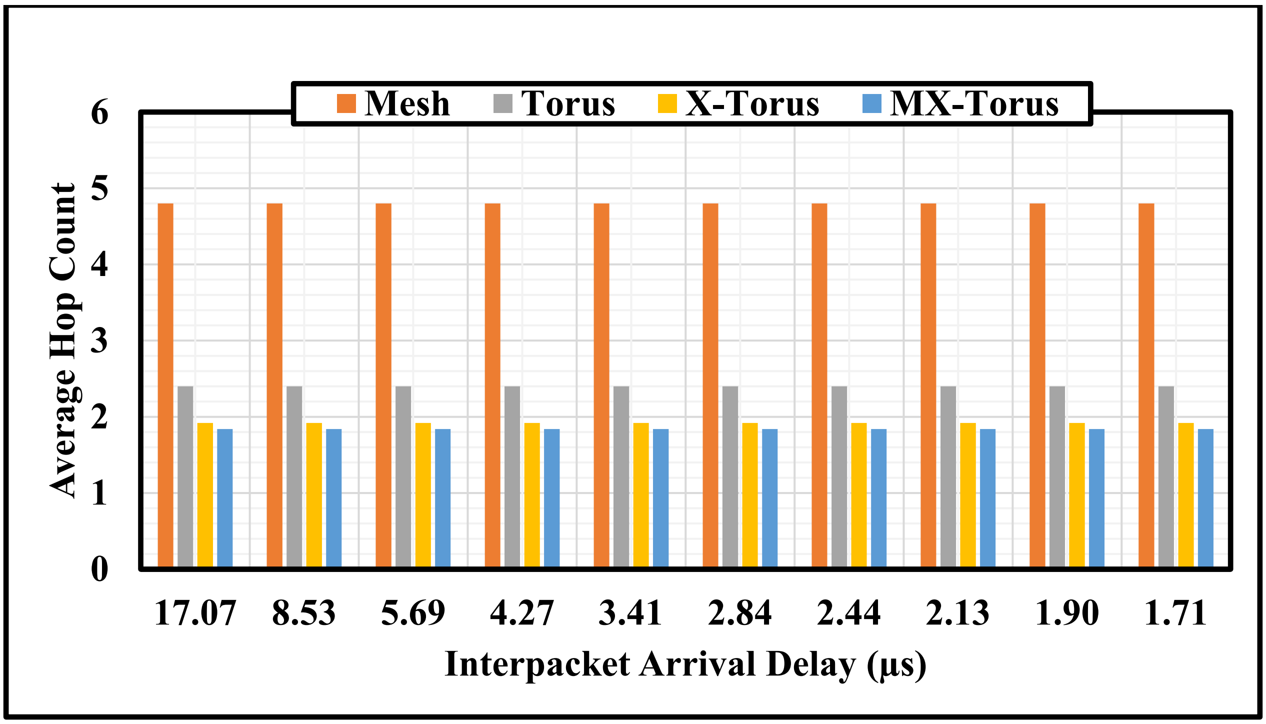

Figure 3.13 Average Hop Count under Bit Complement Traffic

Figure 3.13 describes the average hop count in the Mesh, Torus, X-Torus and MX-Torus based on data points (Interpacket Arrival Delay) provided in Table 3.14. In general, Mesh, Torus, X-Torus and MX-Torus experienced a constant trend throughout the time. The average hop count of Mesh, Torus, X-Torus, and MX-Torus was 4.80, 2.40, 1.92, and 1.84 respectively. Hop count of MX-Torus topology was smaller than other mentioned topologies and found to be 1.84 throughout the time.

Table 3.15 Average Hop Count under Neighbor Traffic

Interpacket Arrival Delay(us)	Mesh	Torus	X-Torus	MX-Torus
17.07	1.92	1.2	1.2	1.2
8.53	1.92	1.2	1.2	1.2
5.69	1.92	1.2	1.2	1.2
4.27	1.92	1.2	1.2	1.2
3.41	1.92	1.2	1.2	1.2
2.84	1.92	1.2	1.2	1.2
2.44	1.92	1.2	1.2	1.2
2.13	1.92	1.2	1.2	1.2
1.90	1.92	1.2	1.2	1.2
1.71	1.92	1.2	1.2	1.2

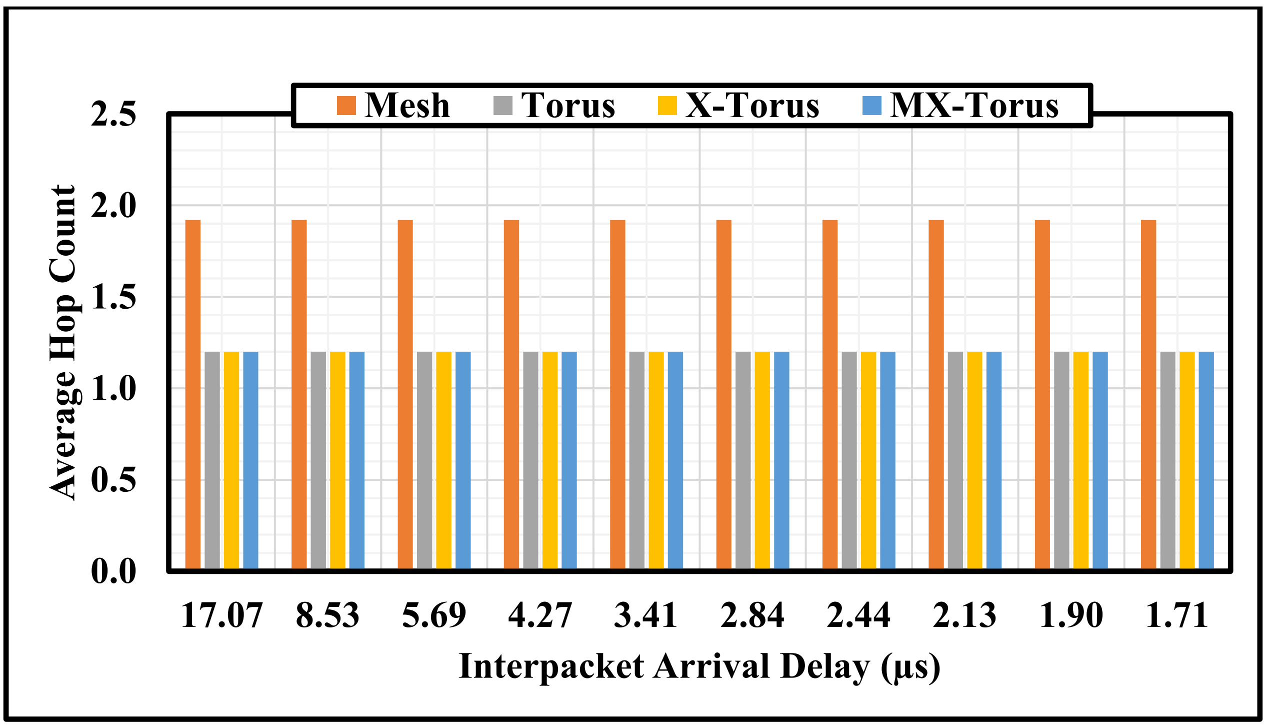

Figure 3.14 Average Hop Count under Neighbor Traffic

Figure 3.14 provides the information about the average hop count in the Mesh, Torus, X-Torus and MX-Torus based on data points (Interpacket Arrival Delay) provided in Table 3.15. In general, Mesh, Torus, X-Torus and MX-Torus experienced a constant trend throughout the time. MX-Torus, X-Torus and Torus' hop count was 1.2. Hop count of Mesh topology was higher than other mentioned topologies and found to be 1.92 throughout the time.

Table 3.16 Average Hop Count under Tornado Traffic

Interpacket Arrival Delay(us)	Mesh	Torus	X-Torus	MX-Torus
17.07	3.98713	3.19875	2.18578	1.91856
8.53	3.36801	3.01152	2.15519	1.91145
5.69	3.07285	2.81700	2.09531	1.85786
4.27	2.91139	2.68167	2.04872	1.81731
3.41	2.81101	2.58443	2.01359	1.78802
2.84	2.74287	2.51120	1.98604	1.76581
2.44	2.69375	2.45418	1.96382	1.74837
2.13	2.65661	2.40799	1.94549	1.73430
1.90	2.62774	2.37030	1.93011	1.72270
1.71	2.60471	2.33879	1.91700	1.71296

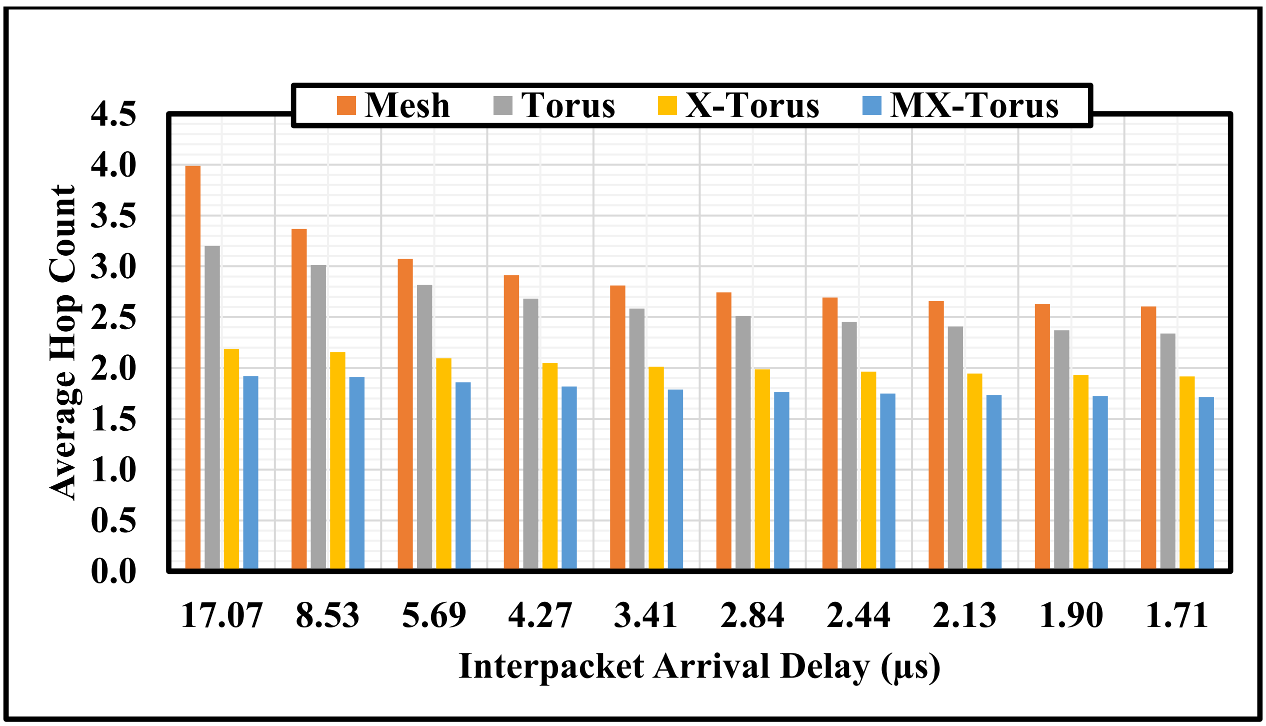

Figure 3.15 Average Hop Count under Tornado Traffic

Figure 3.15 shows the average hop count in the Mesh, Torus, X-Torus and MX-Torus based on data points (Interpacket Arrival Delay) provided in Table 3.16. Overall, Mesh, Torus, X-Torus and

MX-Torus experienced a downwards trend throughout the time. MX-Torus hop count was 1.91856 at 17.07 being lower than Mesh, Torus and X-Torus topologies and continued to decline steadily to the end of the time, reaching around 1.78802 at 3.41 and hitting low point of 1.71296 at 1.71 interpacket arrival delay. Same pattern is also observed for Mesh, Torus and X-Torus topologies.

Table 3.17 Average Hop Count under Hotspot Traffic

Interpacket Arrival Delay(us)	Mesh	Torus	X-Torus	MX-Torus
17.07	3.16822	2.44535	1.96770	1.83187
8.53	2.86186	2.37409	1.95310	1.81803
5.69	2.56708	2.30082	1.91753	1.79469
4.27	2.32875	2.19983	1.87579	1.75669
3.41	2.14647	2.08187	1.83190	1.72110
2.84	1.99732	1.97991	1.78758	1.69184
2.44	1.87563	1.88959	1.74672	1.66475
2.13	1.77394	1.80953	1.70936	1.63768
1.90	1.68587	1.73884	1.67515	1.61230
1.71	1.60838	1.67602	1.64301	1.58878

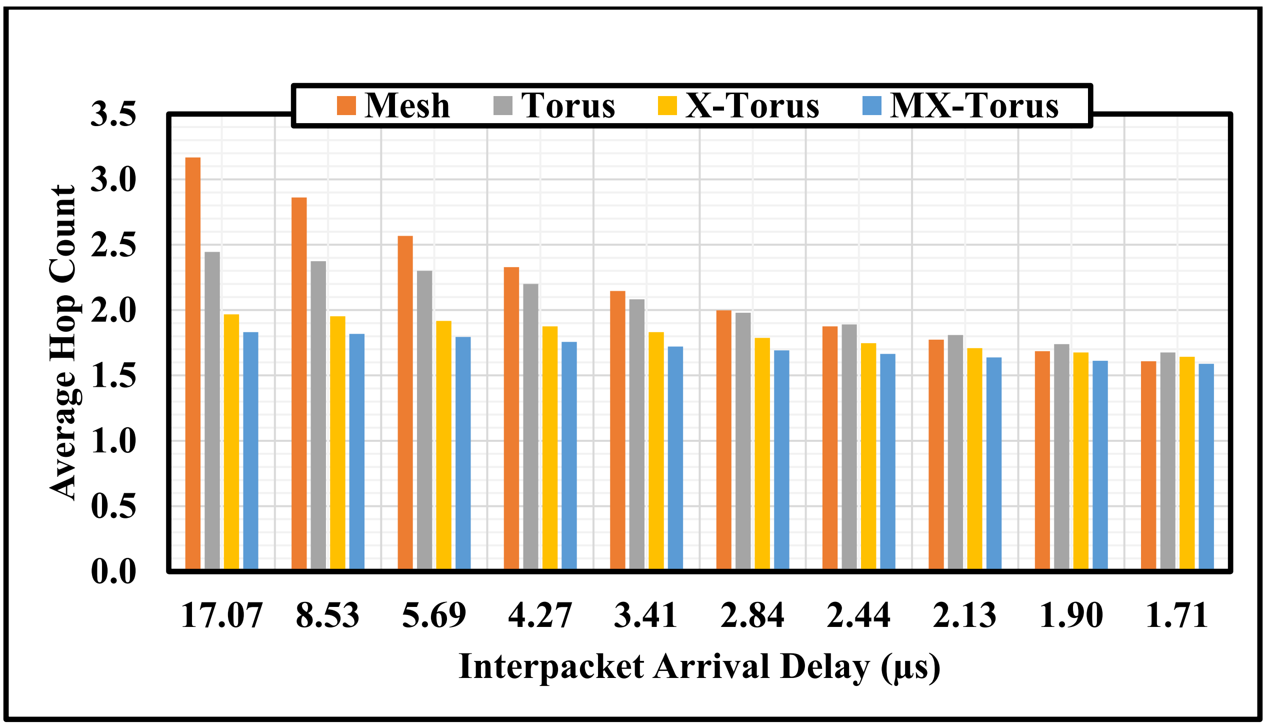

Figure 3.16 Average Hop Count under Hotspot Traffic

Figure 3.16 gives information about the average hop count in the Mesh, Torus, X-Torus and MX-Torus based on data points (Interpacket Arrival Delay) provided in Table 3.17. Overall, Mesh, Torus, X-Torus and MX-Torus experienced a downwards trend throughout the time. MX-Torus hop count was 1.83187 at 17.07 being lower than Mesh, Torus and X-Torus topologies and continued to decline steadily to the end of the time, reaching around 1.72110 at 3.41 and hitting low point of 1.58878 at 1.71 interpacket arrival delay. Same pattern is also observed for Mesh, Torus and X-Torus topologies.

3.4 Results and Discussions

This chapter presented an improved version of X-Torus topology, also called as MX-Torus. It is well suitable for parallel processing system. Based on results obtained in section 3.3, the proposed topology provided better performance in comparison to Mesh, Torus, and X-torus topologies. Maximum reduction in end to end delay obtained was 79.75% under uniform traffic. A major improvement of 28.12% in average throughput was observed under tornado traffic while average hop count was found to reduce by 12.22% under the influence of tornado traffic. The proposed topology has better path diversity and can handle fault tolerance of network in an efficient way. This topology inherits the properties of Torus as well as X-Torus topologies. In terms of degree, diameter, path diversity, average distance and bisectional bandwidth, the proposed topology resulted better performance. Due to its enhanced feature and performance it could be a better option for the large scale of parallel computing system. In addition, further exploration within the topology could result high performance in various quality of service parameter because of an efficient inter processor communication.

CHAPTER 4
CENTER CONCENTRATED X-TORUS TOPOLOGY

4.1 Introduction and Motivation

Today, topology design is one of the significant feature which is mainly used in parallel computers [59], directly impacting the performance of interconnection network [10, 60]. Although, it has better flow control technique or routing algorithm, however, performance of interconnection networks relies on efficient bandwidth [61]. Therefore, today researchers focus to design new topologies with adequate bisection bandwidth. In addition, designing topology plays an important role to make it at one time on network on chip, without modification like other key features such as routing algorithm and flow control technique. Therefore, this motivated us and prompted us to design simple and efficient topology in comparison to the current topology. As discussed earlier, that topology is of two types: one regular and another irregular; however, the con of our study was only the regular topology. Regular topology consists of various nodes, in which each node contains two things: first routing element and others processing unit. Empirical data reported that initial wires could be routed to the destination to improve the performance, and this routing of wires could vary application to application. However, the disadvantage of this process is that it is not cost-effective. To overcome this problem, the concept of tile based architecture (route packets not wires) is an efficient as well as effective approach to design network on chip for various processing cores [12]. Using this concept researchers have designed various topology [40, 62]. Therefore, we proposed a new topology to overcome the drawbacks of center concentrated topologies to improve the overall performance of the system. This topology is an improved version of torus family which takes lesser time to send packets from source node to destination node and provides high processing power to parallel computers. The detailed discussion about the proposed topology is presented in section 4.2. In section 4.3 simulation setup parameters and in section 4.4 result have been discussed, respectively.

4.2 Center Concentrated X-Torus Topology(CCX-Torus)

The proposed topology is another variant of X-Torus topology. In this topology, extra links have been introduced to reduce the distance between nodes. For odd parity of proposed topology, only one center node while for even parity, 4 center nodes. Equation 4.1, Equation 4.2, Equation 4.3, and Equation 4.4 was used to calculate center nodes of even parity topology which is as follows:

$$Center\ node1 = \frac{n^2}{2} - \frac{n}{2} \tag{4.1}$$

$$Center\ node2 = \frac{n^2}{2} + \frac{n}{2} \tag{4.2}$$

$$Center\ node3 = \frac{n^2}{2} - \frac{n}{2} - 1 \tag{4.3}$$

$$Center\ node2 = \frac{n^2}{2} + \frac{n}{2} - 1 \tag{4.4}$$

Similarly, the single center node for odd parity topology is given by Equation 4.5

$$Center\ node = \frac{n^2 - 1}{2} \tag{4.5}$$

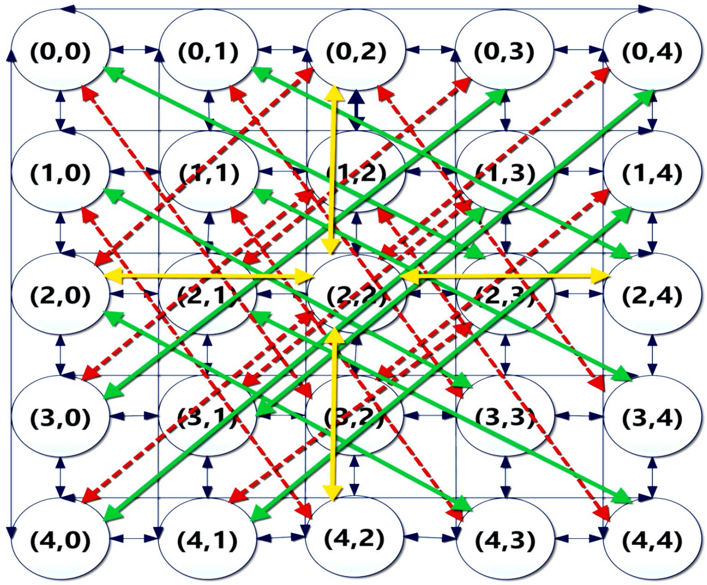

Figure 4.1 CCX-Torus Topology

Figure 4.1 represents CCX-Torus Topology of odd parity and nodes are represented in coordinated form. If it is represented in node id form, the node coordinate (2,2) is equivalent to node id 12. This is a 5x5 odd parity topology, representing the value of n as 5. Using the given equation node id of center node will be 12 which is used to make extra link to connect the nodes in horizontal and vertical edge nodes to reduce the hop count which was originally more than one hop count. This helps to reduce the hop count by one and affects the system performance to a good level. Topological properties of modified CCX-Torus are compared with Mesh, Torus and X-Torus in terms of number of nodes, diameter, bisection width, number of links, degree of various nodes, and path diversity. This comparison is described in Table 4.1.

Table 4.1 Topological Properties

Characteristics	Mesh	Torus	X-Torus	CCX-Torus
Number of Nodes	n^2	n^2	n^2	n^2
Diameter	2n-2	n-1	$\left\lfloor\frac{n}{2}\right\rfloor + 1$	$\left\lfloor\frac{n}{2}\right\rfloor + 1$
Bisection Width	n	2n	n^2 -n	n^2 +n+1
Number of links	$2n^2$-2n	$2n^2$	$2n^2+2\left\lfloor\frac{n}{2}\right\rfloor^2+3\left\lfloor\frac{n}{2}\right\rfloor + 1$	$2n^2+2\left\lfloor\frac{n}{2}\right\rfloor^2+3\left\lfloor\frac{n}{2}\right\rfloor + 2n + 5$
Degree of Nodes	2, 3, 4	4	5, 6	6, 7, 10
Path Diversity	yes	yes	yes	yes

4.3 Testbed for Testing Center Concentrated X-Torus Topology

In this section authors analyzed the performance parameters of proposed topology. These parameters are average throughput, average end to end delay and average hop count. Therefore, authors employed windows 10 operating system of 32-bit, equipped intel® Core™ i3 CPU M330@2.13 GHZ with 4.00 GB and 2.99GB usable. Another key component used was OMNET++ simulator, a component-based C++ simulation library and framework (extensible and modular), which is primarily used for building network simulator based on the Eclipse IDE [58]. All the key parameters for simulation purpose are shown in Table 4.2, that describes the dimension of topology i.e. order of 5. In addition, packet size was 1024 bytes, with data rate of 1 Gbps. Warm up time and simulation time was 0.5ms and 0.5s, respectively. Five traffic patterns such as uniform, bit complement, neighbor, tornado and hotspot were considered. The proposed topology is compared with mesh, torus and x-torus topologies. Simulation results are presented in table and via graphical layout. Table 4.3 to Table 4.7 presents Average End to End Delay under various traffic such as uniform, bit complement, neighbor, tornado and hotspot respectively.

Similarly, Table 4.8 to Table 4.12 present Average Throughput, and Table 4.13 to Table 4.17 show the Average Hop Count for the same traffic patterns. Corresponding to the tabular information, Figure 4.2 to Figure 4.16 represent the graph of Average End to End Delay, Average Throughput and Average Hop Count for the same traffic patterns. Results pertaining to the present section are discussed in section 4.4.

Table 4.2 Simulation Parameters

S.no.	Parameter Name	Value
1	Rows	5
2	Coloums	5
3	Packet size	1024 bytes
4	Data rate	1Gbps
5	Simulation time	0.5 s
6	Warm up time	0.5ms
7	Simulator	OMNeT++
8	Traffic Type	Uniform, Bit Complement, Neighbor, Tornado, Hotspot
9	Link Delay	0.1 ms
10	Routing Algorithm	Table based Shortest Path (Static)

Table 4.3 Average End to End Delay under Uniform Traffic

Interpacket Arrival Delay(us)	Mesh	Torus	X-Torus	CCX-Torus
17.07	0.00239	0.00026	0.00021	0.00019
8.53	0.04343	0.01099	0.00136	0.00020
5.69	0.07309	0.02602	0.01646	0.00922
4.27	0.08986	0.05327	0.02579	0.01829
3.41	0.09687	0.08094	0.03678	0.02344
2.84	0.10134	0.09833	0.04901	0.02897
2.44	0.10392	0.10886	0.05994	0.03831
2.13	0.10559	0.11489	0.06929	0.04958
1.90	0.10740	0.11835	0.07682	0.05866
1.71	0.10951	0.12012	0.08408	0.06573

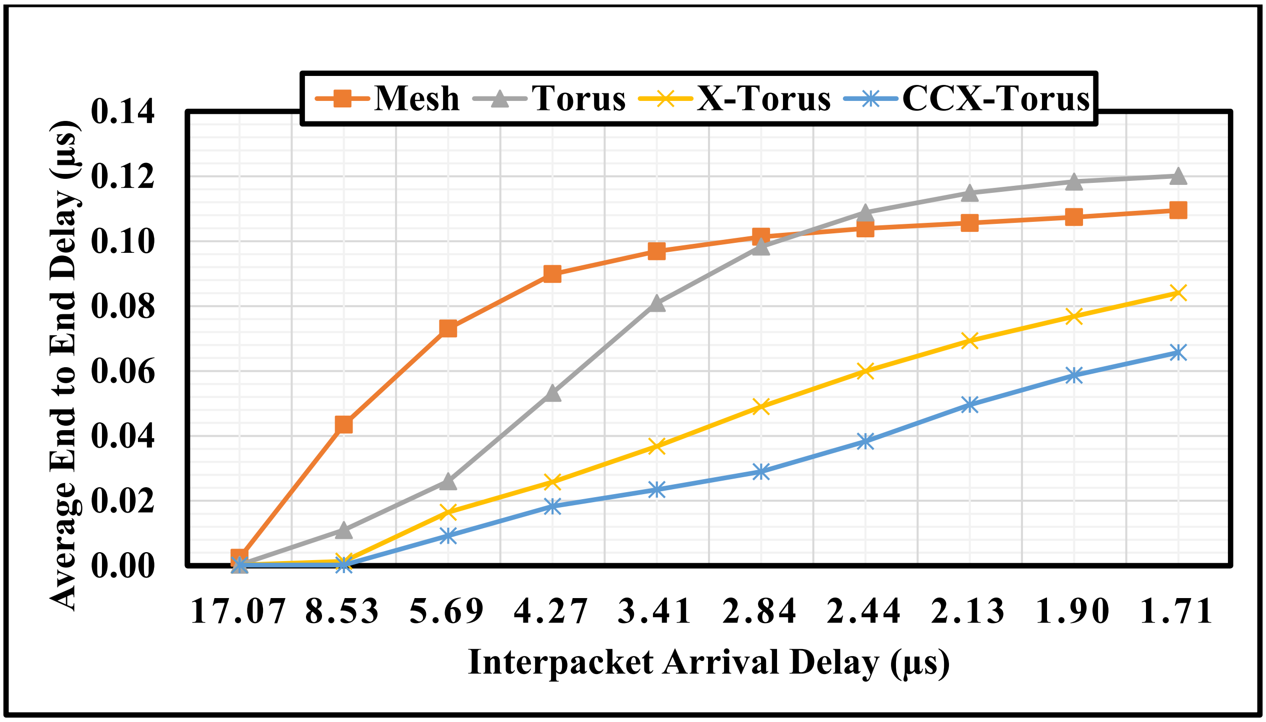

Figure 4.2 Average End to End Delay under Uniform Traffic

Figure 4.2 shows the average end to end delay in the Mesh, Torus, X-Torus and CCX-Torus based on data points (Interpacket Arrival Delay) provided in Table 4.3. The end to end delay of Torus and Mesh showed a steady but significant rise over the interpacket arrival delay, while in X-torus and MX-Torus the end to end delay rose gradually. At 8.53 the end to end delay of X-Torus and CCX-Torus were 0.00136 µs and 0.00020 µs respectively. Torus's delay increased sharply throughout the time, exceeding Mesh topology delay and reaching almost 0.12012 µs at the end of the time. In the meantime, the average end to end delay of X-torus and CCX-Torus grew to 0.01646 μs and 0.00922 μs at 5.69 respectively and then increased gradually until 1.71.

Table 4.4 Average End to End Delay under Bit Complement Traffic

Interpacket Arrival Delay(us)	Mesh	Torus	X-Torus	CCX-Torus
17.07	0.07701	0.00026	0.00021	0.00020
8.53	0.15696	0.07684	0.03851	0.03850
5.69	0.18478	0.12905	0.10127	0.10126
4.27	0.20007	0.15683	0.13598	0.13597
3.41	0.20916	0.17350	0.15681	0.15680
2.84	0.21510	0.18462	0.17069	0.17069
2.44	0.21933	0.19256	0.18061	0.18061
2.13	0.22249	0.19853	0.18806	0.18805
1.90	0.22499	0.20317	0.19384	0.19384
1.71	0.22706	0.20688	0.19847	0.19847

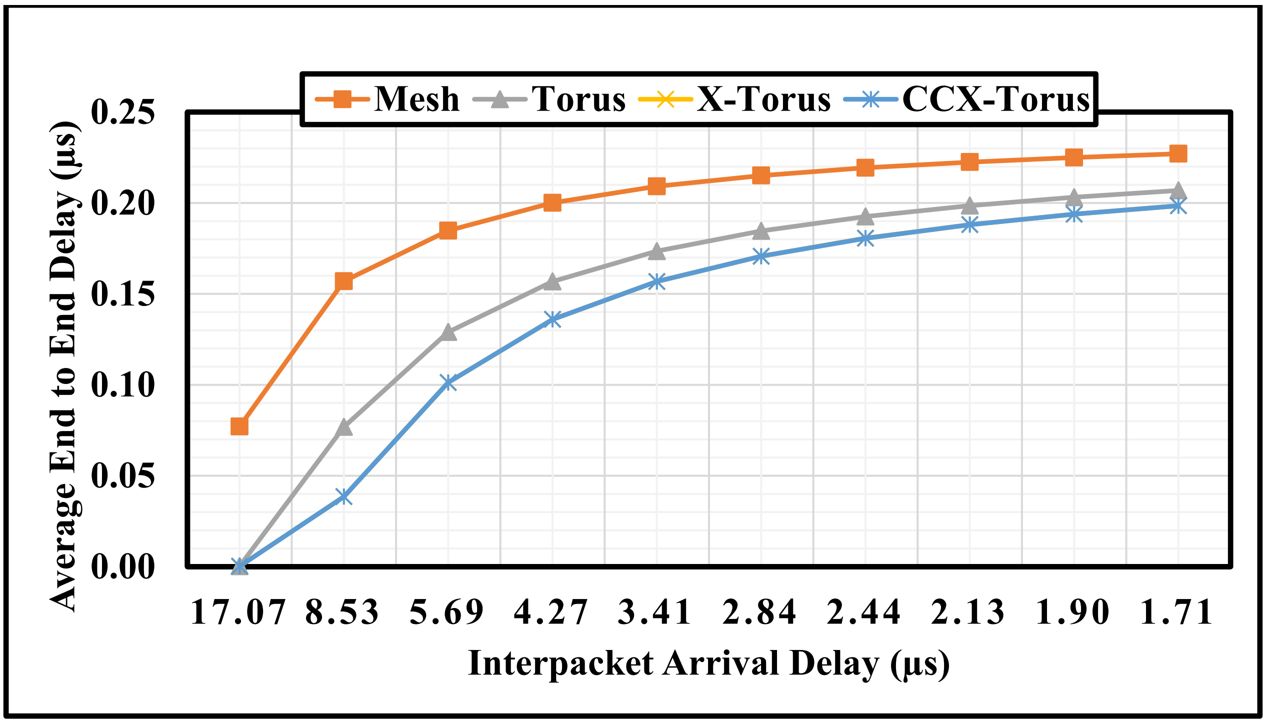

Figure 4.3 Average End to End Delay under Bit Complement Traffic

Figure 4.3 illustrates the average end to end delay in the Mesh, Torus, X-Torus and CCX-Torus based on data points (Interpacket Arrival Delay) provided in Table 4.4. In starting the end to end delay of Mesh, Torus, X-Torus and CCX-Torus rose significantly and then increased gently over the interpacket arrival delay. At 17.07 the end to end delay of Torus, X-Torus and CCX-Torus were nearly 0.00026 µs, 0.00021 µs and 0.00020 µs respectively. Delay of X-Torus and CCX-Torus are almost same. Mesh's delay increased sharply at 2.84, exceeding Torus topology's delay and reaching almost 0.22706 µs at the end of the time. In the meantime, the average end to end delay of X-Torus and CCX-Torus gradually grew to 0.19847 µs at 1.71.

Table 4.5 Average End to End Delay under Neighbor Traffic

Interpacket Arrival Delay(us)	Mesh	Torus	X-Torus	CCX-Torus
17.07	0.00021	0.00013	0.00013	0.00013
8.53	0.00977	0.00971	0.00971	0.00971
5.69	0.08349	0.08344	0.08344	0.08344
4.27	0.12515	0.12509	0.12509	0.12509
3.41	0.15014	0.15009	0.15009	0.15009
2.84	0.16681	0.16675	0.16675	0.16675
2.44	0.17871	0.17865	0.17865	0.17865
2.13	0.18764	0.18758	0.18758	0.18758
1.90	0.19458	0.19453	0.19453	0.19453
1.71	0.20014	0.20008	0.20008	0.20008

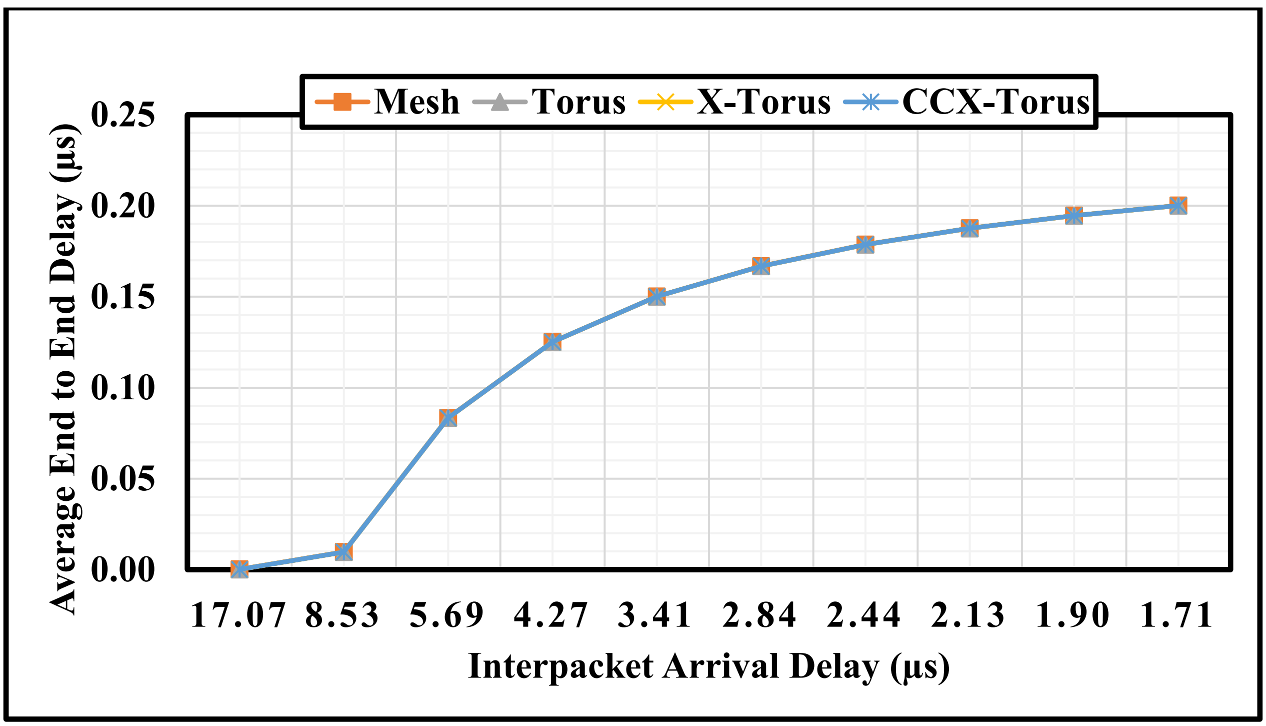

Figure 4.4 Average End to End Delay under Neighbor Traffic

Figure 4.4 gives information about the average end to end delay in the Mesh, Torus, X-Torus and CCX-Torus based on data points (Interpacket Arrival Delay) provided in Table 4.5. In starting the end to end delay of Mesh, Torus, X-Torus and CCX-Torus experienced a gradual rise and then increased dramatically over the interpacket arrival delay. At 17.07 the end to end delay of Torus, X-Torus and CCX-Torus were 0.00013 μs. Delay of Torus, X-Torus and CCX-Torus are almost same. Mesh's delay increased gently at 8.53, exceeding Torus, X-Torus, and CCX-Torus topologies' delay and reaching almost 0.20014 μs at the end of the time. In the meantime, the average end to end delay of Torus, X-Torus and CCX-Torus gradually grew to 0.20008 μs at 1.71.

Table 4.6 Average End to End Delay under Tornado Traffic

Interpacket Arrival Delay(us)	Mesh	Torus	X-Torus	CCX-Torus
17.07	0.19439	0.22507	0.05845	0.02592
8.53	0.21023	0.23804	0.12732	0.08894
5.69	0.21678	0.24140	0.16533	0.13748
4.27	0.22135	0.24304	0.18432	0.16331
3.41	0.22482	0.24414	0.19553	0.17873
2.84	0.22755	0.24492	0.20304	0.18910
2.44	0.22975	0.24556	0.20848	0.19663
2.13	0.23157	0.24613	0.21266	0.20237
1.90	0.23310	0.24654	0.21599	0.20693
1.71	0.23440	0.24701	0.21872	0.21064

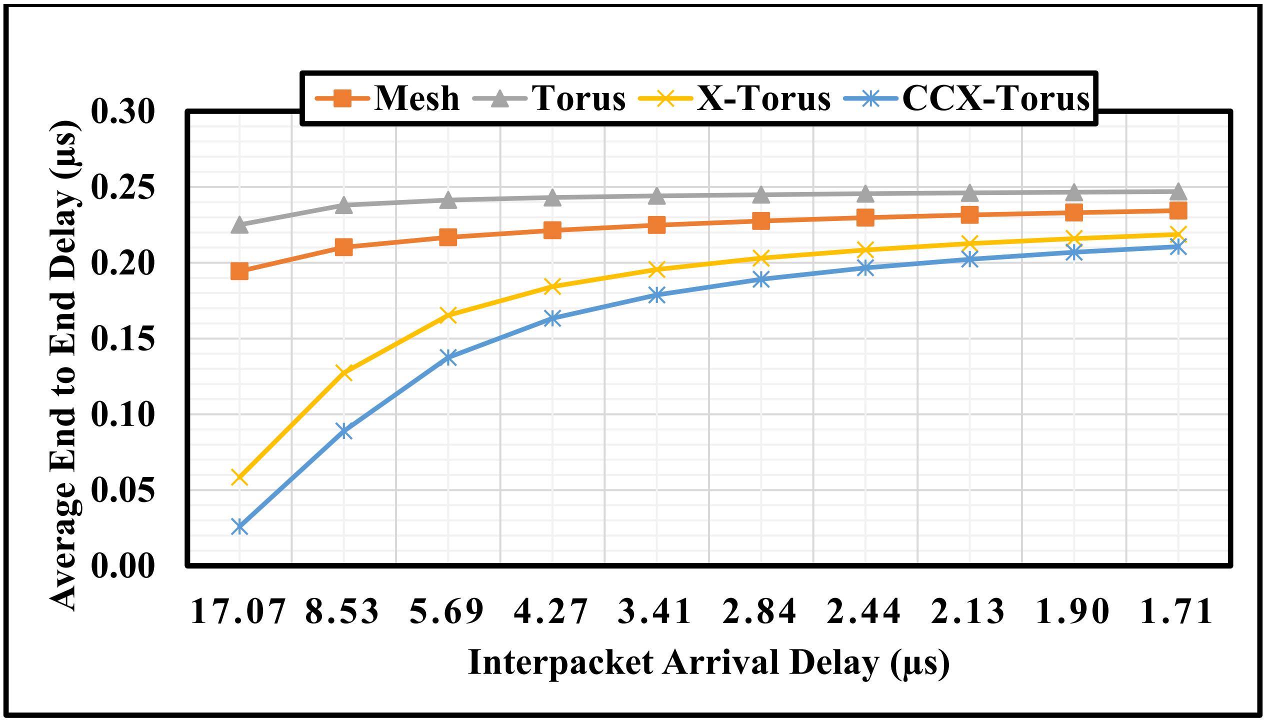

Figure 4.5 Average End to End Delay under Tornado Traffic

Figure 4.5 describes the average end to end delay in the Mesh, Torus, X-Torus and CCX-Torus based on data points (Interpacket Arrival Delay) provided in Table 4.6. In starting the end to end delay of Mesh, Torus, X-Torus and CCX-Torus showed a gradual increase and then remained almost steady over the interpacket arrival delay. At 5.69 the end to end delay of Mesh and Torus were 0.21678 µs and 0.24140 µs and then rose gently to 0.23440 µs and 0.24701 µs at the end of time respectively. In the meantime, the average end to end delay of X-Torus and CCX-Torus increased considerably to 0.20304 µs and 0.18910 µs at 2.84 respectively and then rose gently to 0.21872 µs and 0.21064 µs at 1.71 respectively.

Table 4.7 Average End to End Delay under Hotspot Traffic

Interpacket Arrival Delay(us)	Mesh	Torus	X-Torus	CCX-Torus
17.07	0.00814	0.00227	0.00021	0.00084
8.53	0.04593	0.01680	0.00442	0.00498
5.69	0.07254	0.02580	0.01427	0.00897
4.27	0.08862	0.05244	0.02351	0.01458
3.41	0.09557	0.07955	0.03556	0.02015
2.84	0.10089	0.09610	0.04894	0.02659
2.44	0.10441	0.10782	0.06059	0.03633
2.13	0.10645	0.11513	0.07038	0.04832
1.90	0.10825	0.11958	0.07814	0.05793
1.71	0.11052	0.12225	0.08517	0.06562

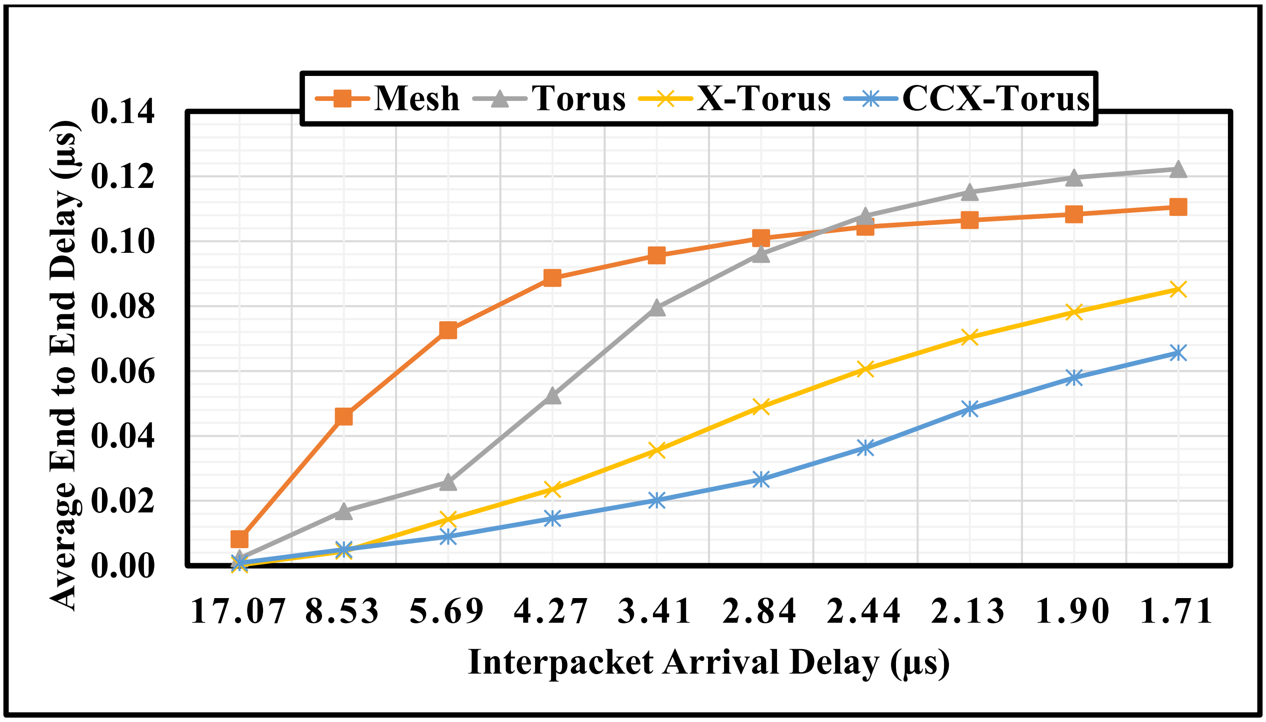

Figure 4.6 Average End to End Delay under Hotspot Traffic

Figure 4.6 indicates the average end to end delay in the Mesh, Torus, X-Torus and CCX-Torus based on data points (Interpacket Arrival Delay) provided in Table 4.7. The end to end delay of Torus and Mesh increased significantly over the interpacket arrival delay, while in X-torus and CCX-Torus the end to end delay rose gradually. At 8.53 the end to end delay of X-Torus and CCX-Torus were 0.00442 µs and 0.00498 µs respectively. Torus's delay increased sharply throughout the time, exceeding Mesh topology delay and reaching almost 0.12225 µs at the end of the time. In the meantime, the average end to end delay of X-torus and CCX-Torus grew to 0.02351 µs and 0.01458 µs at 4.27 respectively and then experienced a significant rise until 1.71.

Table 4.8 Average Throughput under Uniform Traffic

Interpacket Arrival Delay(us)	Mesh	Torus	X-Torus	CCX-Torus
17.07	58061.52	58562.48	58567.83	58570.07
8.53	90463.99	111655.91	116571.67	117138.38
5.69	103380.32	149966.08	161123.82	168016.77
4.27	109051.59	169907.41	199149.69	210279.52
3.41	114654.80	173585.03	229235.51	250053.75
2.84	119390.38	174261.33	251530.40	285781.14
2.44	124002.47	174215.65	268674.31	313048.03
2.13	128169.50	174991.33	282345.84	332246.35
1.90	131656.36	176402.24	293940.15	348941.91
1.71	134041.22	178252.85	302420.27	364127.39

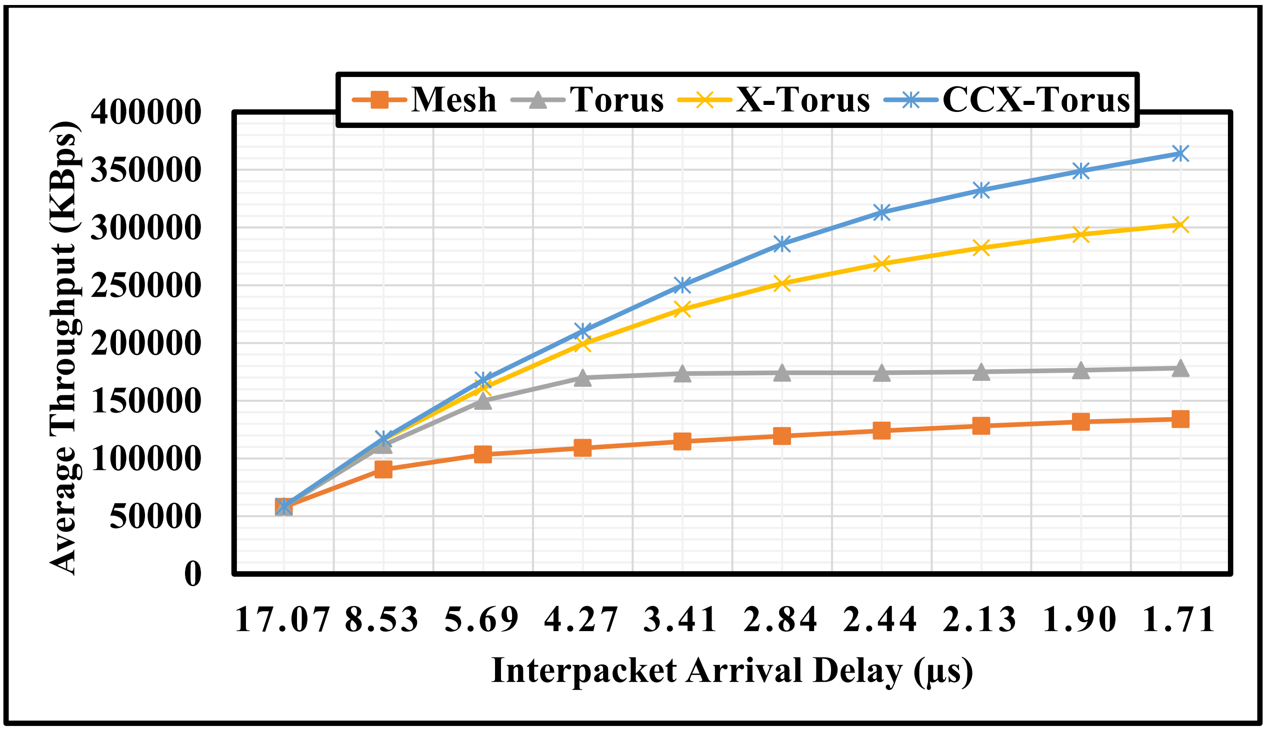

Figure 4.7 Average Throughput under Uniform Traffic

Figure 4.7 shows the average throughput in the Mesh, Torus, X-Torus and CCX-Torus based on data points (Interpacket Arrival Delay) provided in Table 4.8. The average throughput of X-Torus and CCX-Torus showed a steady but significant rise over the interpacket arrival delay, while in Mesh and Torus the average throughput rose gradually. At 8.53 the average throughput of Torus, X-Torus and MX-Torus were 111655.91 KBps, 116571.67 KBps and 117138.38 KBps respectively. CCX-Torus' throughput increased sharply throughout the time, exceeding X-Torus, Torus and Mesh topologies throughput and reaching almost 364127.39 KBps at the end of the period. In the meantime, the average throughput of Torus and Mesh grew to 169907.41 KBps and 109051.59 KBps at 4.27 respectively and then remained steady until 1.71.

Table 4.9 Average Throughput under Bit Complement Traffic

Interpacket Arrival Delay(us)	Mesh	Torus	X-Torus	CCX-Torus
17.07	40592.63	58562.07	58568.31	58569.43
8.53	43720.08	81217.67	99178.61	99180.69
5.69	46063.72	85123.30	104646.45	104648.69
4.27	47182.17	87466.04	106990.20	106992.28
3.41	48462.01	89809.23	109333.94	109336.02
2.84	49977.18	92153.01	111677.81	111679.89
2.44	51658.28	94496.26	114021.34	114023.42
2.13	53459.83	96839.91	116365.16	116367.24
1.90	55352.38	99183.41	118709.00	118711.08
1.71	57314.37	101526.67	121052.42	121054.50

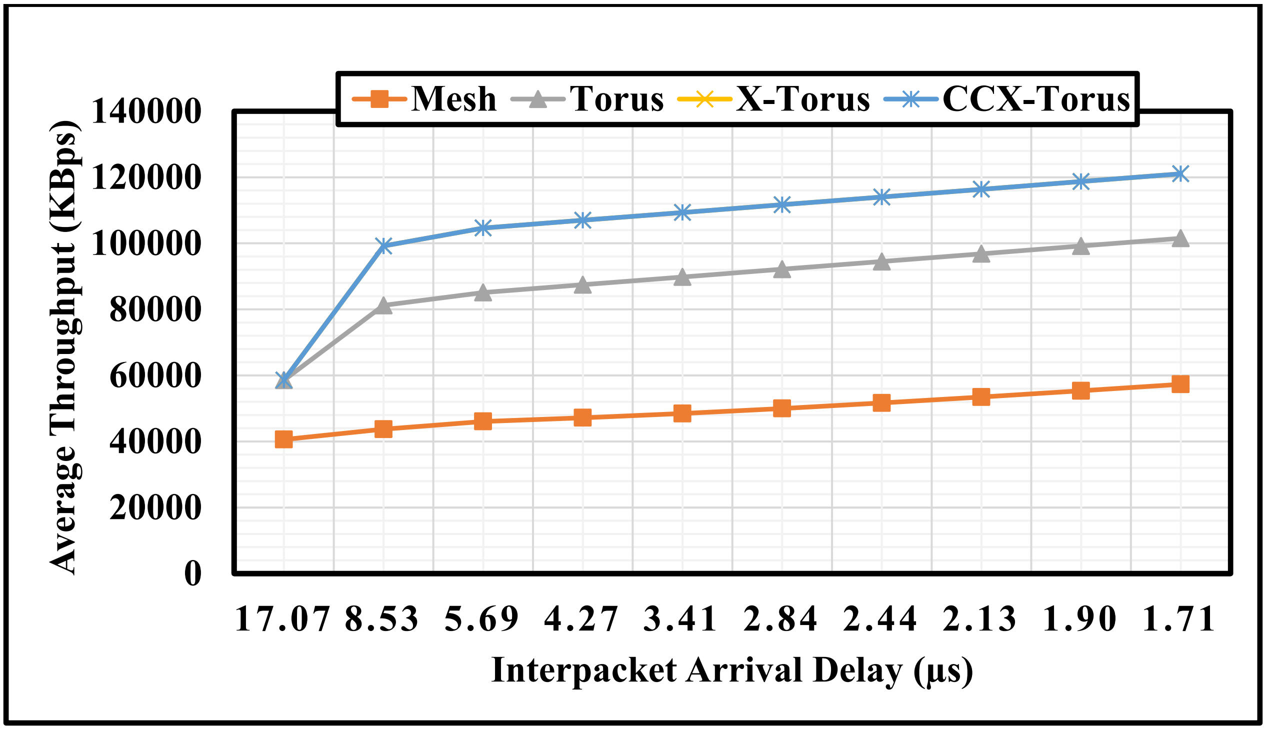

Figure 4.8 Average Throughput under Bit Complement Traffic

Figure 4.8 illustrates the average throughput in the Mesh, Torus, X-Torus and CCX-Torus based on data points (Interpacket Arrival Delay) provided in Table 4.9. In starting, the average throughput of Torus, X-Torus and CCX-Torus experienced a significant trend over the interpacket arrival delay and then increased, while in Mesh the average throughput rose gradually. At 8.53 the average throughput of X-Torus and CCX-Torus were nearly 99178.61 KBps and 99180.69 KBps respectively. CCX-Torus' throughput increased sharply throughout the time, exceeding X-Torus, Torus and Mesh topologies throughput and reaching almost 121054.50 KBps at the end of the period. In the meantime, the average throughput of Torus and Mesh grew to 81217.67 KBps and 43720.08 KBps at 8.53 respectively and then rose gradually until 1.71.

Table 4.10 Average Throughput under Neighbor Traffic

Interpacket Arrival Delay(us)	Mesh	Torus	X-Torus	CCX-Torus
17.07	58568.31	58577.43	58577.43	58577.43
8.53	112654.63	112666.64	112666.64	112666.64
5.69	117145.45	117158.00	117158.00	117158.00
4.27	117145.31	117157.92	117157.92	117157.92
3.41	117145.43	117157.91	117157.91	117157.91
2.84	117145.48	117158.03	117158.03	117158.03
2.44	117145.25	117158.05	117158.05	117158.05
2.13	117145.32	117158.12	117158.12	117158.12
1.90	117145.40	117158.28	117158.28	117158.28
1.71	117145.39	117158.19	117158.19	117158.19

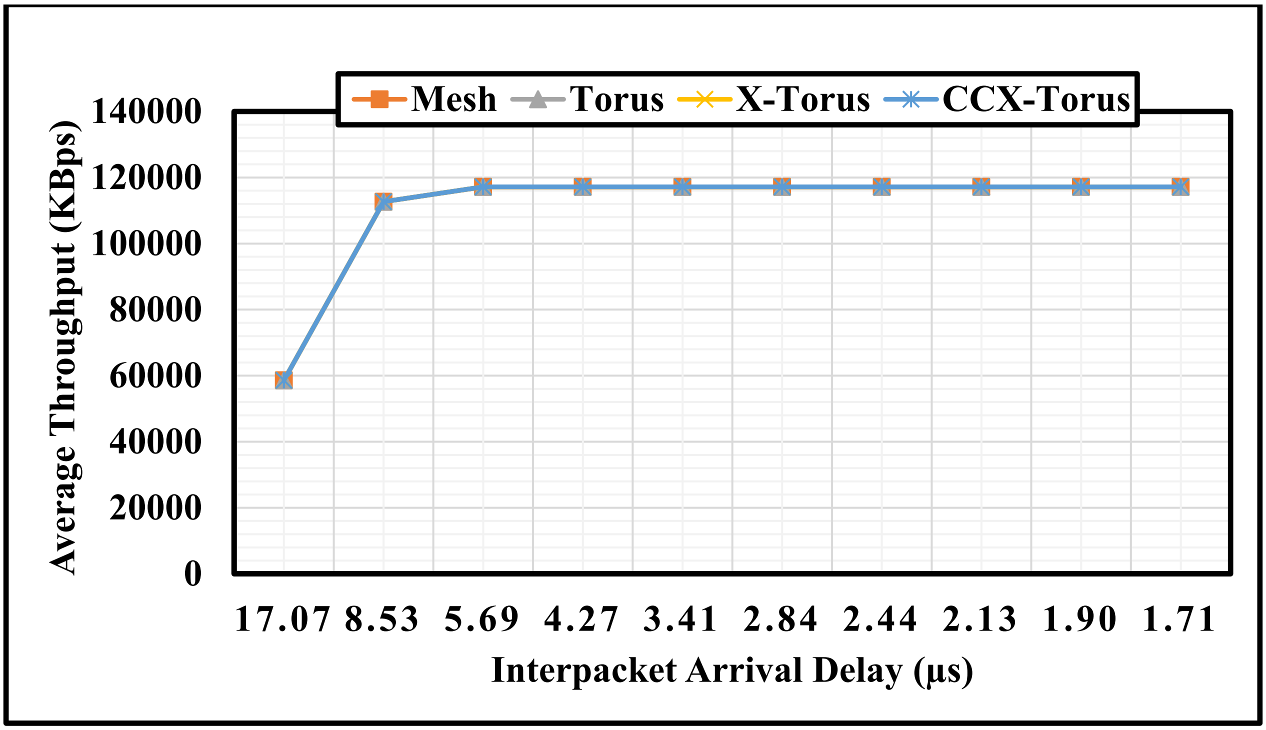

Figure 4.9 Average Throughput under Neighbor Traffic

Figure 4.9 gives information about the average throughput in the Mesh, Torus, X-Torus and CCX-Torus based on data points (Interpacket Arrival Delay) provided in Table 4.10. In starting, the average throughput of Mesh, Torus, X-Torus and CCX-Torus showed a significant rise over the interpacket arrival delay and then remained same. At 17.07 the average throughput of Torus, X-Torus and CCX-Torus were 58577.43 KBps and then reaching 117158.19 KBps at 1.71. In the meantime, the average throughput of Mesh grew to 112654.63 KBps at 8.53 and then rose gradually to 117145.45 KBps at 5.69. However, it remained the same at end of the time.

Table 4.11 Average Throughput under Tornado Traffic

Interpacket Arrival Delay(us)	Mesh	Torus	X-Torus	CCX-Torus
17.07	58638.35	29333.92	202842.77	254251.56
8.53	57048.39	25691.91	209932.06	305694.52
5.69	55897.85	23245.90	204420.76	303743.95
4.27	55028.57	21235.91	199964.78	299041.57
3.41	54348.57	19553.92	196649.58	295520.37
2.84	53802.63	18121.94	194060.99	292769.48
2.44	53354.28	16891.96	191974.76	290546.13
2.13	52979.91	15823.97	190252.08	288710.31
1.90	52661.98	14880.00	188805.14	287165.91
1.71	52389.58	17054.36	187565.91	285851.07

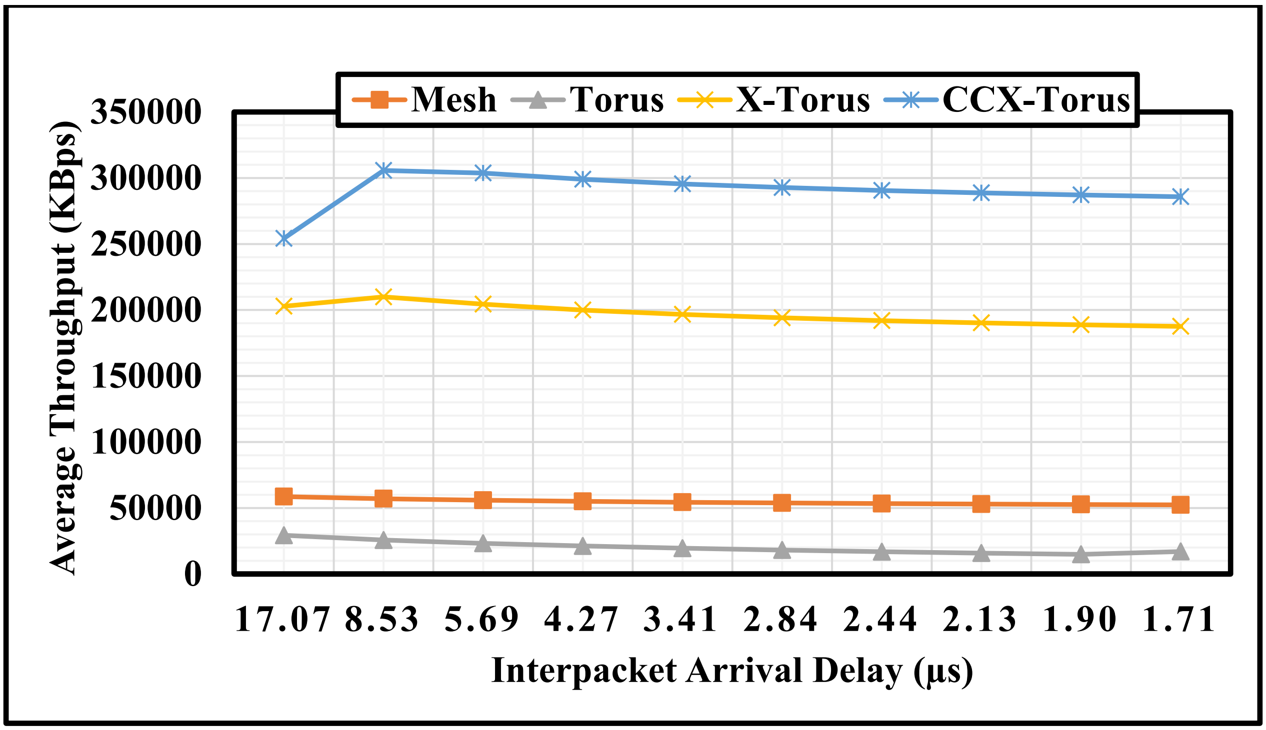

Figure 4.10 Average Throughput under Tornado Traffic

Figure 4.10 indicates the average throughput in the Mesh, Torus, X-Torus and CCX-Torus based on data points (Interpacket Arrival Delay) mentioned in Table 4.11. In the beginning, the average throughput of X-Torus and CCX-Torus showed a significant rise and then experienced a gradual decrease over the interpacket arrival delay, while in Mesh and Torus the average throughput fell gradually. At 8.53 the average throughput of X-Torus and CCX-Torus were 209932.06 KBps, and 305694.52 KBps respectively. CCX-Torus' throughput decreased gently throughout the time, exceeding X-Torus, Torus and Mesh topologies throughput and reaching almost 285851.07 KBps at the end of the period. In the meantime, the average throughput of Mesh, Torus and X-Torus declined gradually to 55897.85 KBps, 23245.90 KBps and 204420.76 KBps at 5.69 respectively and then remained almost same until 1.71.

Table 4.12 Average Throughput under Hotspot Traffic

Interpacket Arrival Delay(us)	Mesh	Torus	X-Torus	CCX-Torus
17.07	55079.43	57406.63	58567.24	58111.11
8.53	84914.95	103012.32	113028.23	110424.73
5.69	99081.62	140841.36	157725.60	159832.36
4.27	105515.98	159733.65	194509.62	203354.02
3.41	111398.23	163831.59	222351.27	242363.03
2.84	115827.19	166145.93	242313.42	276777.26
2.44	120024.39	166655.99	257657.10	303543.07
2.13	124134.58	167330.93	270167.08	322022.19
1.90	127625.49	168573.44	281113.36	338283.74
1.71	129960.14	170252.29	289936.28	352990.04

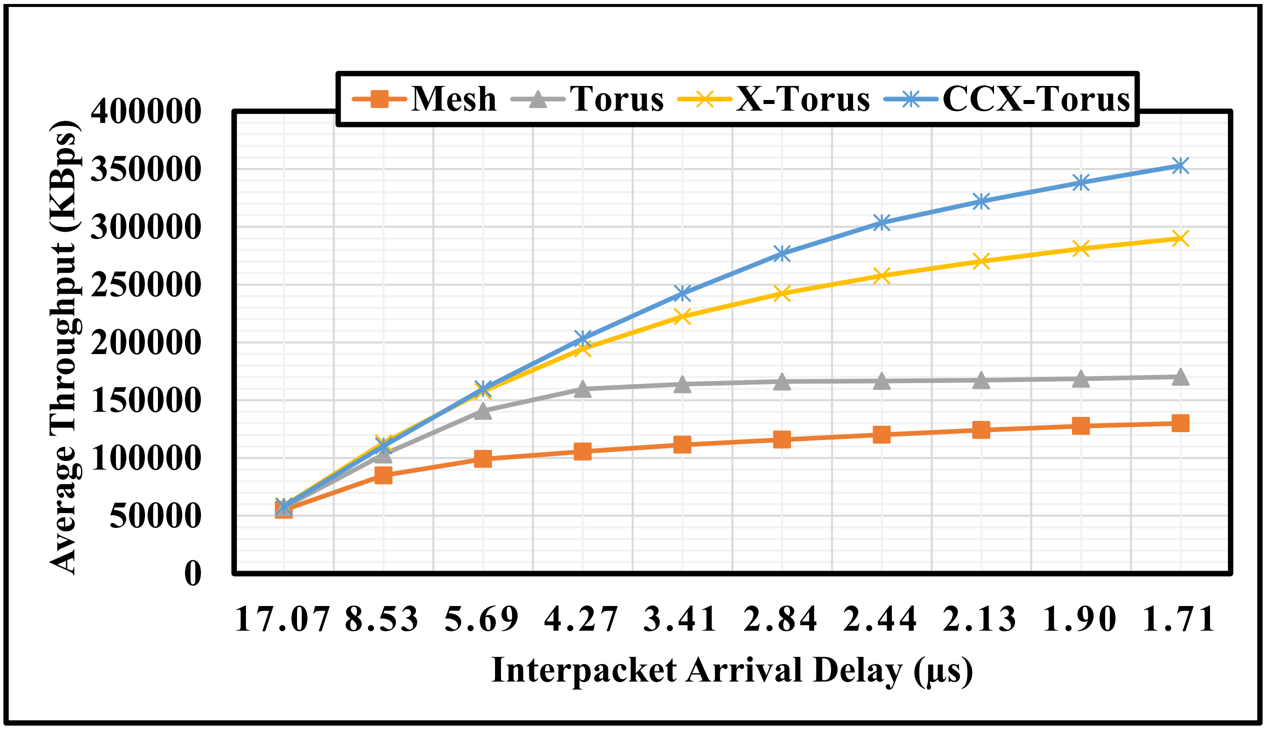

Figure 4.11 Average Throughput under Hotspot Traffic

Figure 4.11 describes the average throughput in the Mesh, Torus, X-Torus and CCX-Torus based on data points (Interpacket Arrival Delay) provided in Table 4.12. The average throughput of X-Torus and CCX-Torus showed a steady but significant rise over the interpacket arrival delay, while in Mesh and Torus the average throughput rose gradually. At 4.27 the average throughput of Torus, X-Torus and CCX-Torus were 159733.65 KBps, 194509.62 KBps and 203354.02 KBps respectively. CCX-Torus' throughput rose sharply throughout the time, exceeding X-Torus, Torus and Mesh topologies throughput and reaching almost 352990.04 KBps at the end of the period. In the meantime, the average throughput of Torus and Mesh increased until 3.41 and then remained same till 1.71.

Table 4.13 Average Hop Count under Uniform Traffic

Interpacket Arrival Delay(us)	Mesh	Torus	X-Torus	CCX-Torus
17.07	3.18904	2.40158	1.95245	1.78852
8.53	2.86184	2.36803	1.94882	1.78850
5.69	2.54338	2.28315	1.89571	1.75959
4.27	2.29060	2.18022	1.84574	1.71887
3.41	2.10101	2.05743	1.79810	1.68744
2.84	1.94998	1.94852	1.75056	1.65981
2.44	1.82617	1.85090	1.70613	1.63161
2.13	1.72065	1.76578	1.66496	1.60179
1.90	1.63042	1.69134	1.62696	1.57336
1.71	1.54987	1.62522	1.59081	1.54652

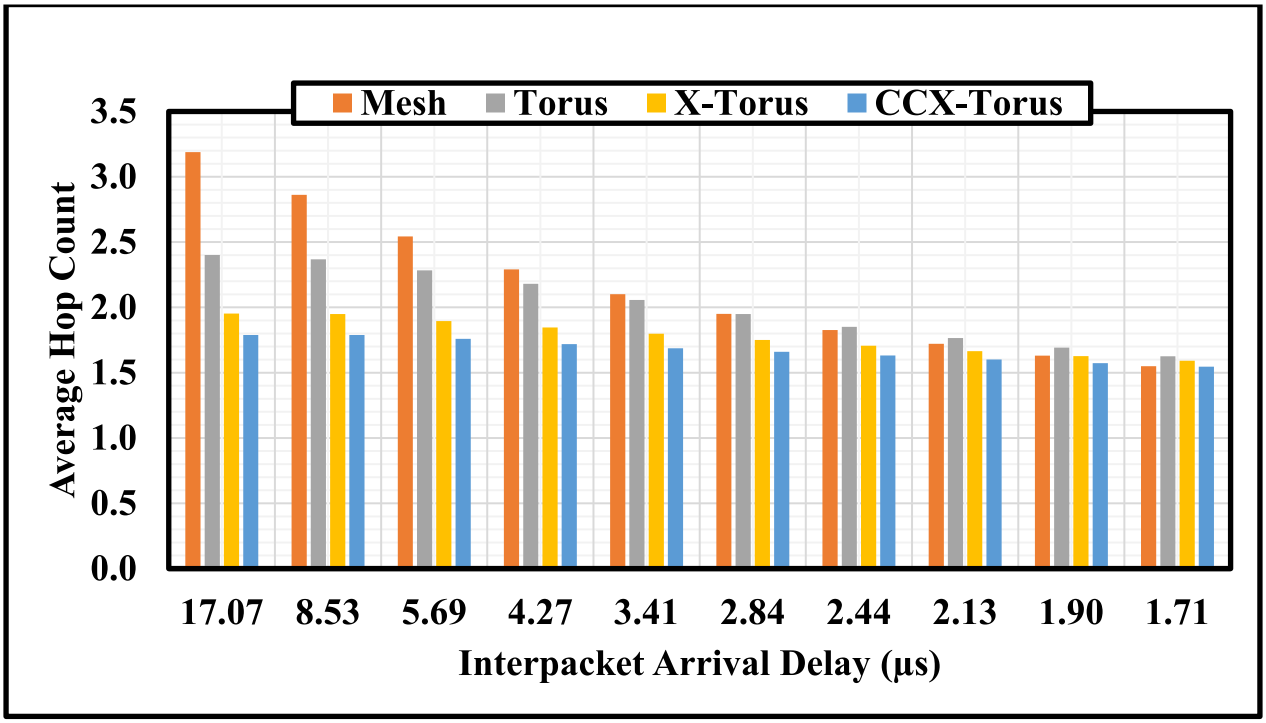

Figure 4.12 Average Hop Count under Uniform Traffic

Figure 4.12 shows the average hop count in the Mesh, Torus, X-Torus and CCX-Torus based on data points (Interpacket Arrival Delay) provided in Table 4.13. Overall, Mesh, Torus, X-Torus and CCX-Torus experienced a downwards trend throughout the time. CCX-Torus hop count was 1.78852 at 17.07 being lower than Mesh, Torus and X-Torus topologies and continued to decline steadily to the end of the time, reaching around 1.68744 at 3.41 and hitting low point of 1.54652 at

1.71 interpacket arrival delay. Same pattern is also observed for Mesh, Torus and X-Torus topologies.

Table 4.14 Average Hop Count under Bit Complement Traffic

Interpacket Arrival Delay(us)	Mesh	Torus	X-Torus	CCX-Torus
17.07	4.80	2.40	1.92	1.84
8.53	4.80	2.40	1.92	1.84
5.69	4.80	2.40	1.92	1.84
4.27	4.80	2.40	1.92	1.84
3.41	4.80	2.40	1.92	1.84
2.84	4.80	2.40	1.92	1.84
2.44	4.80	2.40	1.92	1.84
2.13	4.80	2.40	1.92	1.84
1.90	4.80	2.40	1.92	1.84
1.71	4.80	2.40	1.92	1.84

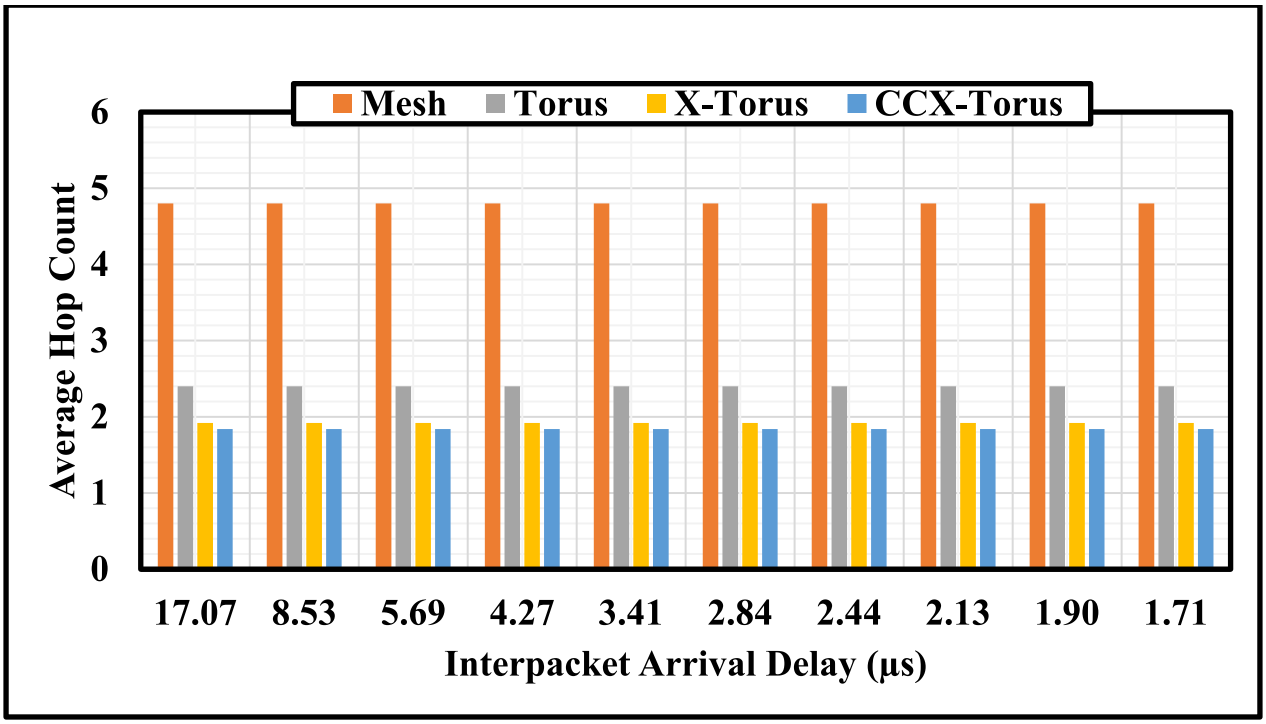

Figure 4.13 Average Hop Count under Bit Complement Traffic

Figure 4.13 describes the average hop count in the Mesh, Torus, X-Torus and CCX-Torus based on data points (Interpacket Arrival Delay) provided in Table 4.14. In general, Mesh, Torus, X-Torus and CCX-Torus experienced a constant trend throughout the time. The average hop count of

Mesh, Torus, X-Torus, and CCX-Torus was 4.8, 2.4, 1.92, and 1.84 respectively. Hop count of CCX-Torus topology was smaller than other mentioned topologies and found to be 1.84 throughout the time.

Table 4.15 Average Hop Count under Neighbor Traffic

Interpacket Arrival Delay(us)	Mesh	Torus	X-Torus	CCX-Torus
17.07	1.92	1.2	1.2	1.2
8.53	1.92	1.2	1.2	1.2
5.69	1.92	1.2	1.2	1.2
4.27	1.92	1.2	1.2	1.2
3.41	1.92	1.2	1.2	1.2
2.84	1.92	1.2	1.2	1.2
2.44	1.92	1.2	1.2	1.2
2.13	1.92	1.2	1.2	1.2
1.90	1.92	1.2	1.2	1.2
1.71	1.92	1.2	1.2	1.2

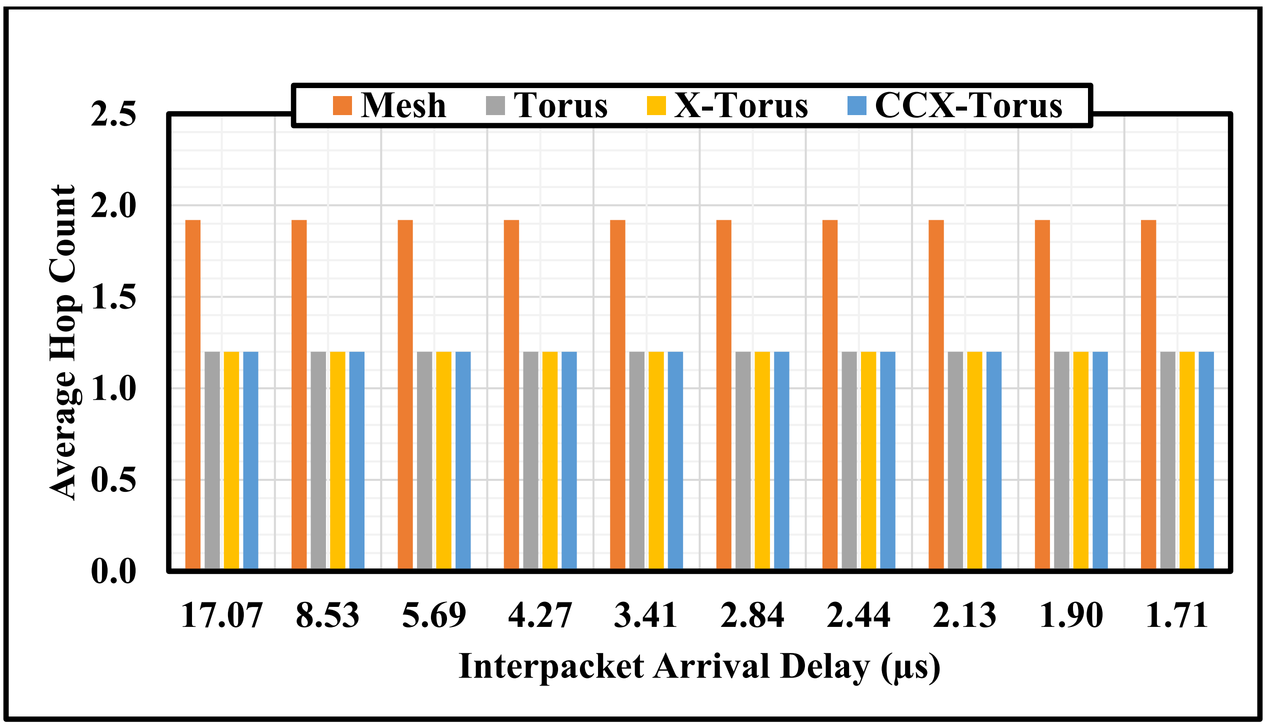

Figure 4.14 Average Hop Count under Neighbor Traffic

Figure 4.14 provides the information about the average hop count in the Mesh, Torus, X-Torus and CCX-Torus based on data points (Interpacket Arrival Delay) provided in Table 4.15. In general, Mesh, Torus, X-Torus and CCX-Torus experienced a constant trend throughout the time. CCX-Torus, X-Torus and Torus' hop count was 1.2. Hop count of Mesh topology was higher than other mentioned topologies and found to be 1.92 throughout the time.

Table 4.16 Average Hop Count under Tornado Traffic

Interpacket Arrival Delay(us)	Mesh	Torus	X-Torus	CCX-Torus
17.07	3.98713	3.19875	2.18578	1.88461
8.53	3.36801	3.01152	2.15519	1.89306
5.69	3.07285	2.81700	2.09531	1.84297
4.27	2.91139	2.68167	2.04872	1.80417
3.41	2.81101	2.58443	2.01359	1.77585
2.84	2.74287	2.51120	1.98604	1.75417
2.44	2.69375	2.45418	1.96382	1.73699
2.13	2.65661	2.40799	1.94549	1.72303
1.90	2.62774	2.37030	1.93011	1.71144
1.71	2.60471	2.33879	1.91700	1.70166

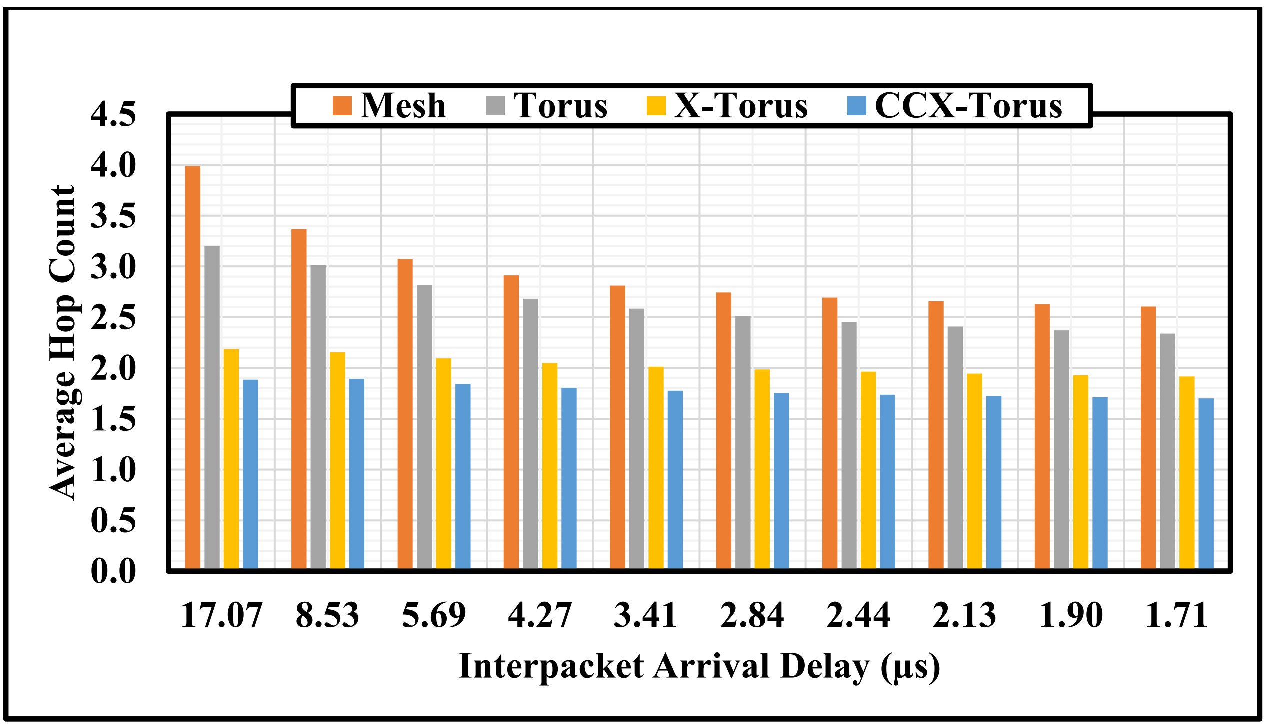

Figure 4.15 Average Hop Count under Tornado Traffic

Figure 4.15 shows the average hop count in the Mesh, Torus, X-Torus and CCX-Torus based on data points (Interpacket Arrival Delay) provided in Table 4.16. Overall, Mesh, Torus, X-Torus and

CCX-Torus experienced a downwards trend throughout the time. CCX-Torus hop count was 1.88461 at 17.07 being lower than Mesh, Torus and X-Torus topologies and continued to decline steadily to the end of the time, reaching around 1.77585 at 3.41 and hitting low point of 1.70166 at 1.71 interpacket arrival delay. Same pattern is also observed for Mesh, Torus and X-Torus topologies.

Table 4.17 Average Hop Count under Hotspot Traffic

Interpacket Arrival Delay(us)	Mesh	Torus	X-Torus	CCX-Torus
17.07	3.16822	2.44535	1.96770	1.77910
8.53	2.86186	2.37409	1.95310	1.76854
5.69	2.56708	2.30082	1.91753	1.75354
4.27	2.32875	2.19983	1.87579	1.72891
3.41	2.14647	2.08187	1.83190	1.70280
2.84	1.99732	1.97991	1.78758	1.67830
2.44	1.87563	1.88959	1.74672	1.65463
2.13	1.77394	1.80953	1.70936	1.62866
1.90	1.68587	1.73884	1.67515	1.60376
1.71	1.60838	1.67602	1.64301	1.58039

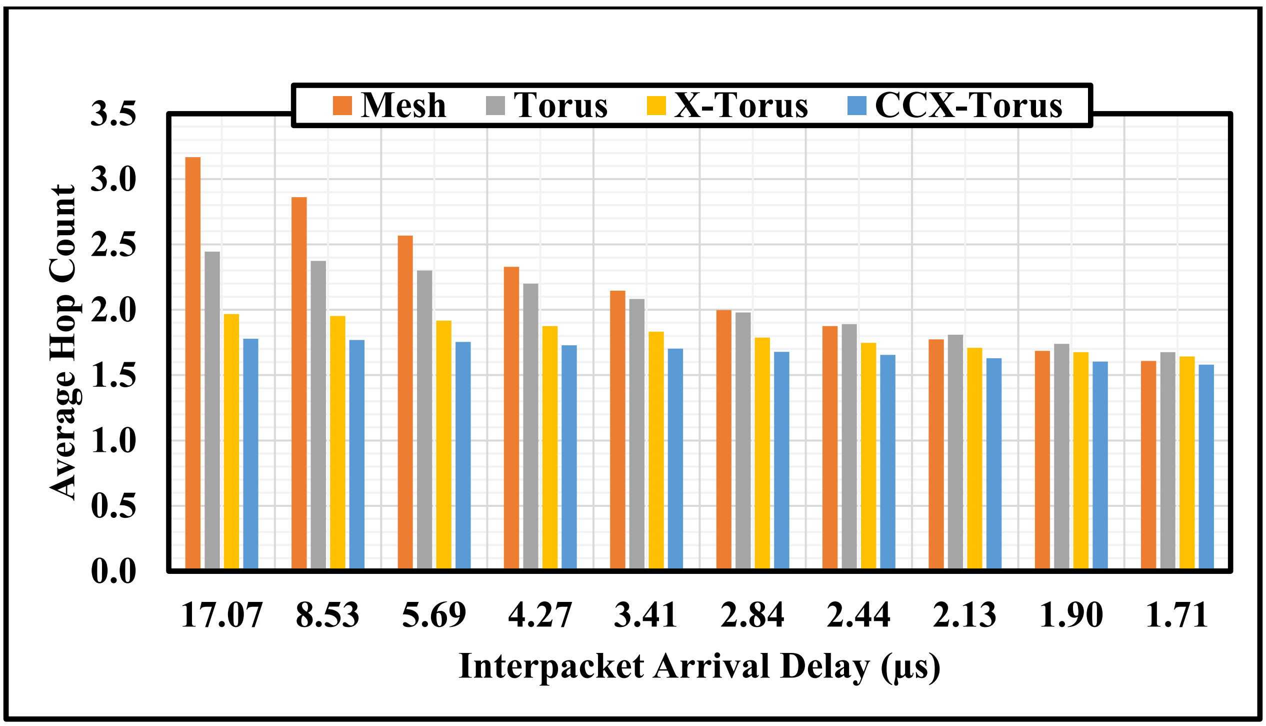

Figure 4.16 Average Hop Count under Hotspot Traffic

Figure 4.16 gives information about the average hop count in the Mesh, Torus, X-Torus and CCX-Torus based on data points (Interpacket Arrival Delay) provided in Table 4.17. Overall, Mesh, Torus, X-Torus and CCX-Torus experienced a downwards trend throughout the time. CCX-Torus hop count was 1.77910 at 17.07 being lower than Mesh, Torus and X-Torus topologies and continued to decline steadily to the end of the time, reaching around 1.70280 at 3.41 and hitting low point of 1.58039 at 1.71 interpacket arrival delay. Same pattern is also observed for Mesh, Torus and X-Torus topologies.

4.4 Results and Discussions

This chapter resulted an improved version of X-Torus topology which we call CCX-Torus. It is well suitable for parallel processing system. Based on results obtained, as presented in section 4.3, the proposed topology revealed better performance in comparison to Mesh, Torus, and X-torus topologies. Maximum reduction in end to end delay obtained was 85.27% under uniform traffic. A major improvement was observed in average throughput which was 34.38% under tornado traffic while average hop count was reduced by 13.77% under the influence of tornado traffic. The proposed topology provides better path diversity and can handle fault tolerance of network in efficient way. This topology inherits the properties of Torus as well as X-Torus topologies. In terms of degree, diameter, path diversity, average distance and bisectional bandwidth the proposed topology provided better performance. Due to its enhanced feature and performance it could be a better option for the large scale of parallel computing system. In addition, because of efficient inter processor communication, it is proposed to explore the topology to attain high performance in varied quality of service parameter.

CHAPTER 5
A HEXAGONAL X-TORUS TOPOLOGY FOR NETWORKS ON CHIP

5.1 Introduction and Motivation

With an immense advancement of VLSI technology, the demand for the parallel computer has increased. There are two types of parallel computers, namely, tightly coupled and loosely coupled. Loosely coupled parallel computers are also known as "multicomputer system". A major demand of multicomputer system has been observed and reported due to its capacity and capability of multitasking such as scientific calculation, earthquake prediction, and weather forecasting etc. Designing of large scale parallel systems are in huge requirement to significantly reduce the communication delay between the processors to handle vast calculations [63]. Therefore, in such scenario an intelligent interconnection network becomes an integral concern [64]. The interconnection network should adhere cost-effectiveness, reliability, and scalability. Published data reveals that many aspects of the interconnection network have been examined in the past such as performance, routing, fault tolerance, and scalability [1, 47, 60, 65]. Since the performance norms, cost constraints, execution environment and application area of each network is different, therefore it is difficult to decide the best network. The torus based topologies are most suitable because of its topological properties like low diameter, high bisectional bandwidth, path diversity and fault tolerance[66]. In recent years, to increase the performance of torus topology, various changes have been made. Many variations of Torus topology have occurred such as Twisted Torus, Folded Torus, Dtorus, CC-Torus, X-Torus and XX-Torus [22, 40, 44]. These are some variants of Torus topology which are designed to reduce the diameter and average distance of the network. However, multicomputer system requires higher processing power that increases the communication overhead to a great level [67]. Therefore, to overcome such mentioned problems, an impressive topology is required which have improved topological properties like diameter, average distance, degree, path diversity, and node degree for faster interprocessor communication. In this chapter, we have proposed an improved topology to achieve faster communication between processors. This topology is called Hexagonal X-Torus topology. This chapter covers the proposed topology Hexagonal X-Torus and is discussed in section 5.2 and its performance is tested using parameters which are given in section 5.3 and results have been discussed in section 5.4.

5.2 Hexagonal X-Torus Topology(HX-Torus)

This proposed topology is based on X-Torus topology which is already described in CHAPTER 2. In this topology, extra links have been added to connect adjacent diagonal nodes of the four corners to reduce the hop count by one. It helped to improve the overall performance of parallel processing system. Figure 5.1 shows HX-Torus topology where nodes are represented in coordinate form and Figure 4.2 describes node id form. Extra links which are used to reduce the hop count are represented by the Equation as given below. Here, 5x5 topology was considered with ‘n’ as 5. This indicated that node id 1 is connected to node id 5 and so on. This helps to reduce the hop count by one and significantly influences the system performance to a better standard. Topological properties of HX-Torus are compared with Mesh, Torus and X-Torus in terms of number of nodes, diameter, bisection width, number of links, degree of various nodes, and path diversity [57] and is tabulated in Table 5.1.

$$1 \leftrightarrow n \tag{5.1}$$

$$n-2 \leftrightarrow 2n-1 \tag{5.2}$$

$$n(n-2) \leftrightarrow n(n-1)+1 \tag{5.3}$$

$$n(n-1)-1 \leftrightarrow n^2-2 \tag{5.4}$$

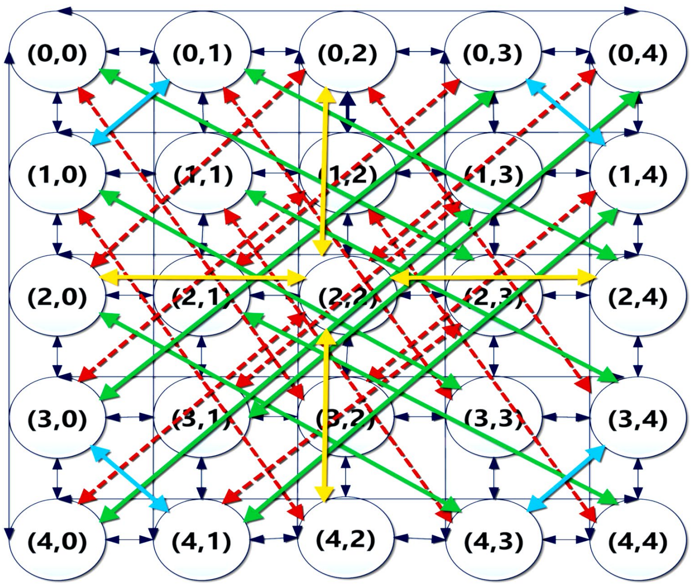

Figure 5.1 HX-Torus Topology

0	1	2	3	4
5	6	7	8	9
10	11	12	13	14
15	16	17	18	19
20	21	22	23	24

Figure 5.2 Node Id Form

Table 5.1 Topological Properties

Characteristics	Mesh	Torus	X-Torus	HX-Torus
Number of Nodes	n^2	n^2	n^2	n^2
Diameter	2n-2	n-1	$\left\lfloor \frac{n}{2} \right\rfloor + 1$	$\left\lfloor \frac{n}{2} \right\rfloor + 1$
Bisection Width	n	2n	$n^2 - n$	$n^2 + n + 1$
Number of links	$2n^2-2n$	$2n^2$	$2n^2+2\left\lfloor \frac{n}{2} \right\rfloor^2+3\left\lfloor \frac{n}{2} \right\rfloor + 1$	$2n^2+2\left\lfloor \frac{n}{2} \right\rfloor^2+3\left\lfloor \frac{n}{2} \right\rfloor + 2n + 9$
Degree of Nodes	2, 3, 4	4	5, 6	6, 7, 10
Path Diversity	yes	yes	yes	yes

5.3 Testbed for Testing Hexagonal X-Torus Topology

In this section, authors analyzed the performance parameters of proposed topology. These parameters are average throughput, average end to end delay and average hop count. Therefore, windows 10 operating system of 32-bit, equipped intel® Core™ i3 CPU M330@2.13 GHZ with 4.00 GB and 2.99GB usable was employed in the study. Another key component was OMNET++ simulator, a component-based C++ simulation library and framework (extensible and modular), primarily used for building network simulator based on the Eclipse IDE. All the proposed key parameters for simulation are shown in Table 5.2, describing the dimension of topology which is order of 5. Packet size of 1024 bytes, and data rate of 1 Gbps with the warm up time and simulation time of 0.5ms and 0.5s, respectively. Five traffic patterns such as uniform, bit complement, neighbor, tornado and hotspot were considered by the authors. The proposed topology was compared with mesh, torus and x-torus topologies [68]. Simulation results are presented in tabular as well as via graphical layout. Table 5.3 to Table 5.7 presents Average End to End Delay under various traffic such as uniform, bit complement, neighbor, tornado and hotspot. Similarly, Table 5.8 to Table 5.12 presents Average Throughput and Table 5.13 to Table 5.17 shows Average Hop Count for the same traffic patterns. In addition, Figure 5.3 to Figure 4.17 represents the graph of Average End to End Delay, Average Throughput and Average Hop Count for the same traffic patterns, as mentioned. Furthermore, the corresponding results are discussed in section 5.4.

Table 5.2 Simulation Parameters

S.no.	Parameter Name	Value
1	Rows	5
2	Coloums	5
3	Packet size	1024 bytes
4	Data rate	1Gbps
5	Simulation time	0.5 s
6	Warm up time	0.5ms
7	Simulator	OMNeT++
8	Traffic Type	Uniform, Bit Complement, Neighbor, Tornado, Hotspot
9	Link Delay	0.1 ms
10	Routing Algorithm	Table based Shortest Path (Static)

Table 5.3 Average End to End Delay under Uniform Traffic

Interpacket Arrival Delay(us)	Mesh	Torus	X-Torus	HX-Torus
17.07	0.00239	0.00026	0.00021	0.00019
8.53	0.04343	0.01099	0.00136	0.00019
5.69	0.07309	0.02602	0.01646	0.00359
4.27	0.08986	0.05327	0.02579	0.01212
3.41	0.09687	0.08094	0.03678	0.01628
2.84	0.10134	0.09833	0.04901	0.02117
2.44	0.10392	0.10886	0.05994	0.02986
2.13	0.10559	0.11489	0.06929	0.04173
1.90	0.10740	0.11835	0.07682	0.05233
1.71	0.10951	0.12012	0.08408	0.06030

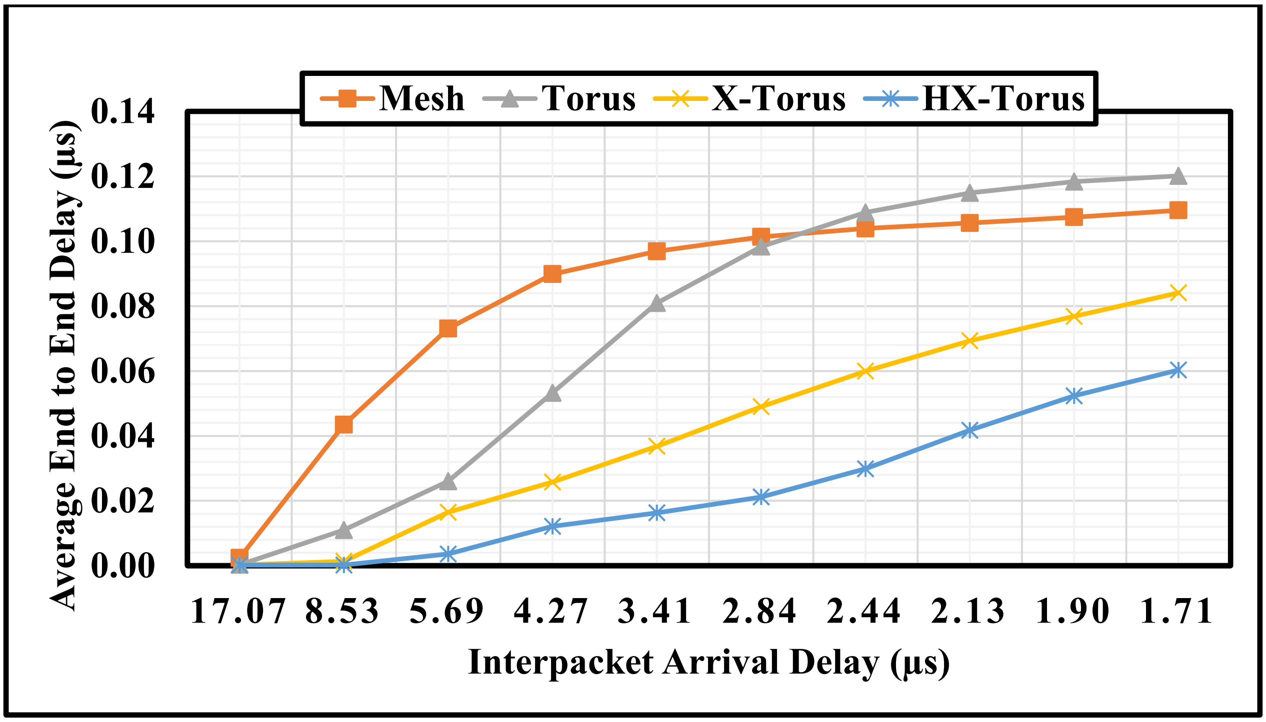

Figure 5.3Average End to End Delay under Uniform Traffic

Figure 5.3 shows the average end to end delay in the Mesh, Torus, X-Torus and HX-Torus based on data points (Interpacket Arrival Delay) provided in Table 5.3. The end to end delay of Torus and Mesh showed a steady but significant rise over the interpacket arrival delay, while in X-torus and HX-Torus the end to end delay rose gradually. At 8.53 the end to end delay of X-Torus and HX-Torus were 0.00136 µs and 0.00019 µs respectively. Torus's delay increased sharply throughout the time, exceeding Mesh topology delay and reaching almost 0.12012 µs at the end of the time. In the meantime, the average end to end delay of X-torus and HX-Torus grew to 0.01646 μs and 0.00359 μs at 5.69 respectively and then increased gradually until 1.71.

Table 5.4 Average End to End Delay under Bit Complement Traffic

Interpacket Arrival Delay(us)	Mesh	Torus	X-Torus	HX-Torus
17.07	0.07701	0.00026	0.00021	0.00018
8.53	0.15696	0.07684	0.03851	0.05764
5.69	0.18478	0.12905	0.10127	0.11512
4.27	0.20007	0.15683	0.13598	0.14637
3.41	0.20916	0.17350	0.15681	0.16512
2.84	0.21510	0.18462	0.17069	0.17762
2.44	0.21933	0.19256	0.18061	0.18655
2.13	0.22249	0.19853	0.18806	0.19325
1.90	0.22499	0.20317	0.19384	0.19847
1.71	0.22706	0.20688	0.19847	0.20263

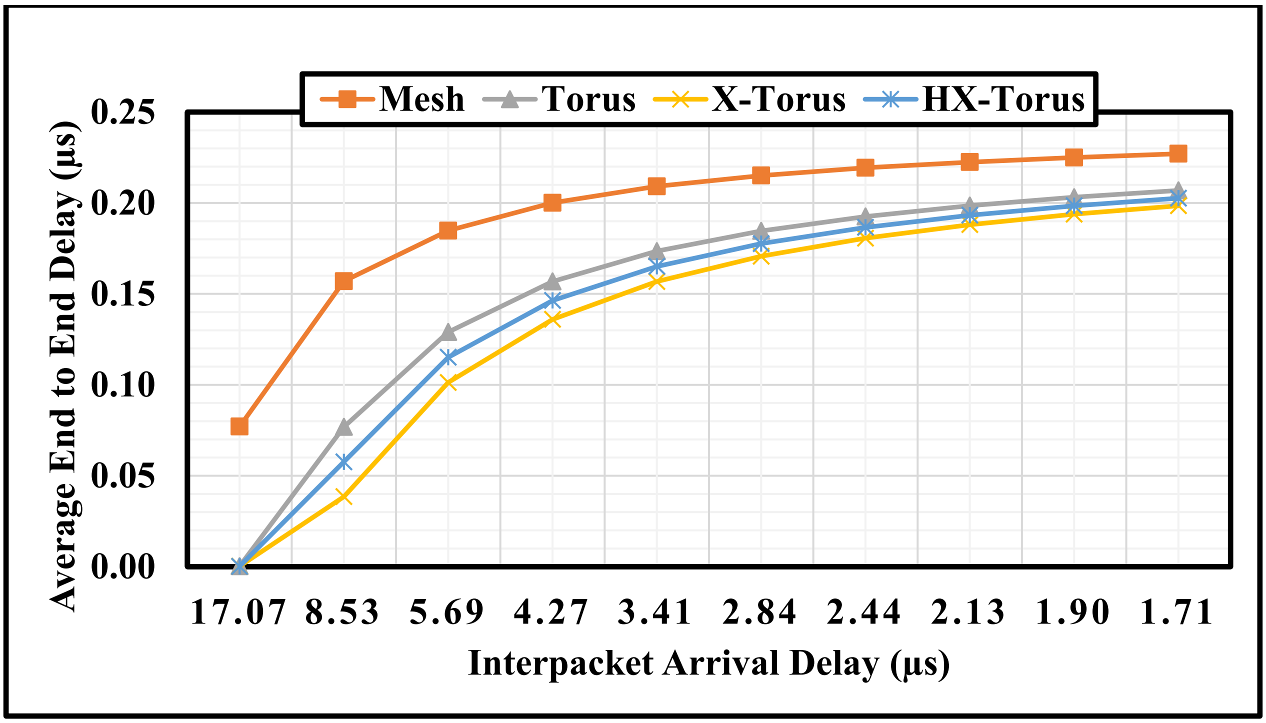

Figure 5.4 Average End to End Delay under Bit Complement Traffic

Figure 5.4 illustrates the average end to end delay in the Mesh, Torus, X-Torus and HX-Torus based on data points (Interpacket Arrival Delay) provided in Table 5.4. In starting the end to end delay of Mesh, Torus, X-Torus and HX-Torus rose significantly and then increased gently over the interpacket arrival delay. At 17.07 the end to end delay of Torus, X-Torus and HX-Torus were nearly 0.00026 μs, 0.00021 μs and 0.00018 μs respectively. Delay of HX-Torus is slightly higher than X-Torus. Mesh's delay increased sharply at 2.84, exceeding Torus topology's delay and reaching almost 0.22706 μs at the end of the time. In the meantime, the average end to end delay of X-Torus and HX-Torus gradually grew to 0.19847 μs and 0.20263 μs at 1.71.

Table 5.5 Average End to End Delay under Neighbor Traffic

Interpacket Arrival Delay(us)	Mesh	Torus	X-Torus	HX-Torus
17.07	0.00021	0.00013	0.00013	0.00013
8.53	0.00977	0.00971	0.00971	0.00971
5.69	0.08349	0.08344	0.08344	0.08344
4.27	0.12515	0.12509	0.12509	0.12509
3.41	0.15014	0.15009	0.15009	0.15009
2.84	0.16681	0.16675	0.16675	0.16675
2.44	0.17871	0.17865	0.17865	0.17865
2.13	0.18764	0.18758	0.18758	0.18758
1.90	0.19458	0.19453	0.19453	0.19453
1.71	0.20014	0.20008	0.20008	0.20008

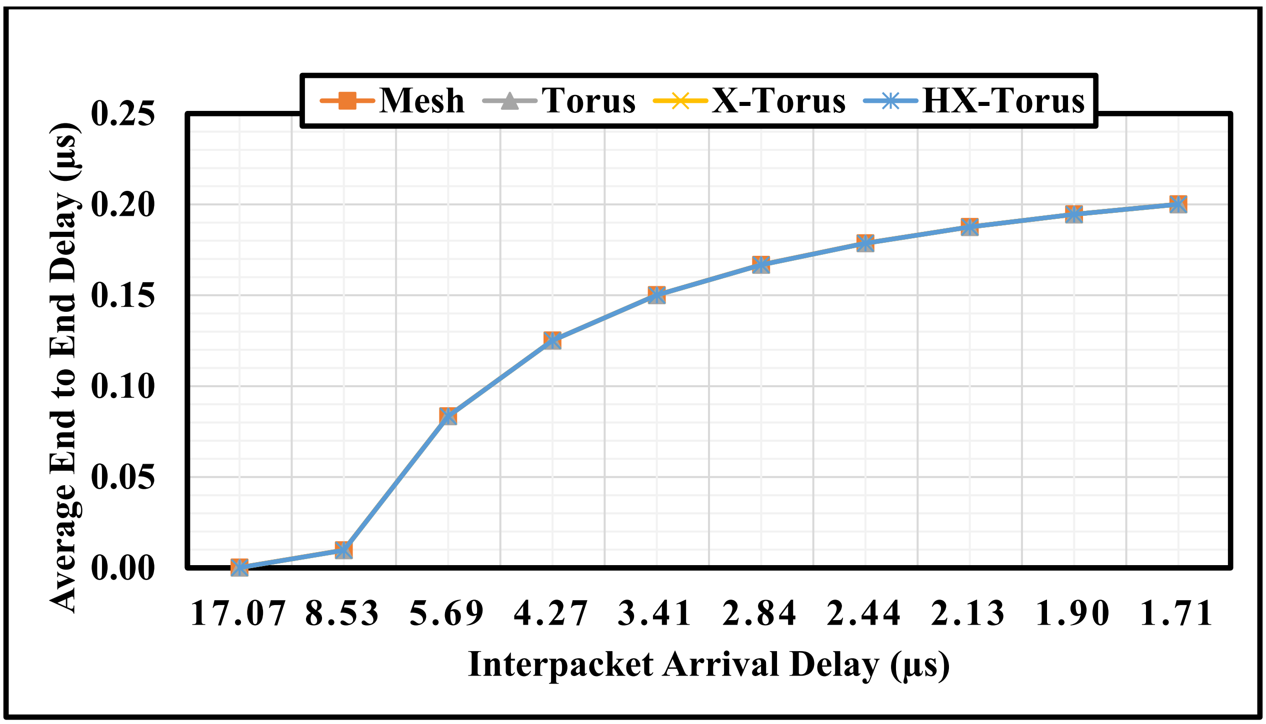

Figure 5.5 Average End to End Delay under Neighbor Traffic

Figure 5.5 gives information about the average end to end delay in the Mesh, Torus, X-Torus and HX-Torus based on data points (Interpacket Arrival Delay) provided in Table 5.5. In starting the end to end delay of Mesh, Torus, X-Torus and HX-Torus experienced a gradual rise and then increased dramatically over the interpacket arrival delay. At 17.07 the end to end delay of Torus, X-Torus and HX-Torus were 0.00013 µs. Delay of Torus, X-Torus and HX-Torus are almost same. Mesh's delay increased gently at 8.53, exceeding Torus, X-Torus, and HX-Torus topologies' delay and reaching almost 0.20014 µs at the end of the time. In the meantime, the average end to end delay of Torus, X-Torus and HX-Torus gradually grew to 0.20008 µs at 1.71.

Table 5.6 Average End to End Delay under Tornado Traffic

Interpacket Arrival Delay(us)	Mesh	Torus	X-Torus	HX-Torus
17.07	0.19439	0.22507	0.05845	0.02297
8.53	0.21023	0.23804	0.12732	0.07072
5.69	0.21678	0.24140	0.16533	0.12511
4.27	0.22135	0.24304	0.18432	0.15479
3.41	0.22482	0.24414	0.19553	0.17250
2.84	0.22755	0.24492	0.20304	0.18435
2.44	0.22975	0.24556	0.20848	0.19289
2.13	0.23157	0.24613	0.21266	0.19935
1.90	0.23310	0.24654	0.21599	0.20444
1.71	0.23440	0.24701	0.21872	0.20855

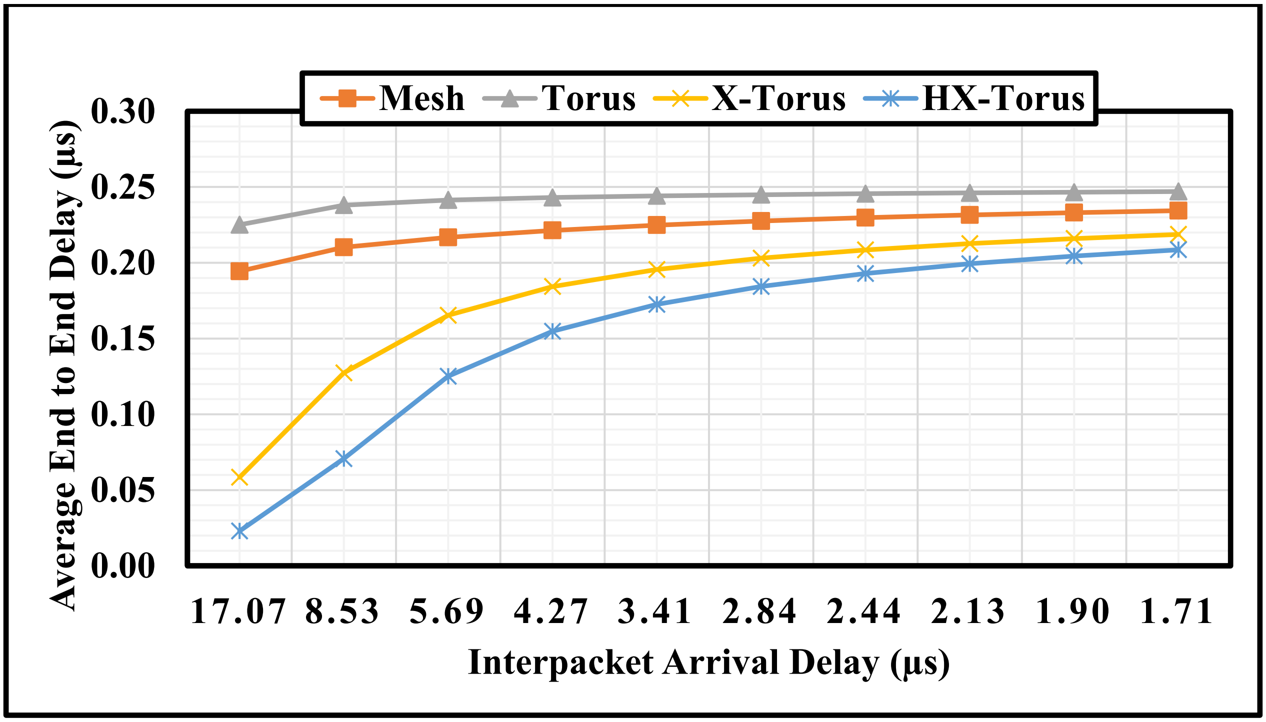

Figure 5.6 Average End to End Delay under Tornado Traffic

Figure 5.6 describes the average end to end delay in the Mesh, Torus, X-Torus and HX-Torus based on data points (Interpacket Arrival Delay) provided in Table 5.6. In starting the end to end delay of Mesh, Torus, X-Torus and HX-Torus showed a gradual increase and then remained almost steady over the interpacket arrival delay. At 5.69 the end to end delay of Mesh and Torus were 0.21678 µs and 0.24140 µs and then rose gently to 0.23440 µs and 0.24701 µs at the end of time respectively. In the meantime, the average end to end delay of X-Torus and HX-Torus increased considerably to 0.20304 µs and 0.18435 µs at 2.84 respectively and then rose gently to 0.21872 µs and 0.20855 µs at 1.71 respectively.

Table 5.7 Average End to End Delay under Hotspot Traffic

Interpacket Arrival Delay(us)	Mesh	Torus	X-Torus	HX-Torus
17.07	0.00814	0.00227	0.00021	0.00071
8.53	0.04593	0.01680	0.00442	0.00485
5.69	0.07254	0.02580	0.01427	0.00552
4.27	0.08862	0.05244	0.02351	0.01024
3.41	0.09557	0.07955	0.03556	0.01311
2.84	0.10089	0.09610	0.04894	0.01805
2.44	0.10441	0.10782	0.06059	0.02760
2.13	0.10645	0.11513	0.07038	0.03950
1.90	0.10825	0.11958	0.07814	0.04997
1.71	0.11052	0.12225	0.08517	0.05842

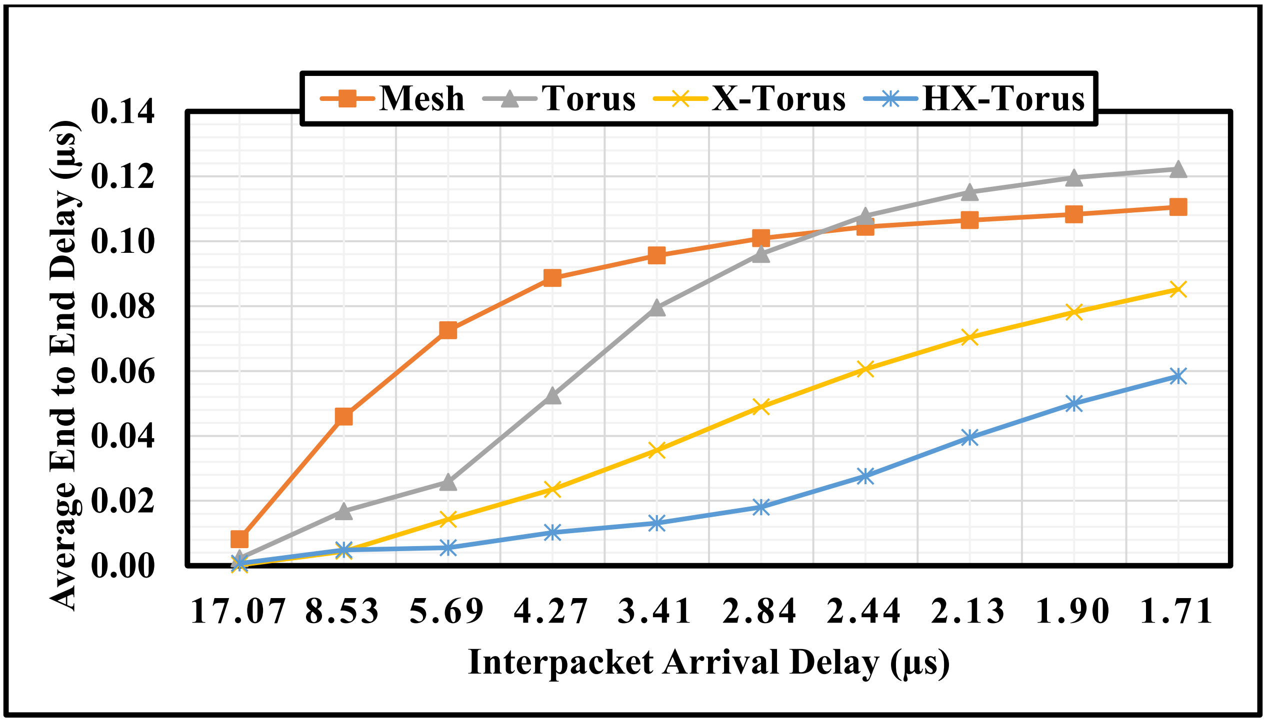

Figure 5.7 Average End to End Delay under Hotspot Traffic

Figure 5.7 indicates the average end to end delay in the Mesh, Torus, X-Torus and HX-Torus based on data points (Interpacket Arrival Delay) provided in Table 5.7. The end to end delay of Torus and Mesh increased significantly over the interpacket arrival delay, while in X-torus and HX-Torus the end to end delay rose gradually. At 8.53 the end to end delay of X-Torus and HX-Torus were 0.00442 µs and 0.00485 µs respectively. Torus's delay increased sharply throughout the time, exceeding Mesh topology delay and reaching almost 0.12225 µs at the end of the time. In the meantime, the average end to end delay of X-torus and HX-Torus grew to 0.02351 µs and 0.01024 µs at 4.27 respectively and then experienced a significant rise until 1.71.

Table 5.8 Average Throughput under Uniform Traffic

Interpacket Arrival Delay(us)	Mesh	Torus	X-Torus	HX-Torus
17.07	58061.52	58562.48	58567.83	58570.55
8.53	90463.99	111655.91	116571.67	117141.43
5.69	103380.32	149966.08	161123.82	172999.02
4.27	109051.59	169907.41	199149.69	219569.92
3.41	114654.80	173585.03	229235.51	264492.95
2.84	119390.38	174261.33	251530.40	304973.85
2.44	124002.47	174215.65	268674.31	336634.50
2.13	128169.50	174991.33	282345.84	358023.99
1.90	131656.36	176402.24	293940.15	375037.59
1.71	134041.22	178252.85	302420.27	390957.47

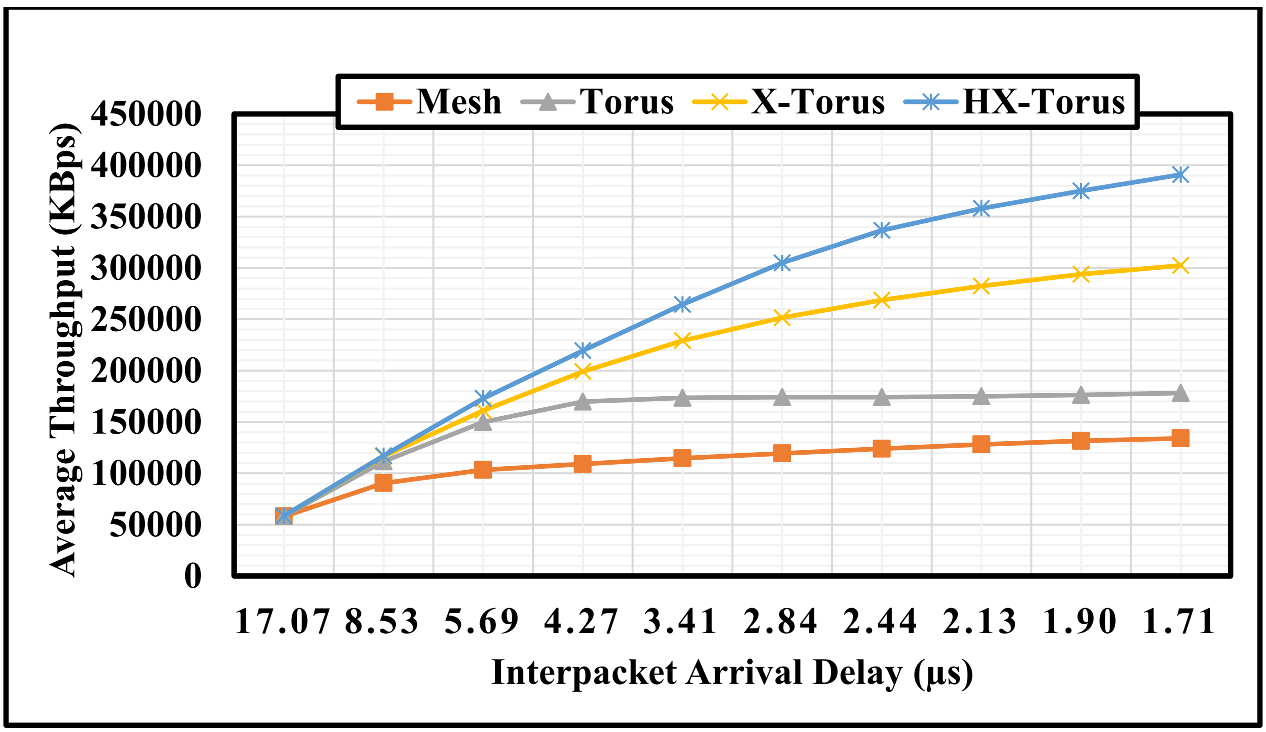

Figure 5.8 Average Throughput under Uniform Traffic

Figure 5.8 shows the average throughput in the Mesh, Torus, X-Torus and HX-Torus based on data points (Interpacket Arrival Delay) provided in Table 5.8. The average throughput of X-Torus and

HX-Torus showed a steady but significant rise over the interpacket arrival delay, while in Mesh and Torus the average throughput rose gradually. At 8.53 the average throughput of Torus, X-Torus and HX-Torus were 111655.91 KBps, 116571.67 KBps and 117141.43 KBps respectively. HX-Torus' throughput increased sharply throughout the time, exceeding X-Torus, Torus and Mesh topologies throughput and reaching almost 390957.47 KBps at the end of the period. In the meantime, the average throughput of Torus and Mesh grew to 169907.41 KBps and 109051.59 KBps at 4.27 respectively and then remained steady until 1.71.

Table 5.9 Average Throughput under Bit Complement Traffic

Interpacket Arrival Delay(us)	Mesh	Torus	X-Torus	HX-Torus
17.07	40592.63	58562.07	58568.31	58571.67
8.53	43720.08	81217.67	99178.61	90205.36
5.69	46063.72	85123.30	104646.45	94892.41
4.27	47182.17	87466.04	106990.20	97235.68
3.41	48462.01	89809.23	109333.94	99579.42
2.84	49977.18	92153.01	111677.81	101923.28
2.44	51658.28	94496.26	114021.34	104266.64
2.13	53459.83	96839.91	116365.16	106610.46
1.90	55352.38	99183.41	118709.00	108954.29
1.71	57314.37	101526.67	121052.42	111297.55

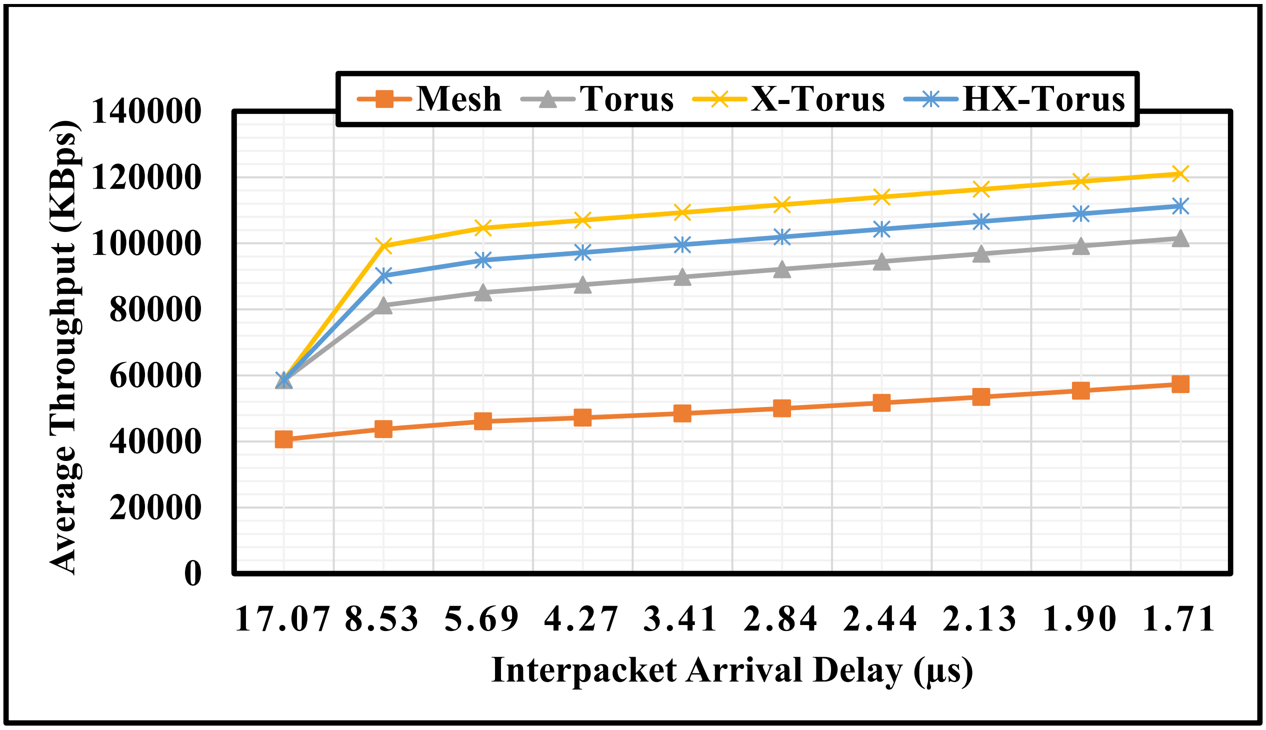

Figure 5.9 Average Throughput under Bit Complement Traffic

Figure 5.9 illustrates the average throughput in the Mesh, Torus, X-Torus and HX-Torus based on data points (Interpacket Arrival Delay) provided in Table 5.9. In starting, the average throughput of Torus, X-Torus and HX-Torus experienced a significant trend over the interpacket arrival delay and then increased, while in Mesh the average throughput rose gradually. At 8.53 the average throughput of X-Torus and HX-Torus were nearly 99178.61 KBps and 90205.36 KBps respectively. Throughput of HX-Torus topology is less than X-Torus' throughput over the time. However, HX-Torus' throughput increased sharply throughout the time, exceeding Torus and Mesh topologies throughput and reaching almost 111297.55 KBps at the end of the period. In the meantime, the average throughput of Torus and Mesh grew to 81217.67 KBps and 43720.08 KBps at 8.53 respectively and then rose gradually until 1.71.

Table 5.10 Average Throughput under Neighbor Traffic

Interpacket Arrival Delay(us)	Mesh	Torus	X-Torus	HX-Torus
17.07	58568.31	58577.43	58577.43	58577.43
8.53	112654.63	112666.64	112666.64	112666.64
5.69	117145.45	117158.00	117158.00	117158.00
4.27	117145.31	117157.92	117157.92	117157.92
3.41	117145.43	117157.91	117157.91	117157.91
2.84	117145.48	117158.03	117158.03	117158.03
2.44	117145.25	117158.05	117158.05	117158.05
2.13	117145.32	117158.12	117158.12	117158.12
1.90	117145.40	117158.28	117158.28	117158.28
1.71	117145.39	117158.19	117158.19	117158.19

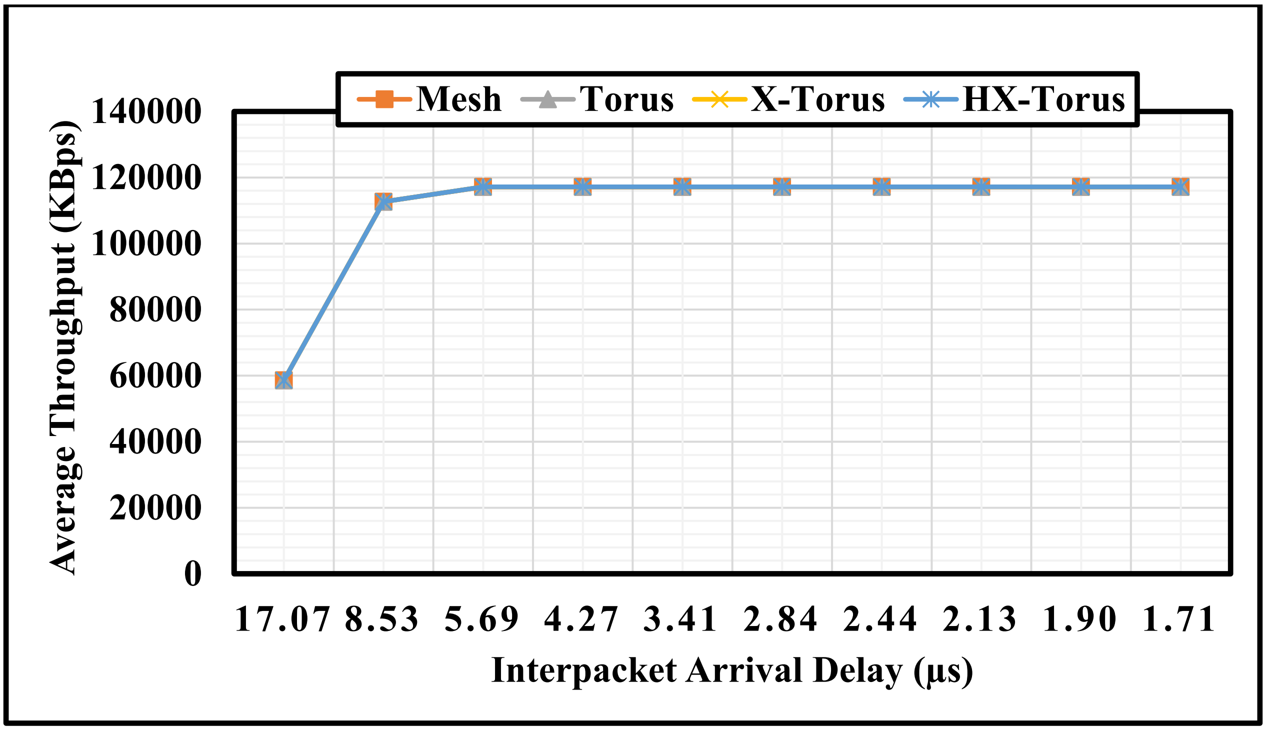

Figure 5.10 Average Throughput under Neighbor Traffic

Figure 5.10 gives information about the average throughput in the Mesh, Torus, X-Torus and HX-Torus based on data points (Interpacket Arrival Delay) provided in Table 5.10. In starting, the average throughput of Mesh, Torus, X-Torus and HX-Torus showed a significant rise over the interpacket arrival delay and then remained same. At 17.07 the average throughput of Torus, X-Torus and HX-Torus were 58577.43 KBps and then reaching 117158.19 KBps at 1.71. In the meantime, the average throughput of Mesh grew to 112654.63 KBps at 8.53 and then rose gradually to 117145.45 KBps at 5.69. However, it remained the same at end of the time.

Table 5.11 Average Throughput under Tornado Traffic

Interpacket Arrival Delay(us)	Mesh	Torus	X-Torus	HX-Torus
17.07	58638.35	29333.92	202842.77	258555.16
8.53	57048.39	25691.91	209932.06	344709.58
5.69	55897.85	23245.90	204420.76	342817.39
4.27	55028.57	21235.91	199964.78	336011.82
3.41	54348.57	19553.92	196649.58	330960.62
2.84	53802.63	18121.94	194060.99	327036.58
2.44	53354.28	16891.96	191974.76	323878.45
2.13	52979.91	15823.97	190252.08	321277.05
1.90	52661.98	14880.00	188805.14	319093.90
1.71	52389.58	17054.36	187565.91	317237.45

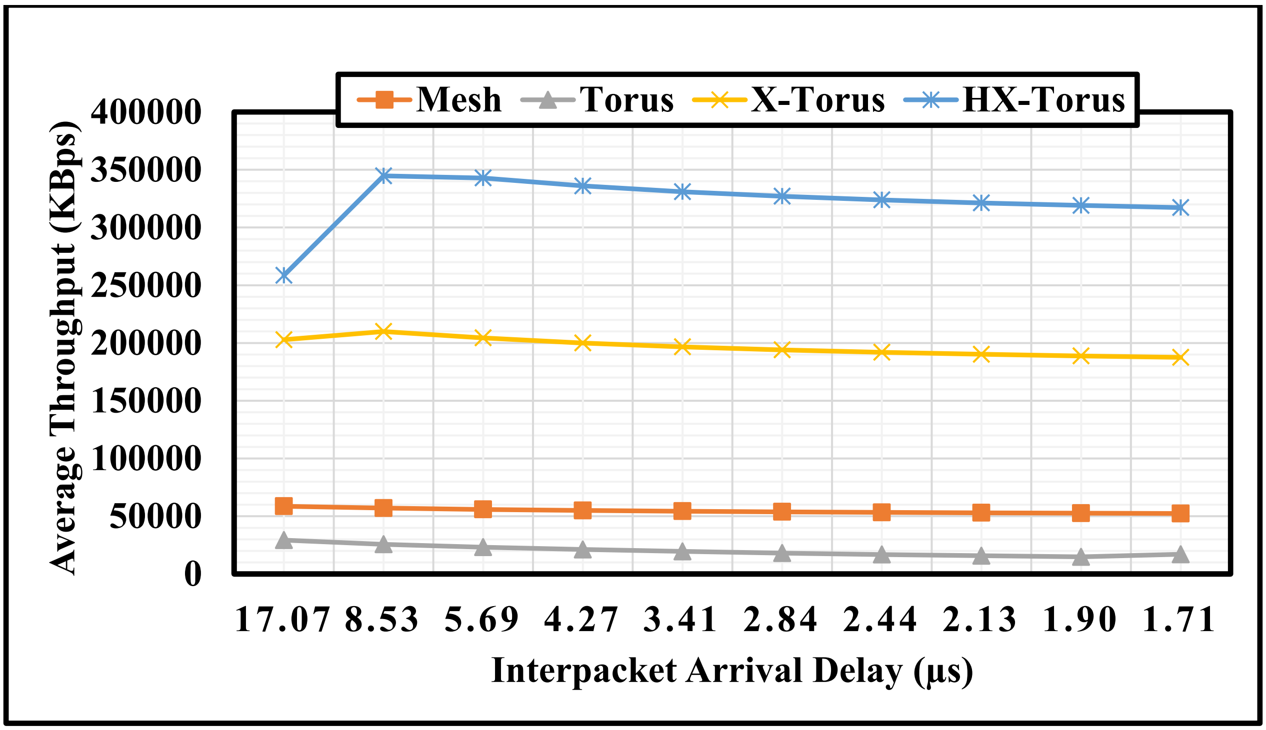

Figure 5.11 Average Throughput under Tornado Traffic

Figure 5.11 indicates the average throughput in the Mesh, Torus, X-Torus and HX-Torus based on data points (Interpacket Arrival Delay) mentioned in Table 5.11. In the beginning, the average throughput of X-Torus and HX-Torus showed a significant rise and then experienced a gradual decrease over the interpacket arrival delay, while in Mesh and Torus the average throughput fell gradually. At 8.53 the average throughput of X-Torus and HX-Torus were 209932.06 KBps, and 344709.58 KBps respectively. HX-Torus' throughput decreased gently throughout the time, exceeding X-Torus, Torus and Mesh topologies throughput and reaching almost 317237.45 KBps at the end of the period. In the meantime, the average throughput of Mesh, Torus and X-Torus declined gradually to 55897.85 KBps, 23245.90 KBps and 204420.76 KBps at 5.69 respectively and then remained almost same until 1.71.

Table 5.12 Average Throughput under Hotspot Traffic

Interpacket Arrival Delay(us)	Mesh	Torus	X-Torus	HX-Torus
17.07	55079.43	57406.63	58567.24	58203.75
8.53	84914.95	103012.32	113028.23	110611.52
5.69	99081.62	140841.36	157725.60	162469.60
4.27	105515.98	159733.65	194509.62	209381.21
3.41	111398.23	163831.59	222351.27	254283.99
2.84	115827.19	166145.93	242313.42	294113.79
2.44	120024.39	166655.99	257657.10	324430.31
2.13	124134.58	167330.93	270167.08	345988.18
1.90	127625.49	168573.44	281113.36	363807.91
1.71	129960.14	170252.29	289936.28	379741.27

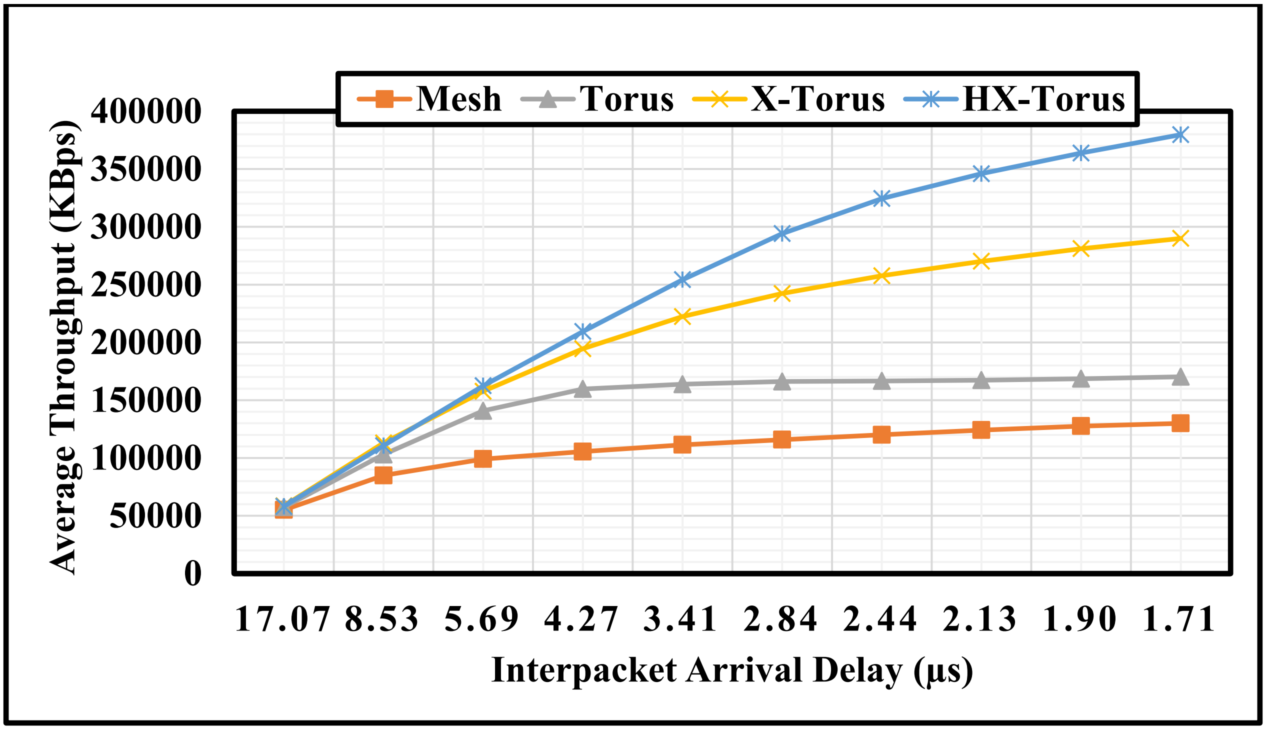

Figure 5.12 Average Throughput under Hotspot Traffic

Figure 5.12 describes the average throughput in the Mesh, Torus, X-Torus and HX-Torus based on data points (Interpacket Arrival Delay) provided in Table 5.12. The average throughput of X-Torus and HX-Torus showed a steady but significant rise over the interpacket arrival delay, while in Mesh and Torus the average throughput rose gradually. At 4.27 the average throughput of Torus, X-Torus and HX-Torus were 159733.65 KBps, 194509.62 KBps and 209381.21 KBps respectively. HX-Torus' throughput rose sharply throughout the time, exceeding X-Torus, Torus and Mesh topologies throughput and reaching almost 379741.27 KBps at the end of the period. In the meantime, the average throughput of Torus and Mesh increased until 3.41 and then remained same till 1.71.

Table 5.13 Average Hop Count under Uniform Traffic

Interpacket Arrival Delay(us)	Mesh	Torus	X-Torus	HX-Torus
17.07	3.18904	2.40158	1.95245	1.73096
8.53	2.86184	2.36803	1.94882	1.73109
5.69	2.54338	2.28315	1.89571	1.72102
4.27	2.29060	2.18022	1.84574	1.69224
3.41	2.10101	2.05743	1.79810	1.67056
2.84	1.94998	1.94852	1.75056	1.64945
2.44	1.82617	1.85090	1.70613	1.62638
2.13	1.72065	1.76578	1.66496	1.59997
1.90	1.63042	1.69134	1.62696	1.57374
1.71	1.54987	1.62522	1.59081	1.54931

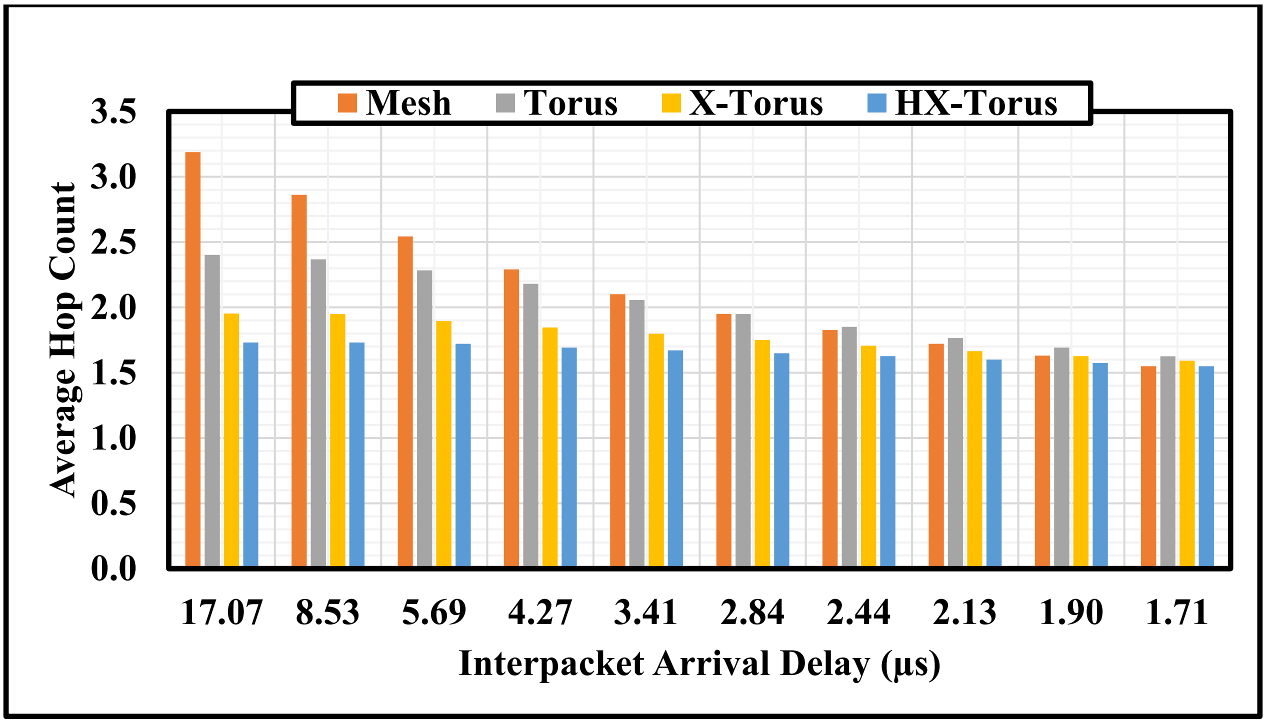

Figure 5.13 Average Hop Count under Uniform Traffic

Figure 5.13 shows the average hop count in the Mesh, Torus, X-Torus and HX-Torus based on data points (Interpacket Arrival Delay) provided in Table 5.13. Overall, Mesh, Torus, X-Torus and HX-Torus experienced a downwards trend throughout the time. HX-Torus hop count was 1.73096 at 17.07 being lower than Mesh, Torus and X-Torus topologies and continued to decline steadily to the end of the time, reaching around 1.67056 at 3.41 and hitting low point of 1.54931 at 1.71 interpacket arrival delay. Same pattern is also observed for Mesh, Torus and X-Torus topologies.

Table 5.14 Average Hop Count under Bit Complement Traffic

Interpacket Arrival Delay(us)	Mesh	Torus	X-Torus	HX-Torus
17.07	4.80	2.40	1.92	1.68
8.53	4.80	2.40	1.92	1.68
5.69	4.80	2.40	1.92	1.68
4.27	4.80	2.40	1.92	1.68
3.41	4.80	2.40	1.92	1.68
2.84	4.80	2.40	1.92	1.68
2.44	4.80	2.40	1.92	1.68
2.13	4.80	2.40	1.92	1.68
1.90	4.80	2.40	1.92	1.68
1.71	4.80	2.40	1.92	1.68

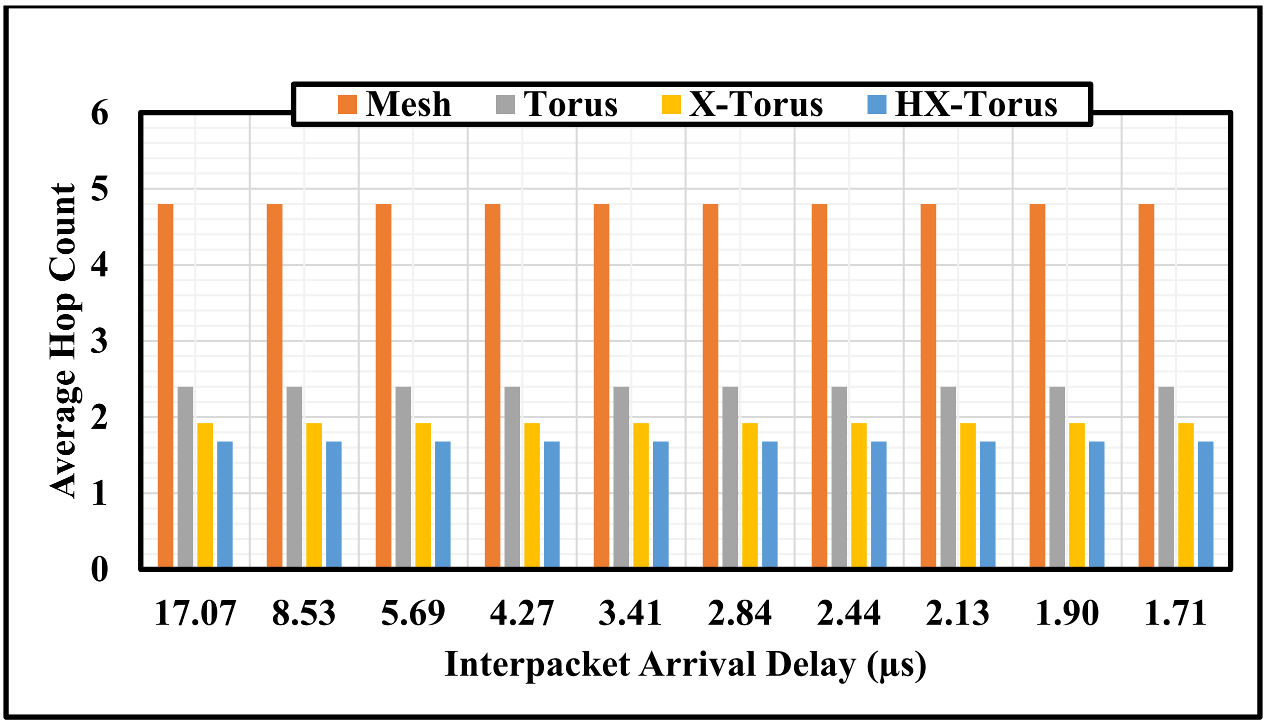

Figure 5.14 Average Hop Count under Bit Complement Traffic

Figure 5.14 describes the average hop count in the Mesh, Torus, X-Torus and HX-Torus based on data points (Interpacket Arrival Delay) provided in Table 5.14. In general, Mesh, Torus, X-Torus and HX-Torus experienced a constant trend throughout the time. The average hop count of Mesh, Torus, X-Torus, and HX-Torus was 4.80, 2.40, 1.92, and 1.80 respectively. Hop count of HX-

Torus topology was smaller than other mentioned topologies and found to be 1.84 throughout the time.

Table 5.15 Average Hop Count under Neighbor Traffic

Interpacket Arrival Delay(us)	Mesh	Torus	X-Torus	HX-Torus
17.07	1.92	1.2	1.2	1.2
8.53	1.92	1.2	1.2	1.2
5.69	1.92	1.2	1.2	1.2
4.27	1.92	1.2	1.2	1.2
3.41	1.92	1.2	1.2	1.2
2.84	1.92	1.2	1.2	1.2
2.44	1.92	1.2	1.2	1.2
2.13	1.92	1.2	1.2	1.2
1.90	1.92	1.2	1.2	1.2
1.71	1.92	1.2	1.2	1.2

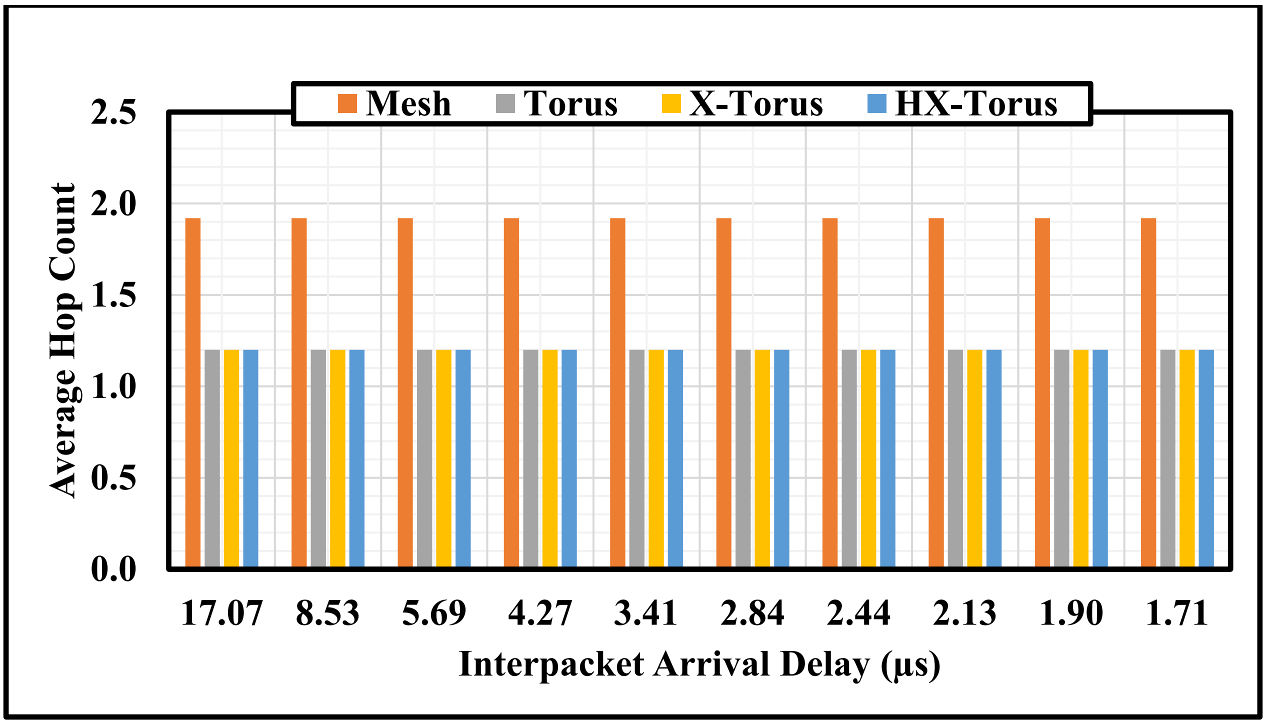

Figure 5.15 Average Hop Count under Neighbor Traffic

Figure 5.15 provides the information about the average hop count in the Mesh, Torus, X-Torus and HX-Torus based on data points (Interpacket Arrival Delay) provided in Table 5.15. In general, Mesh, Torus, X-Torus and HX-Torus experienced a constant trend throughout the time. HX-Torus, X-Torus and Torus' hop count was 1.2. Hop count of Mesh topology was higher than other mentioned topologies and found to be 1.92 throughout the time.

Table 5.16 Average Hop Count under Tornado Traffic

Interpacket Arrival Delay(us)	Mesh	Torus	X-Torus	HX-Torus
17.07	3.98713	3.19875	2.18578	1.85898
8.53	3.36801	3.01152	2.15519	1.83307
5.69	3.07285	2.81700	2.09531	1.80017
4.27	2.91139	2.68167	2.04872	1.77528
3.41	2.81101	2.58443	2.01359	1.75663
2.84	2.74287	2.51120	1.98604	1.74205
2.44	2.69375	2.45418	1.96382	1.73031
2.13	2.65661	2.40799	1.94549	1.72064
1.90	2.62774	2.37030	1.93011	1.71251
1.71	2.60471	2.33879	1.91700	1.70560

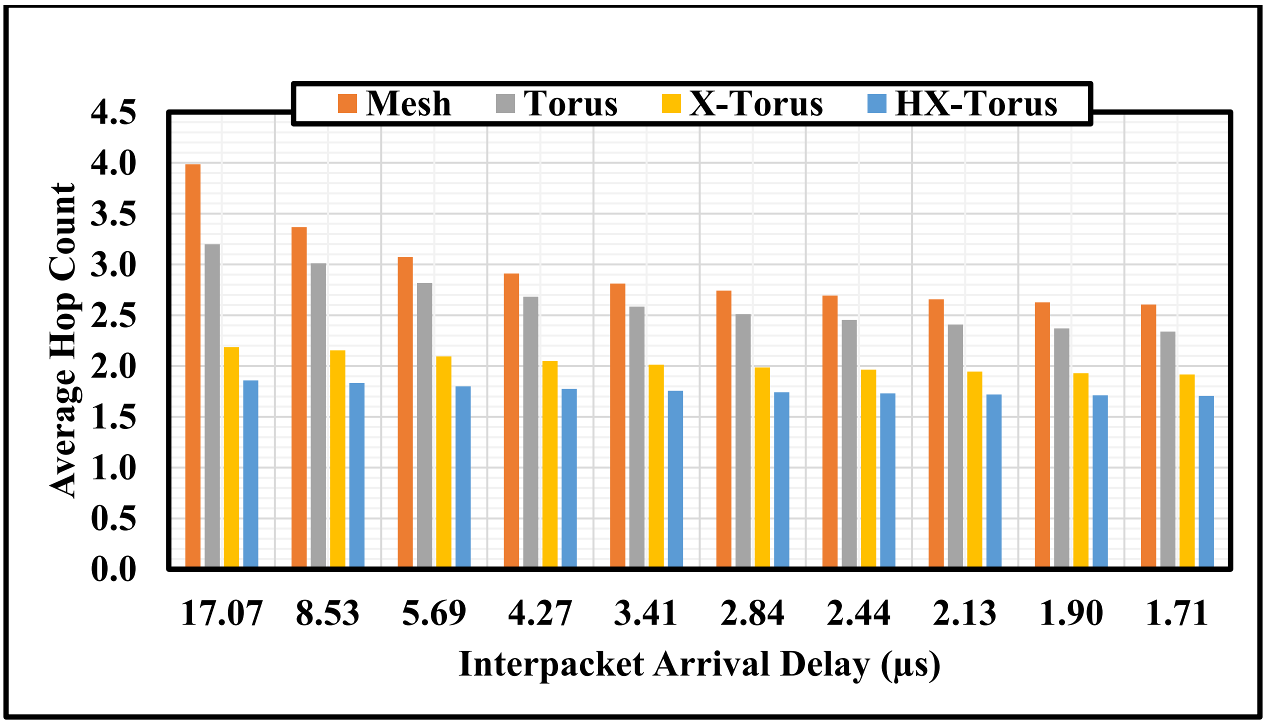

Figure 5.16 Average Hop Count under Tornado Traffic

Figure 5.16 shows the average hop count in the Mesh, Torus, X-Torus and HX-Torus based on data points (Interpacket Arrival Delay) provided in Table 5.16. Overall, Mesh, Torus, X-Torus and HX-Torus experienced a downwards trend throughout the time. HX-Torus' hop count was 1.85898 at 17.07 being lower than Mesh, Torus and X-Torus topologies and continued to decline steadily to the end of the time, reaching around 1.78802 at 3.41 and hitting low point of 1.70560 at 1.71 interpacket arrival delay. Same pattern is also observed for Mesh, Torus and X-Torus topologies.

Table 5.17 Average Hop Count under Hotspot Traffic

Interpacket Arrival Delay(us)	Mesh	Torus	X-Torus	HX-Torus
17.07	3.16822	2.44535	1.96770	1.72101
8.53	2.86186	2.37409	1.95310	1.71061
5.69	2.56708	2.30082	1.91753	1.70439
4.27	2.32875	2.19983	1.87579	1.69009
3.41	2.14647	2.08187	1.83190	1.67770
2.84	1.99732	1.97991	1.78758	1.66182
2.44	1.87563	1.88959	1.74672	1.64210
2.13	1.77394	1.80953	1.70936	1.61977
1.90	1.68587	1.73884	1.67515	1.59758
1.71	1.60838	1.67602	1.64301	1.57664

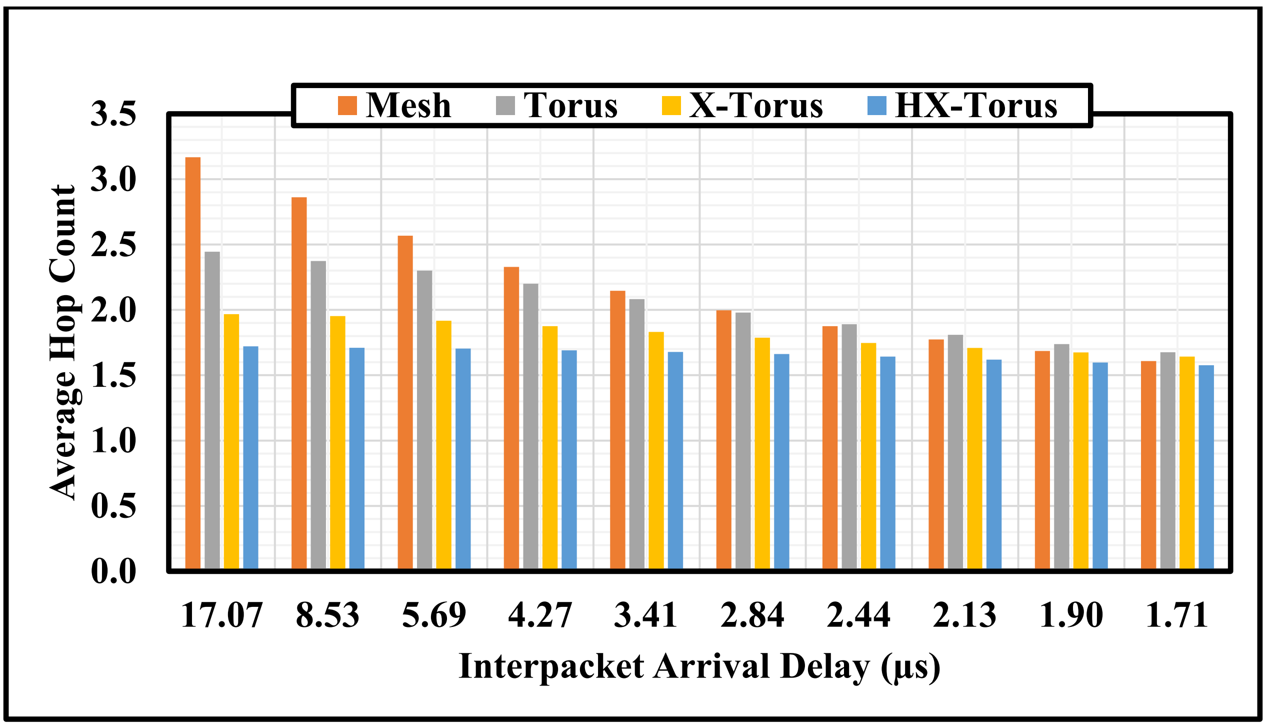

Figure 5.17 Average Hop Count under Hotspot Traffic

Figure 5.17 gives information about the average hop count in the Mesh, Torus, X-Torus and HX-Torus based on data points (Interpacket Arrival Delay) provided in Table 5.17. Overall, Mesh, Torus, X-Torus and HX-Torus experienced a downwards trend throughout the time. HX-Torus hop count was 1.72101 at 17.07 being lower than Mesh, Torus and X-Torus topologies and continued to decline steadily to the end of the time, reaching around 1.67770 at 3.41 and hitting low point of 1.57664 at 1.71 interpacket arrival delay. Same pattern is also observed for Mesh, Torus and X-Torus topologies.

5.4 Results and Discussions

This chapter presented an improved version of X-Torus topology, called as HX-Torus. It is well suitable for parallel processing system. Based on results obtained (section 5.3), the proposed topology revealed better performance in comparison to Mesh, Torus, and X-torus topologies. Maximum reduction in end to end delay obtained was 86.01% under uniform traffic. There was a major improvement in average throughput which was 40.88% under tornado traffic while average hop count was reduced by 14.95% under the influence of tornado traffic. The proposed topology attributed better path diversity to handle fault tolerance of network in an efficient manner. This topology inherits the properties of Torus as well as X-Torus topologies. In terms of degree, diameter, path diversity, average distance and bisectional bandwidth the proposed topology provide better performance. Due to its enhanced feature and performance, it is proposed that this approach could be a better option for the large scale of parallel computing system. In addition, this topology could be further explored to get high performance in various quality of service parameter because of efficient inter processor communication.

CHAPTER 6
A MODIFIED DIAGONAL TORUS TOPOLOGY FOR INTERCONNECTION NETWORKS

6.1 Introduction and Motivation

In recent time, parallel computers have done revolutionary work in the field of scientific calculation to obtain the optimal performance of the system [2, 69]. Based on the published literature, it was noted that improvement in interconnection networks enhances the parallel computing to a better standard. However, having an improved topology containing better node degree, scalability level, diameter and average distance is more beneficial. The biggest issue in parallel computing is communication delay overhead over the nodes of network due to exchange of data packets [70]. This results in the reduction of the performance of parallel computing system. Therefore, to overcome this problem there is a need to design a topology with better distribution of nodes within the network. Per the empirical information published within the literature, this is well known that the parallel computing system resolve the slow processing problem of uniprocessor [71]. Therefore, the study was aimed at authentic computation rather than communication delay, which could help to achieve higher throughput and minimal delay. In addition, network performance also depends on topological properties such as node degree, diameter, average distance and bisection width [72]. In order to gain optimal performance of network diameter and average distance must be smaller [73]. In addition, network bisectional width must be higher. From the published data, it is revealed that path diversity is a key feature to obtain fault tolerance in the system [74, 75]. Several researchers attributed that various topologies like mesh, torus, d-mesh, and d-torus are used to improve the network performance [20, 42]. This chapter covers the proposed topology Modified Diagonal Torus discussed in section 6.2 and its performance is tested using parameters which has been provided in section 6.3. The results of the study have been discussed in section 6.4.

6.2 Modified Diagonal Torus topology(MD-Torus)

This section presents the proposed topology which is the enhanced version of Diagonal Torus topology [76]. This topology is described as given below:

Let's assume any node X(i,j) is connected to any other node Y(p,q) where 0≤ i, j, p, q <n if it is satisfied by one of the eight equations.

1: $|p-i|+|q-j|=1$ (6.1)

2: $i=0, j=q, p=n-1$ (6.2)

3: $i=p, j=0, q=n-1$ (6.3)

4: $|p-i|+|q-j|=2$ (6.4)

5: $2k \leq |p-i|+|q-j| < 2n \; where \; k=2$ (6.5)

5.1: $i=q, j=p$ (6.6)

5.2.1: $p=i+q, p=n-1$ (6.7)

5.2.2: $q=j+p, q=n-1$ (6.8)

Figure 6.1 represents MD-Torus with node id and this formula is used in coordinate representation of node. This means node id 0 is equivalent to node (0,0) and node id 1 to node (0,1) and finally node id 24 with node (4,4) respectively. As presented, the green links are added to improve the system performance. 5x5 topology was considered, with 'n' as 5. This helped to reduce the hop count and the system performance was increased to a good level. Topological properties of MD-Torus were compared with Mesh, Torus, D-Mesh, and D-Torus in terms of number of nodes, diameter, bisection width, number of links, degree of various nodes, and path diversity. This comparison is described in Table 6.1.

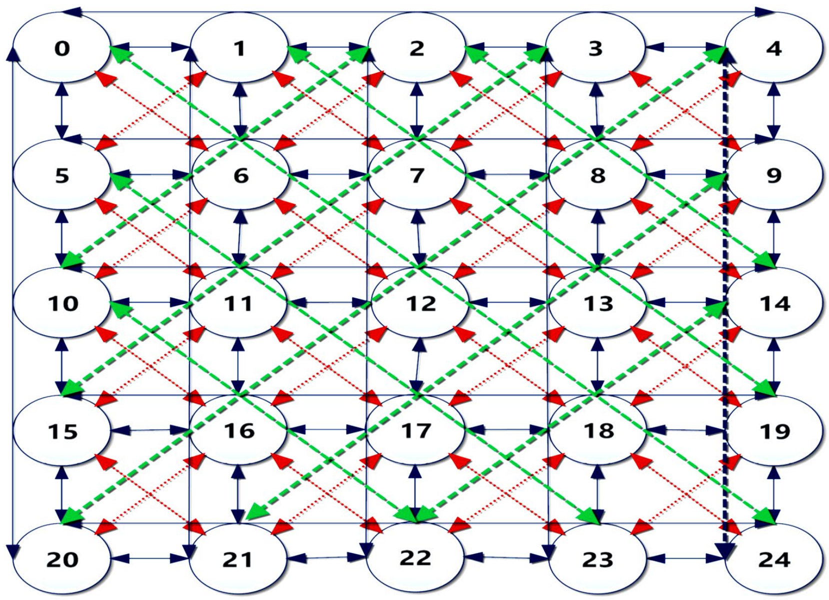

Figure 6.1 Modified Diagonal Torus Topology

Table 6.1 Topological Properties

Characteristics	**Mesh**	**Torus**	**D-Mesh**	**D-Torus**	**MD-Torus**
Number of Nodes	n^2	n^2	n^2	n^2	n^2
Diameter	2n-2	n-1	n-1	n-1	n-2
Bisection Width	n	2n	3n-2	4n-2	6n-3
Number of links	$2n^2-2n$	$2n^2$	$4n^2-6n+2$	$4n^2-4n+2$	$4n^2-2n+2$
Degree of Nodes	2, 3, 4	4	3, 5, 8	5, 6, 8	6, 7, 8
Path Diversity	yes	yes	yes	yes	yes

6.3 Testbed for Testing Modified Diagonal Torus topology

In this section, authors analyzed the performance parameters of proposed topology. These parameters are average throughput, average end to end delay and average hop count. Windows 10 operating system of 32-bit, equipped intel® Core™ i3 CPU M330@2.13 GHZ with 4.00 GB and 2.99GB usable was used for the study. Another key component which was used is OMNET++ simulator a component-based C++ simulation library and framework (extensible and modular) and is primarily used for building network simulator based on the Eclipse IDE [58]. All the key parameters for simulation purpose are shown in Table 6.2. This table describes the dimension of topology i.e. order of 5. Packet size was 1024 bytes, and data rate was 1 Gbps, with warm up time and simulation time of 0.5ms and 0.5s, respectively. Authors considered 5 traffic patterns such as uniform, bit complement, neighbor, tornado and hotspot. The proposed topology was compared with mesh, torus, d-mesh and d-torus topologies. Simulation results are tabulated and correspondingly represented graphically. Table 6.3 to Table 6.7 presents Average End to End Delay under various traffic such as uniform, bit complement, neighbor, tornado and hotspot, respectively [54]. Similarly, Table 6.8 to Table 6.12 presents Average Throughput and Table 6.13 to Table 6.17 shows Average Hop Count for the same traffic patterns. In addition, Figure 6.2 to Figure 6.16 represents the graph of Average End to End Delay, Average Throughput and Average Hop Count for the same traffic patterns which are mentioned. Results of the study are discussed in section 6.4.

Table 6.2 Simulation Parameters

S.no.	Parameter Name	Value
1	Rows	5
2	Coloums	5
3	Packet size	1024 bytes
4	Data rate	1Gbps
5	Simulation time	0.5 s
6	Warm up time	0.5ms
7	Simulator	OMNeT++
8	Traffic Type	Uniform, Bit Complement, Neighbor, Tornado, Hotspot
9	Link Delay	0.1 ms
10	Routing Algorithm	Table based Shortest Path (Static)

Table 6.3Average End to End Delay under Uniform Traffic

Interpacket Arrival Delay(us)	Mesh	Torus	D-Mesh	D-Torus	MD-Torus
17.07	0.00239	0.00026	0.00025	0.00020	0.00018
8.53	0.04343	0.01099	0.00470	0.00020	0.00018
5.69	0.07309	0.02602	0.02843	0.00334	0.00019
4.27	0.08986	0.05327	0.05211	0.01013	0.00228
3.41	0.09687	0.08094	0.06368	0.02146	0.00562
2.84	0.10134	0.09833	0.07144	0.03400	0.01203
2.44	0.10392	0.10886	0.07608	0.04758	0.02146
2.13	0.10559	0.11489	0.07844	0.05957	0.03300
1.90	0.10740	0.11835	0.07936	0.06911	0.04473
1.71	0.10951	0.12012	0.07942	0.07657	0.05540

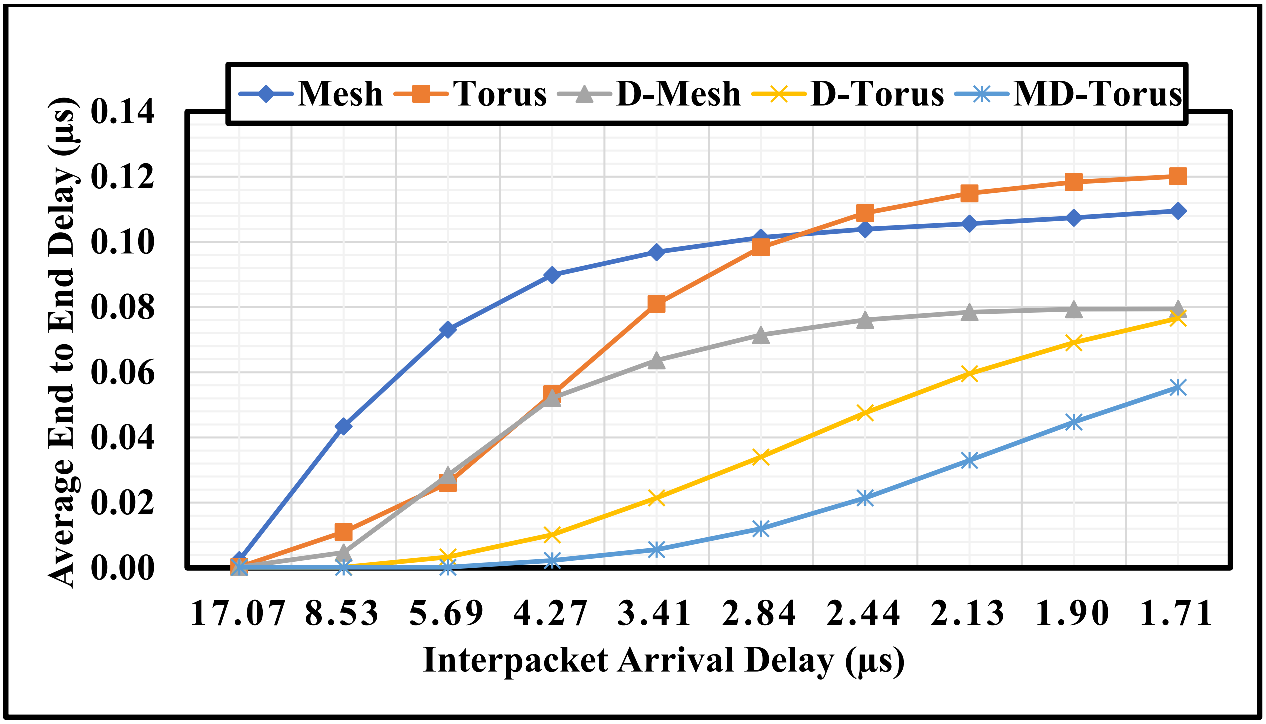

Figure 6.2 Average End to End Delay under Uniform Traffic

Figure 6.2 shows the average end to end delay in the Mesh, Torus, D-Mesh, D-Torus, and MD-Torus based on data points (Interpacket Arrival Delay) provided in Table 6.3. The end to end delay of Torus, Mesh and D-Mesh showed a steady but significant rise over the interpacket arrival delay, while in D-Torus and MD-Torus the end to end delay rose gradually. At 8.53 the end to end delay of D-Torus and MD-Torus were 0.00020 µs and 0.00018 µs respectively. Torus's delay increased sharply throughout the time, exceeding Mesh topology delay and reaching almost 0.12012 µs at the end of the time. In the meantime, the average end to end delay of D-Mesh, D-Torus and MD-Torus grew to 0.02843 µs, 0.00334 µs and 0.00019 µs at 5.69 respectively and then increased gradually until 1.71.

Table 6.4 Average End to End Delay under Bit Complement Traffic

Interpacket Arrival Delay(us)	Mesh	Torus	D-Mesh	D-Torus	MD-Torus
17.07	0.07701	0.00026	0.03864	0.00023	0.00017
8.53	0.15696	0.07684	0.13704	0.07682	0.05763
5.69	0.18478	0.12905	0.17428	0.12903	0.11512
4.27	0.20007	0.15683	0.19216	0.15681	0.14636
3.41	0.20916	0.17350	0.20255	0.17348	0.16511
2.84	0.21510	0.18462	0.20933	0.18460	0.17762
2.44	0.21933	0.19256	0.21409	0.19254	0.18655
2.13	0.22249	0.19853	0.21762	0.19850	0.19325
1.90	0.22499	0.20317	0.22035	0.20314	0.19846
1.71	0.22706	0.20688	0.22252	0.20685	0.20263

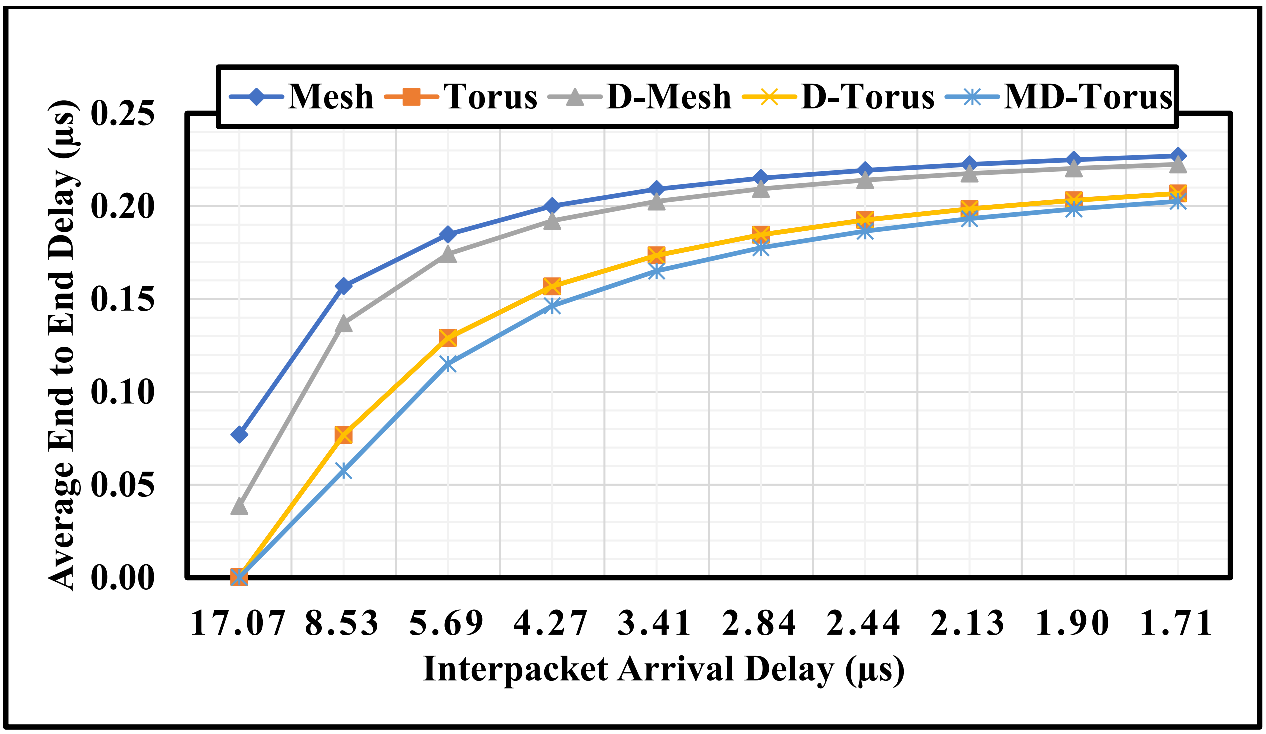

Figure 6.3Average End to End Delay under Bit Complement Traffic

Figure 6.3 illustrates the average end to end delay in the Mesh, Torus, D-Mesh, D-Torus, and MD-Torus based on data points (Interpacket Arrival Delay) provided in Table 6.4. In starting the end to end delay of Mesh, Torus, D-Mesh, D-Torus and MD-Torus rose significantly and then increased gently over the interpacket arrival delay. At 17.07 the end to end delay of D-Mesh, D-Torus, and MD-Torus were nearly 0.03864 μs, 0.00023 μs and 0.00017 μs respectively. Delay of D-Mesh and D-Torus are almost same. Mesh's delay increased sharply at 2.84, exceeding other mentioned topologies' delay and reaching almost 0.22706 μs at the end of the time. In the meantime, the average end to end delay of D-Torus and MD-Torus gradually grew to 0.20685 μs and 0.20263 μs at 1.71.

Table 6.5 Average End to End Delay under Neighbor Traffic

Interpacket Arrival Delay(us)	Mesh	Torus	D-Mesh	D-Torus	MD-Torus
17.07	0.00021	0.00013	0.00017	0.00013	0.00013
8.53	0.00977	0.00971	0.00975	0.00971	0.00013
5.69	0.08349	0.08344	0.08347	0.08344	0.07649
4.27	0.12515	0.12509	0.12513	0.12509	0.11989
3.41	0.15014	0.15009	0.15012	0.15009	0.14592
2.84	0.16681	0.16675	0.16678	0.16675	0.16328
2.44	0.17871	0.17865	0.17868	0.17865	0.17567
2.13	0.18764	0.18758	0.18760	0.18758	0.18497
1.90	0.19458	0.19453	0.19455	0.19453	0.19220
1.71	0.20014	0.20008	0.20010	0.20008	0.19799

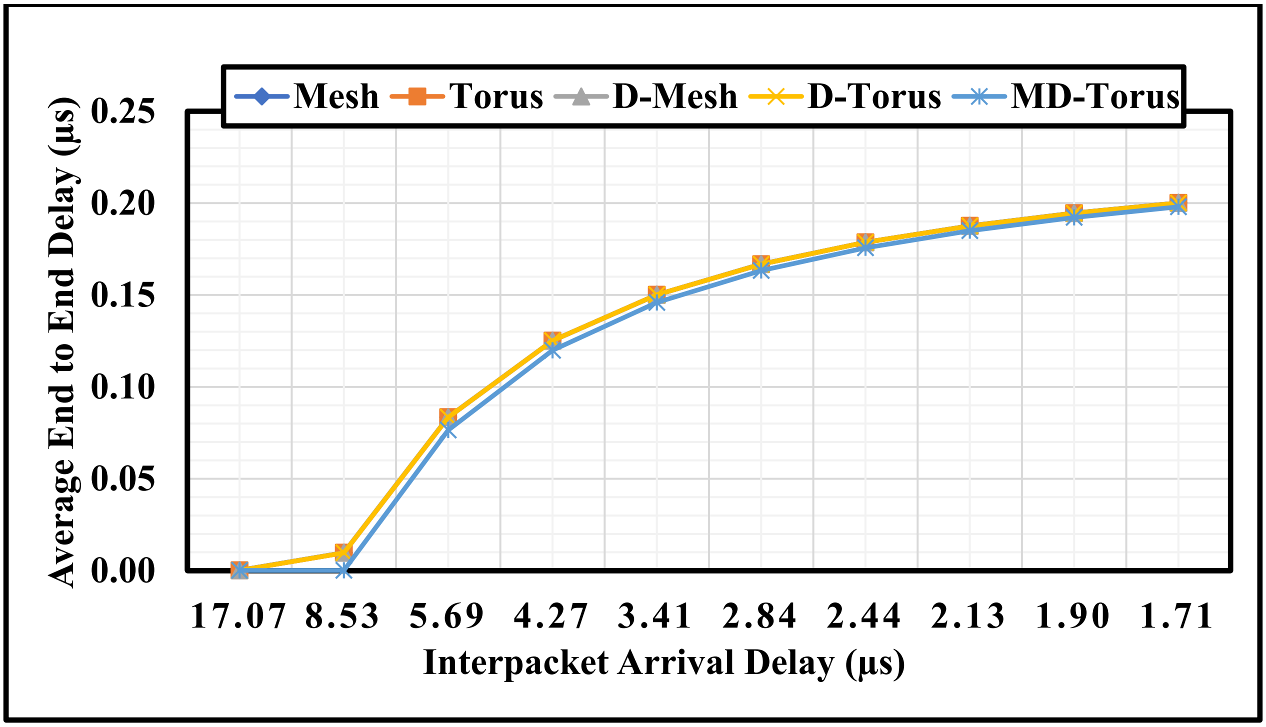

Figure 6.4 Average End to End Delay under Neighbor Traffic

Figure 6.4 gives information about the average end to end delay in the Mesh, Torus, D-Mesh, D-Torus, and MD-Torus based on data points (Interpacket Arrival Delay) provided in Table 6.5. In starting the end to end delay of Mesh, Torus, D-Mesh, D-Torus and MD-Torus experienced a gradual rise and then increased dramatically over the interpacket arrival delay. Delay of Torus and D-torus are nearly same. At 17.07 the end to end delay of D-Mesh, D-Torus and MD-Torus were 0.00017 µs, 0.00013 µs and 0.00013 µs respectively. Mesh's delay increased gently at 8.53, exceeding Torus, D-Mesh, D-Torus, and MD-Torus topologies' delay and reaching almost 0.20014 µs at the end of the time. In the meantime, the average end to end delay of D-Mesh, D-Torus and MD-Torus gradually grew to 0.20010 µs, 0.20008 µs and 0.19799 at 1.71 respectively.

Table 6.6 Average End to End Delay under Tornado Traffic

Interpacket Arrival Delay(us)	Mesh	Torus	D-Mesh	D-Torus	MD-Torus
17.07	0.19439	0.22507	0.07506	0.07796	0.04464
8.53	0.21023	0.23804	0.15033	0.14899	0.08579
5.69	0.21678	0.24140	0.18638	0.18586	0.13804
4.27	0.22135	0.24304	0.20405	0.20343	0.16621
3.41	0.22482	0.24414	0.21418	0.21333	0.18272
2.84	0.22755	0.24492	0.22071	0.21966	0.19358
2.44	0.22975	0.24556	0.22526	0.22407	0.20130
2.13	0.23157	0.24613	0.22860	0.22733	0.20707
1.90	0.23310	0.24654	0.23117	0.22985	0.21157
1.71	0.23440	0.24701	0.23320	0.23184	0.21518

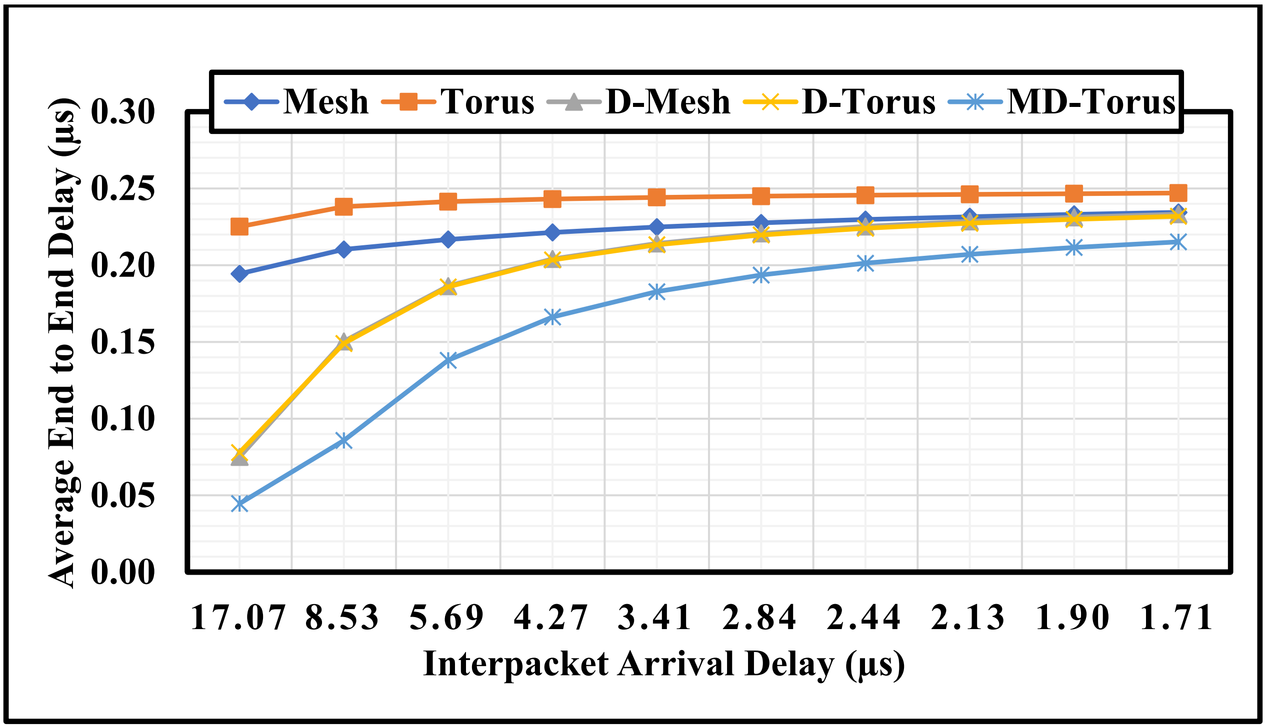

Figure 6.5 Average End to End Delay under Tornado Traffic

Figure 6.5 describes the average end to end delay in the Mesh, Torus, D-Mesh, D-Torus, and MD-Torus based on data points (Interpacket Arrival Delay) provided in Table 6.6. In starting the end to end delay of Mesh, Torus, D-Mesh, D-Torus and MD-Torus showed a gradual increase and then remained almost steady over the interpacket arrival delay. At 5.69 the end to end delay of Mesh and Torus were 0.21678 µs and 0.24140 µs and then rose gently to 0.23440 µs and 0.24701 µs at the end of time respectively. In the meantime, the average end to end delay of D-Mesh, D-Torus and MD-Torus increased considerably to 0.22071 µs, 0.21966 µs and 0.19358 µs at 2.84 and then rose gently to 0.23320 µs, 0.23184 and 0.21518 µs at 1.71 respectively.

Table 6.7 Average End to End Delay under Hotspot Traffic

Interpacket Arrival Delay(us)	Mesh	Torus	D-Mesh	D-Torus	MD-Torus
17.07	0.00814	0.00227	0.00245	0.00109	0.00072
8.53	0.04593	0.01680	0.00799	0.00410	0.00400
5.69	0.07254	0.02580	0.03296	0.00575	0.00416
4.27	0.08862	0.05244	0.05125	0.01063	0.00425
3.41	0.09557	0.07955	0.06079	0.01976	0.00569
2.84	0.10089	0.09610	0.06732	0.03036	0.01049
2.44	0.10441	0.10782	0.07195	0.04374	0.01949
2.13	0.10645	0.11513	0.07466	0.05616	0.03066
1.90	0.10825	0.11958	0.07609	0.06599	0.04171
1.71	0.11052	0.12225	0.07667	0.07364	0.05215

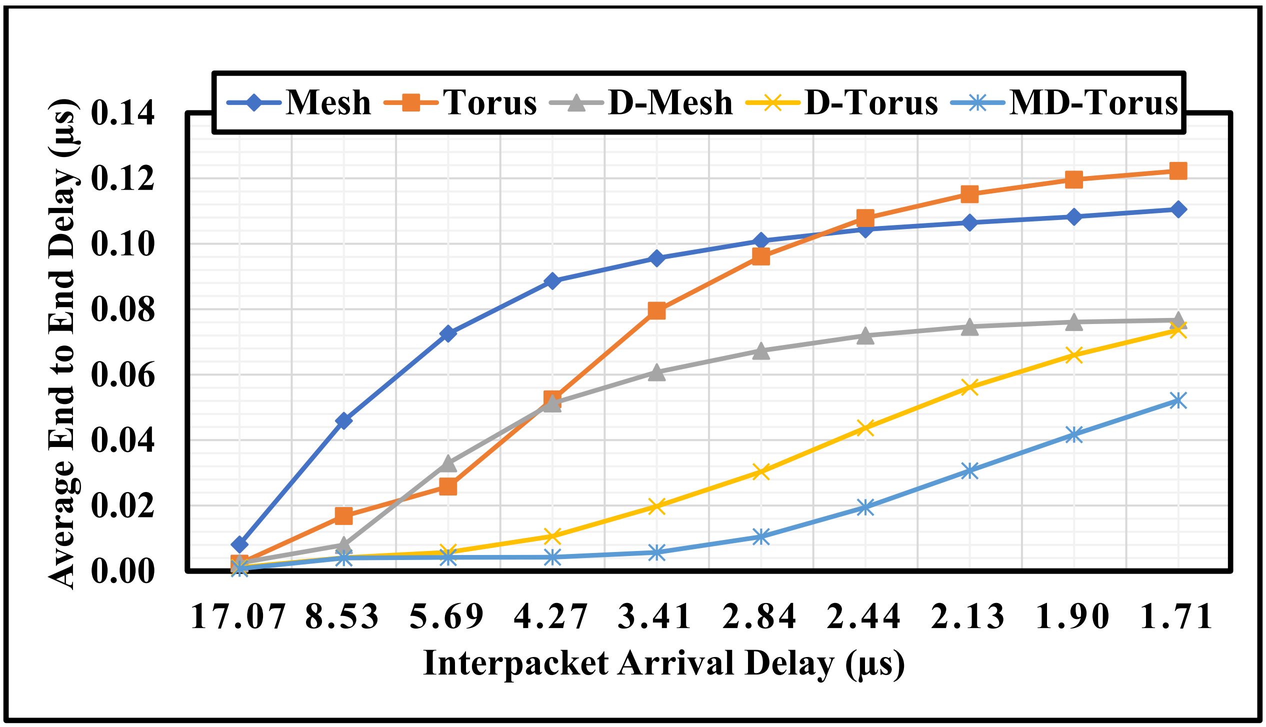

Figure 6.6 Average End to End Delay under Hotspot Traffic

Figure 6.6 indicates the average end to end delay in the Mesh, Torus, D-Mesh, D-Torus, and MD-Torus based on data points (Interpacket Arrival Delay) provided in Table 6.7. The end to end delay of Torus, Mesh and D-Mesh showed a steady but significant rise over the interpacket arrival delay, while in D-Torus and MD-Torus the end to end delay rose gradually. At 8.53 the end to end delay of D-Torus and MD-Torus were 0.00410 µs and 0.00400 µs respectively. Torus's delay increased sharply throughout the time, exceeding Mesh topology delay and reaching almost 0.12225 µs at the end of the time. In the meantime, the average end to end delay of D-Mesh, D-Torus and MD-Torus grew to 0.03296 µs, 0.00575 µs and 0.00416 µs at 5.69 respectively and then increased gradually until 1.71.

Table 6.8 Average Throughput under Uniform Traffic

Interpacket Arrival Delay(us)	Mesh	Torus	D-Mesh	D-Torus	MD-Torus
17.07	58061.52	58562.48	58564.71	58570.55	58572.71
8.53	90463.99	111655.91	114880.46	117138.87	117142.86
5.69	103380.32	149966.08	150788.31	173080.64	175714.40
4.27	109051.59	169907.41	168471.81	222549.45	232067.84
3.41	114654.80	173585.03	183682.47	260091.27	284677.67
2.84	119390.38	174261.33	196150.37	287114.88	329885.52
2.44	124002.47	174215.65	207490.71	304685.02	365073.52
2.13	128169.50	174991.33	218769.16	316983.55	390302.91
1.90	131656.36	176402.24	229970.32	326932.91	407974.25
1.71	134041.22	178252.85	241150.60	335527.63	420779.23

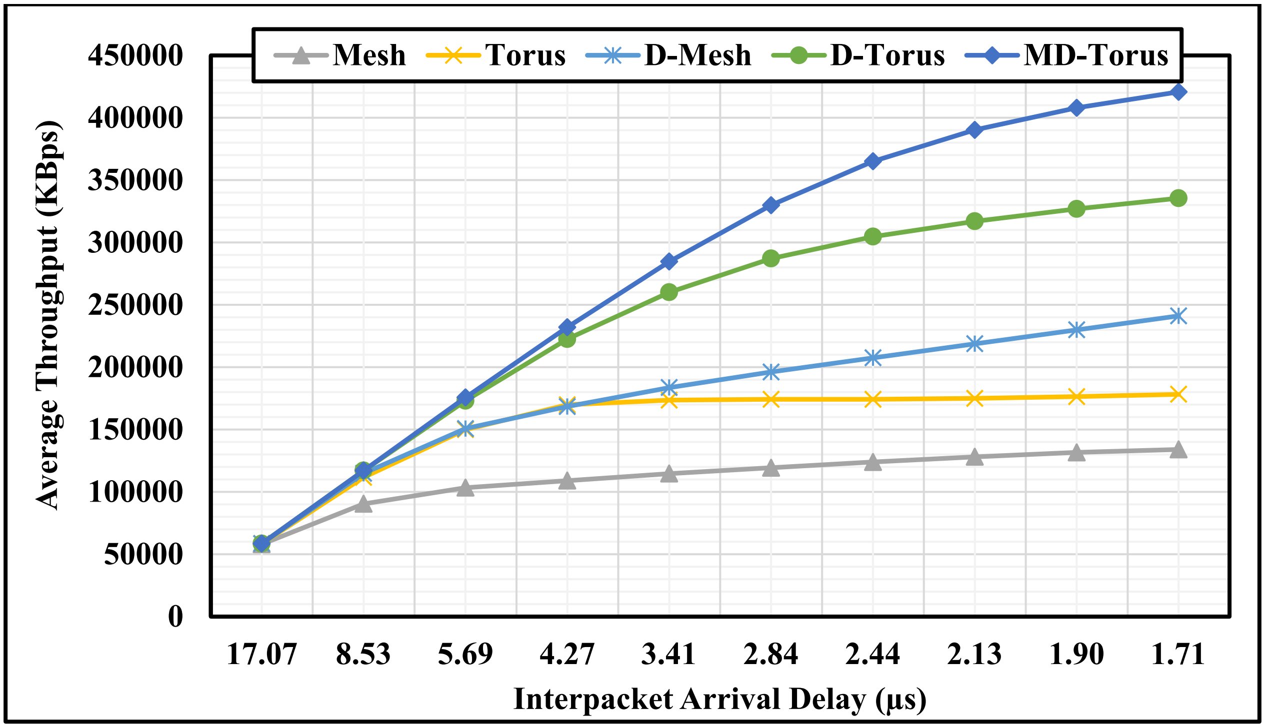

Figure 6.7 Average Throughput under Uniform Traffic

Figure 6.7 shows the average throughput in the Mesh, Torus, D-Mesh, D-Torus and MD-Torus based on data points (Interpacket Arrival Delay) provided in Table 6.8. The average throughput of MD-Torus and D-Torus showed a steady but significant rise over the interpacket arrival delay, while in Mesh, Torus and D-Mesh, the average throughput rose gradually. At 8.53 the average throughput of Torus, D-Mesh, D-Torus and MD-Torus were 111655.91 KBps, 114880.46 KBps, 117138.87 KBps and 117109.74 KBps respectively. MD-Torus' throughput increased sharply throughout the time, exceeding D-Torus, Torus, D-Mesh and Mesh topologies' throughput and reaching almost 420779.23 KBps at the end of the period. In the meantime, the average throughput of D-Mesh, Torus and Mesh grew to 168471.81 KBps, 169907.41 KBps and 109051.59 KBps at 4.27 respectively and then increased gradually until 1.71.

Table 6.9 Average Throughput under Bit Complement Traffic

Interpacket Arrival Delay(us)	Mesh	Torus	D-Mesh	D-Torus	MD-Torus
17.07	40592.63	58562.07	49574.74	58566.23	58572.63
8.53	43720.08	81217.67	53019.27	81220.87	90207.44
5.69	46063.72	85123.30	53353.30	85126.53	94894.49
4.27	47182.17	87466.04	54412.07	87469.48	97237.76
3.41	48462.01	89809.23	55876.73	89812.91	99581.50
2.84	49977.18	92153.01	57580.38	92156.59	101925.36
2.44	51658.28	94496.26	59435.38	94499.94	104268.72
2.13	53459.83	96839.91	61392.62	96843.59	106612.54
1.90	55352.38	99183.41	63421.98	99187.25	108956.37
1.71	57314.37	101526.67	65504.29	101530.51	111299.63

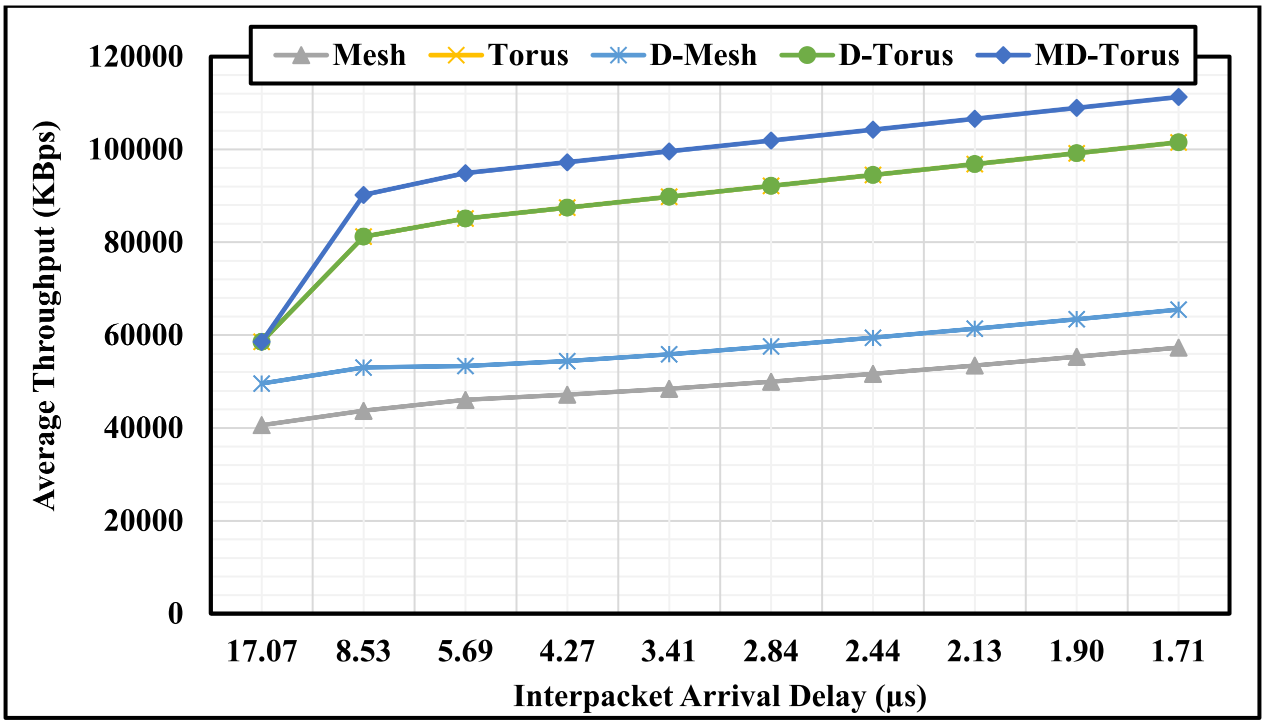

Figure 6.8 Average Throughput under Bit Complement Traffic

Figure 6.8 illustrates the average throughput in the Mesh, Torus, D-Mesh, D-Torus and MD-Torus based on data points (Interpacket Arrival Delay) provided in Table 6.9. In starting, the average throughput of Torus, D-Torus and MD-Torus experienced a significant trend over the interpacket arrival delay and then increased, while in Mesh and D-Mesh, the average throughput rose gradually. At 8.53 the average throughput of Torus, D-Torus and Md-Torus were nearly 81217.67 KBps, 81220.87 KBps and 90207.44 KBps respectively. MD-Torus' throughput increased sharply throughout the time, exceeding D-Torus, D-Mesh, Torus and Mesh topologies throughput and reaching almost 111299.63 KBps at the end of the period. In the meantime, the average throughput of Torus and Mesh grew to 81217.67 KBps and 43720.08 KBps at 8.53 respectively and then rose gradually until 1.71.

Table 6.10 Average Throughput under Neighbor Traffic

Interpacket Arrival Delay(us)	Mesh	Torus	D-Mesh	D-Torus	MD-Torus
17.07	58568.31	58577.43	58572.20	58577.43	58577.83
8.53	112654.63	112666.64	112658.31	112666.64	117157.42
5.69	117145.45	117158.00	117149.13	117158.00	122039.34
4.27	117145.31	117157.92	117149.07	117157.92	122039.34
3.41	117145.43	117157.91	117149.27	117157.91	122039.33
2.84	117145.48	117158.03	117149.40	117158.03	122039.46
2.44	117145.25	117158.05	117149.49	117158.05	122039.56
2.13	117145.32	117158.12	117149.56	117158.12	122039.63
1.90	117145.40	117158.28	117149.72	117158.28	122039.80
1.71	117145.39	117158.19	117149.71	117158.19	122039.78

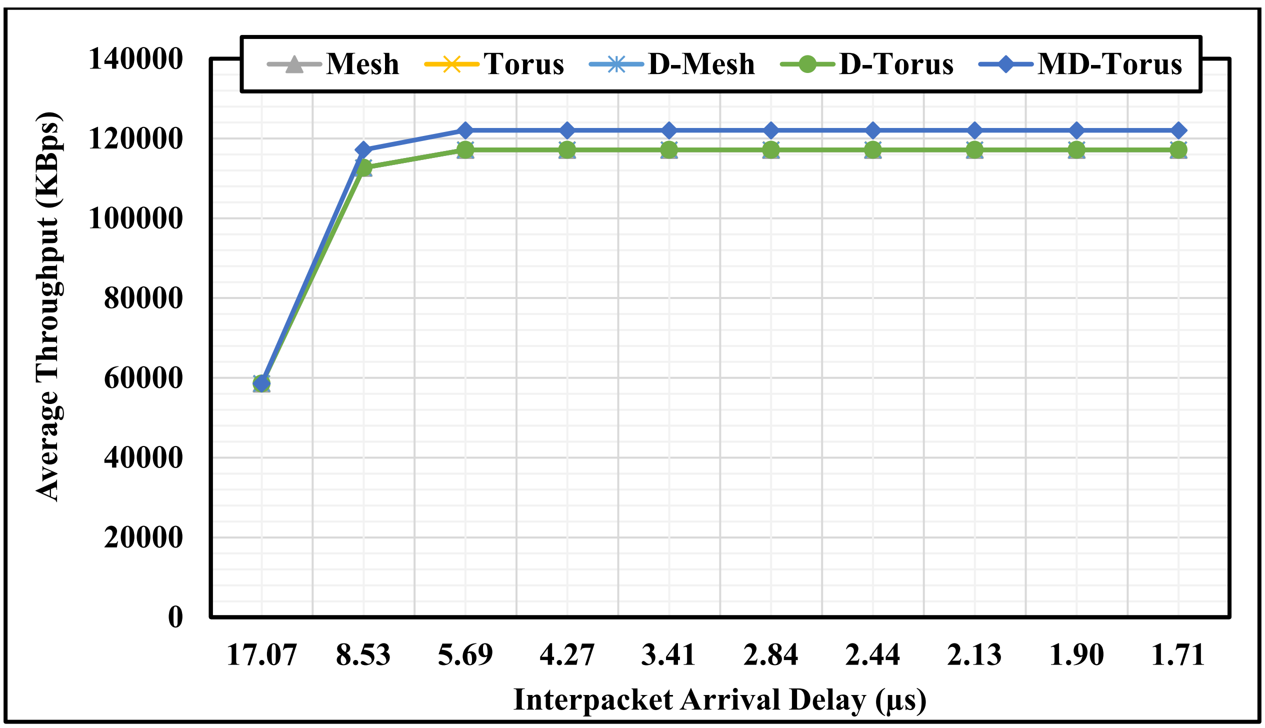

Figure 6.9 Average Throughput under Neighbor Traffic

Figure 6.9 gives information about the average throughput in the Mesh, Torus, D-Mesh, D-Torus and MD-Torus based on data points (Interpacket Arrival Delay) provided in Table 6.10. In starting, the average throughput of Mesh, Torus, D-Mesh, D-Torus and MD-Torus showed a significant rise over the interpacket arrival delay and then remained same. Throughput of Torus and D-Torus are same. At 17.07 the average throughput of Torus, D-Mesh and MD-Torus were 58577.643 KBps, 58572.20 KBps, and 58577.83 KBps and then increased gradually to 117149.13 KBps, 117158.00 KBps and 122039.34 KBps at 5.69 and then remained almost same till 1.71. In the meantime, the average throughput of Mesh grew to 112654.63 KBps at 8.53 and then rose gradually to 117145.45 KBps at 5.69. However, it remained the same at end of the time.

Table 6.11 Average Throughput under Tornado Traffic

Interpacket Arrival Delay(us)	Mesh	Torus	D-Mesh	D-Torus	MD-Torus
17.07	58638.35	29333.92	188911.85	147669.58	213008.99
8.53	57048.39	25691.91	197860.78	151950.78	273833.03
5.69	55897.85	23245.90	188325.11	136638.04	264303.31
4.27	55028.57	21235.91	179037.03	124687.49	252672.56
3.41	54348.57	19553.92	172103.75	115853.92	243841.38
2.84	53802.63	18121.94	166712.75	109004.46	236856.06
2.44	53354.28	16891.96	162393.62	103514.16	231172.66
2.13	52979.91	15823.97	158850.53	99000.63	226443.61
1.90	52661.98	14880.00	155888.75	95214.77	222440.73
1.71	52389.58	17054.36	153374.33	91983.96	218998.30

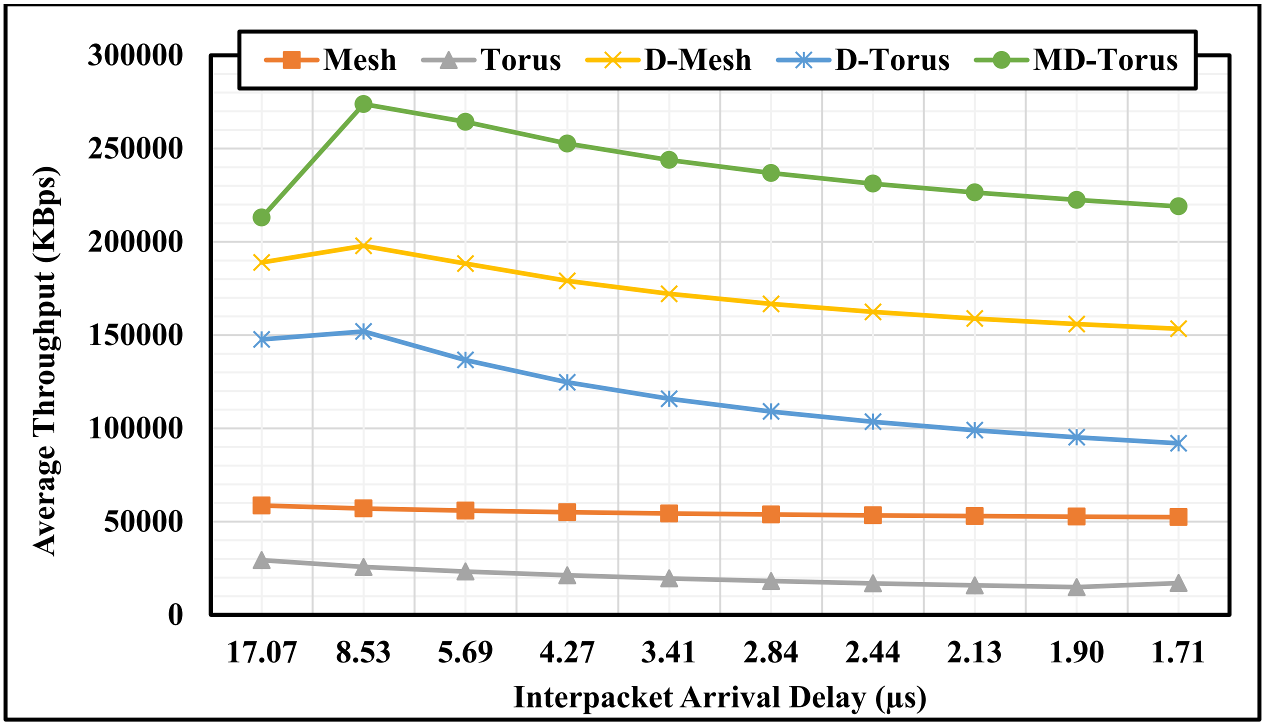

Figure 6.10 Average Throughput under Tornado Traffic

Figure 6.10 indicates the average throughput in the Mesh, Torus, D-Mesh, D-Torus and MD-Torus based on data points (Interpacket Arrival Delay) provided in Table 6.11. In the beginning, the average throughput of MD-Torus showed a significant rise and then experienced a gradual decrease over the interpacket arrival delay, while in Mesh and Torus the average throughput fell gradually. However, there was a gradual increase in D-Mesh and D-Torus in the starting then it fell sharply till end of the time. At 8.53 the average throughput of MD-Torus, D-Mesh and D-Torus were 273833.03 KBps, 197860.78 KBps and 151950.78 KBps respectively. After the interpacket arrival delay of 8.53, MX-Torus' throughput decreased gently throughout the time, exceeding D-Torus, D-Mesh, Torus and Mesh topologies' throughput and reaching almost 218998.30 KBps at the end of the period. In the meantime, the average throughput of Mesh, Torus and D-Torus declined gradually to 55897.85 KBps, 23245.90 KBps and 136638.04 KBps at 5.69 respectively and then remained almost same until 1.71.

Table 6.12 Average Throughput under Hotspot Traffic

Interpacket Arrival Delay(us)	Mesh	Torus	D-Mesh	D-Torus	MD-Torus
17.07	55079.43	57406.63	57245.92	57998.71	58257.43
8.53	84914.95	103012.32	108089.43	110731.51	111059.75
5.69	99081.62	140841.36	139355.90	161905.68	163538.25
4.27	105515.98	159733.65	159076.35	208427.49	215661.30
3.41	111398.23	163831.59	175852.87	245666.47	265727.91
2.84	115827.19	166145.93	189680.54	274197.00	309906.64
2.44	120024.39	166655.99	201590.02	292630.57	344141.41
2.13	124134.58	167330.93	212824.21	305492.50	369548.39
1.90	127625.49	168573.44	223862.84	316230.07	388815.03
1.71	129960.14	170252.29	234798.21	325780.28	402958.35

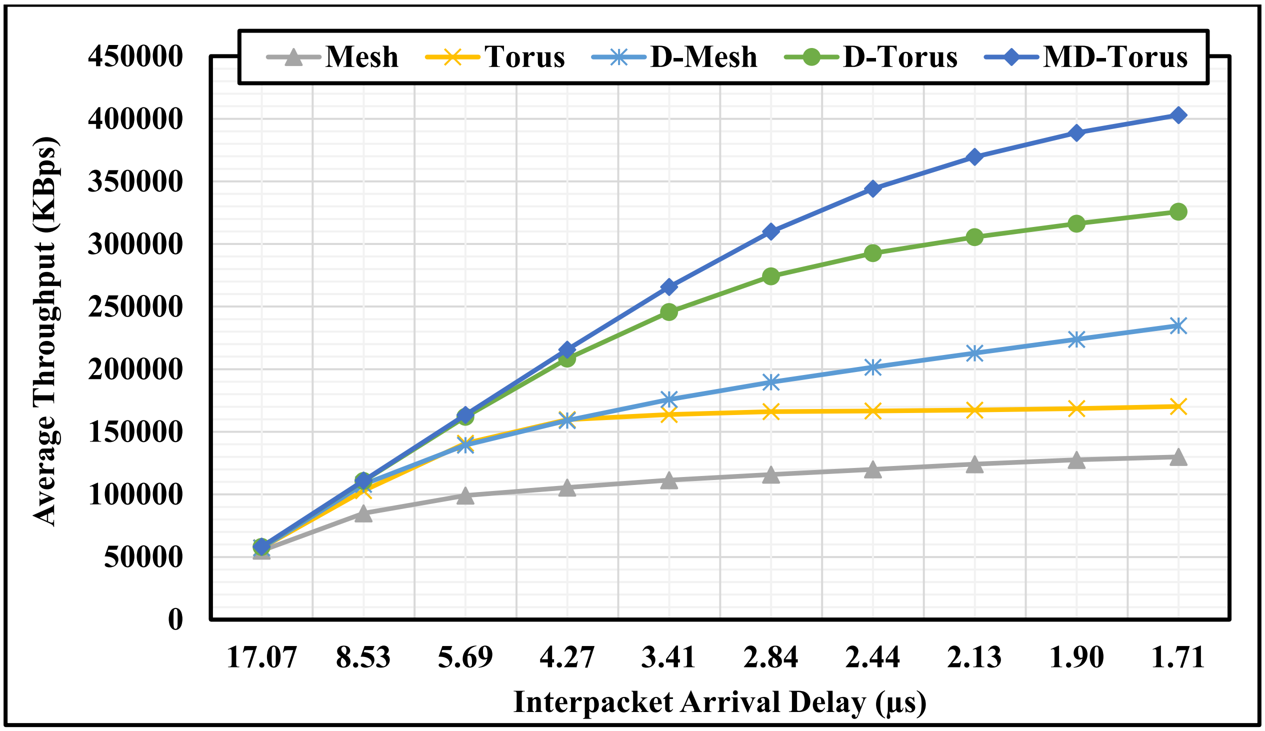

Figure 6.11 Average Throughput under Hotspot Traffic

Figure 6.11 describes the average throughput in the Mesh, Torus, D-Mesh, D-Torus and MD-Torus based on data points (Interpacket Arrival Delay) provided in Table 6.12. The average throughput of MD-Torus and D-Torus showed a steady but significant rise over the interpacket arrival delay, while in Mesh, Torus and D-Mesh, the average throughput rose gradually. At 8.53 the average throughput of Torus, D-Mesh, D-Torus and MD-Torus were 103012.32 KBps, 108089.43 KBps, 110731.51 KBps and 111059.75 KBps respectively. MD-Torus' throughput increased sharply throughout the time, exceeding D-Torus, Torus, D-Mesh and Mesh topologies' throughput and reaching almost 402958.35 KBps at the end of the period. In the meantime, the average throughput of D-Mesh, Torus and Mesh grew to 159076.35 KBps, 159737.65 KBps and 105515.98 KBps at 4.27 respectively and then increased gradually until 1.71.

Table 6.13 Average Hop Count under Uniform Traffic

Interpacket Arrival Delay(us)	Mesh	Torus	D-Mesh	D-Torus	MD-Torus
17.07	3.18904	2.40158	2.26470	1.82482	1.67707
8.53	2.86184	2.36803	2.24986	1.82408	1.67656
5.69	2.54338	2.28315	2.14186	1.81495	1.67677
4.27	2.29060	2.18022	2.00107	1.79354	1.67074
3.41	2.10101	2.05743	1.88755	1.75758	1.65996
2.84	1.94998	1.94852	1.79123	1.71600	1.64368
2.44	1.82617	1.85090	1.70934	1.67115	1.62320
2.13	1.72065	1.76578	1.64015	1.62790	1.59953
1.90	1.63042	1.69134	1.58124	1.58820	1.57405
1.71	1.54987	1.62522	1.53023	1.55139	1.54805

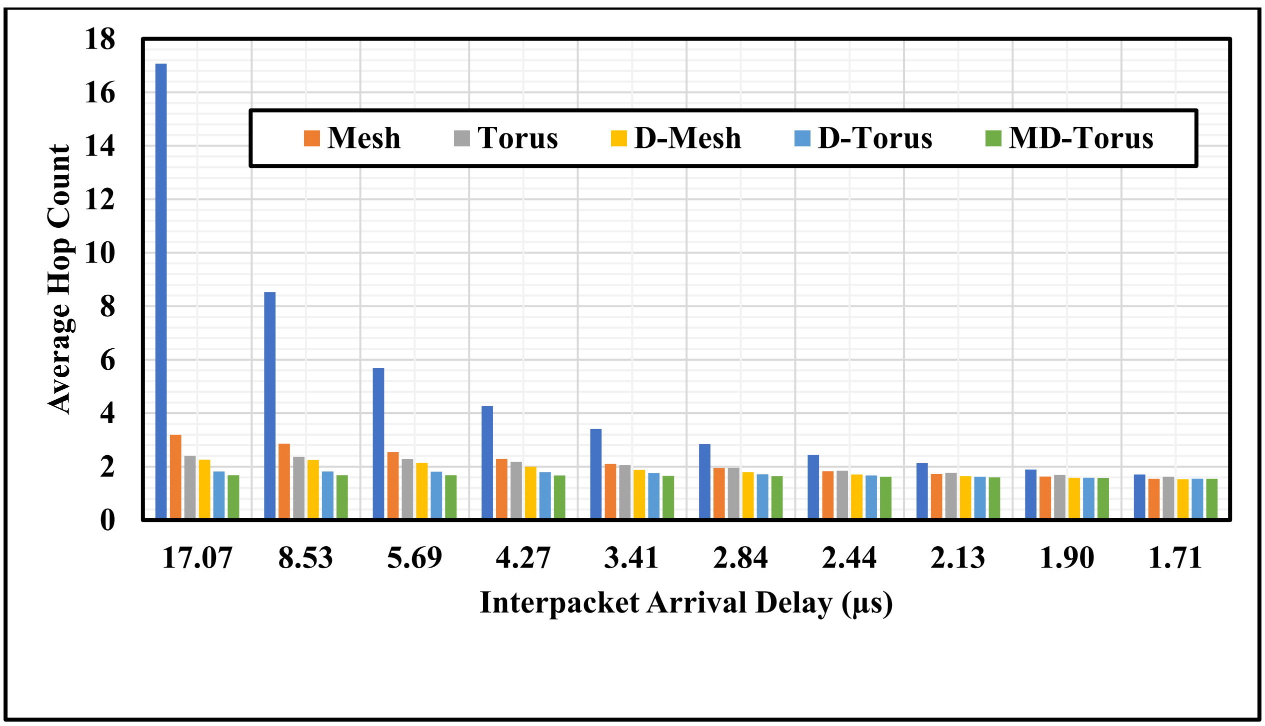

Figure 6.12 Average Hop Count under Uniform Traffic

Figure 6.12 shows the average hop count in the Mesh, Torus, D-Mesh, D-Torus and MD-Torus based on data points (Interpacket Arrival Delay) provided in Table 6.13. Overall, Mesh, Torus, D-Mesh, D-Torus and MD-Torus experienced a downwards trend throughout the time. MD-Torus hop count was 1.67707 at 17.07 being lower than Mesh, Torus, D-Mesh and D-Torus topology's hop count and continued to decline steadily to the end of the time, reaching around 1.65996 at 3.41 and hitting low point of 1.54805 at 1.71 interpacket arrival delay. Same pattern was also observed for Mesh, Torus, D-Mesh and D-Torus topologies.

Table 6.14 Average Hop Count under Bit Complement Traffic

Interpacket Arrival Delay(us)	Mesh	Torus	D-Mesh	D-Torus	MD-Torus
17.07	4.8	2.4	3.2	2.08	1.6
8.53	4.8	2.4	3.2	2.08	1.6
5.69	4.8	2.4	3.2	2.08	1.6
4.27	4.8	2.4	3.2	2.08	1.6
3.41	4.8	2.4	3.2	2.08	1.6
2.84	4.8	2.4	3.2	2.08	1.6
2.44	4.8	2.4	3.2	2.08	1.6
2.13	4.8	2.4	3.2	2.08	1.6
1.90	4.8	2.4	3.2	2.08	1.6
1.71	4.8	2.4	3.2	2.08	1.6

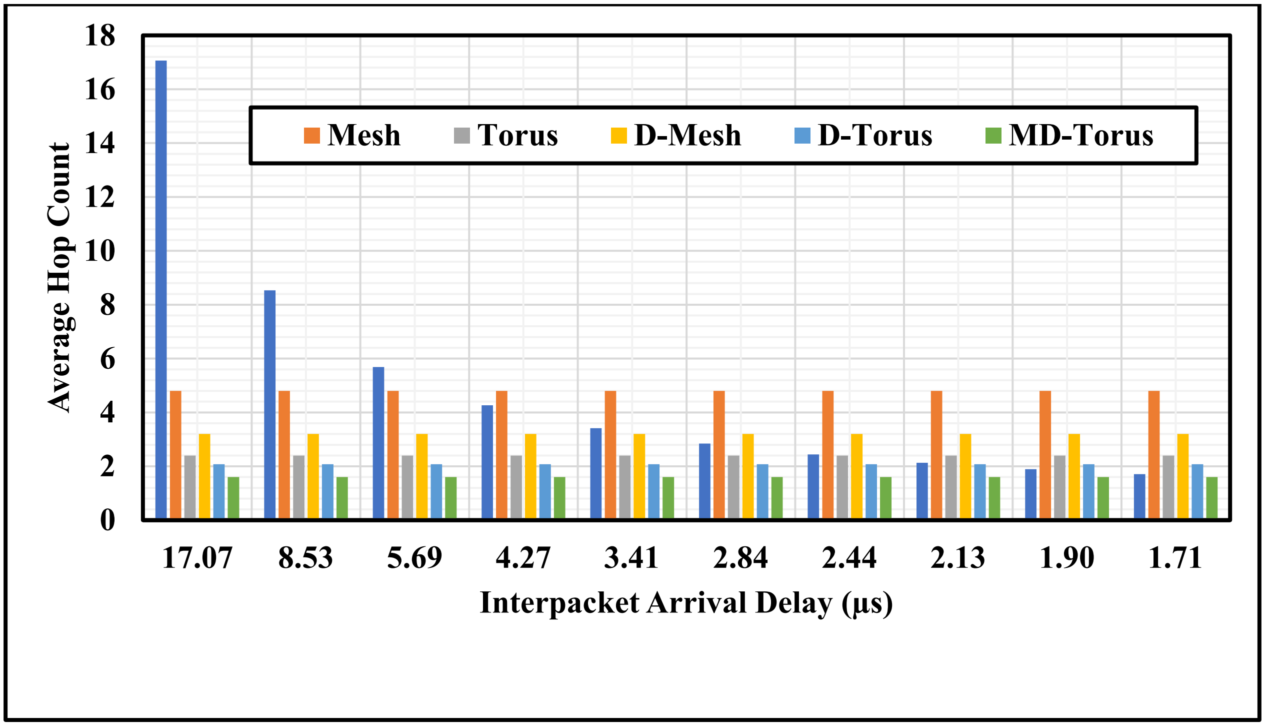

Figure 6.13 Average Hop Count under Bit Complement Traffic

Figure 6.13 describes the average hop count in the Mesh, Torus, D-Mesh, D-Torus and MD-Torus based on data points (Interpacket Arrival Delay) provided in Table 6.14. In general, Mesh, Torus, D-Mesh, D-Torus and MX-Torus experienced a constant trend throughout the time. The average hop count of Mesh, Torus, D-Mesh, D-Torus, and MD-Torus was 4.8, 2.4, 3.2, 2.08 and 1.6 respectively. Hop count of MD-Torus topology was smaller than other mentioned topologies and found to be 1.6 throughout the time.

Table 6.15 Average Hop Count under Neighbor Traffic

Interpacket Arrival Delay(us)	Mesh	Torus	D-Mesh	D-Torus	MD-Torus
17.07	1.92	1.2	1.6	1.2	1.16
8.53	1.92	1.2	1.6	1.2	1.16
5.69	1.92	1.2	1.6	1.2	1.16
4.27	1.92	1.2	1.6	1.2	1.16
3.41	1.92	1.2	1.6	1.2	1.16
2.84	1.92	1.2	1.6	1.2	1.16
2.44	1.92	1.2	1.6	1.2	1.16
2.13	1.92	1.2	1.6	1.2	1.16
1.90	1.92	1.2	1.6	1.2	1.16
1.71	1.92	1.2	1.6	1.2	1.16

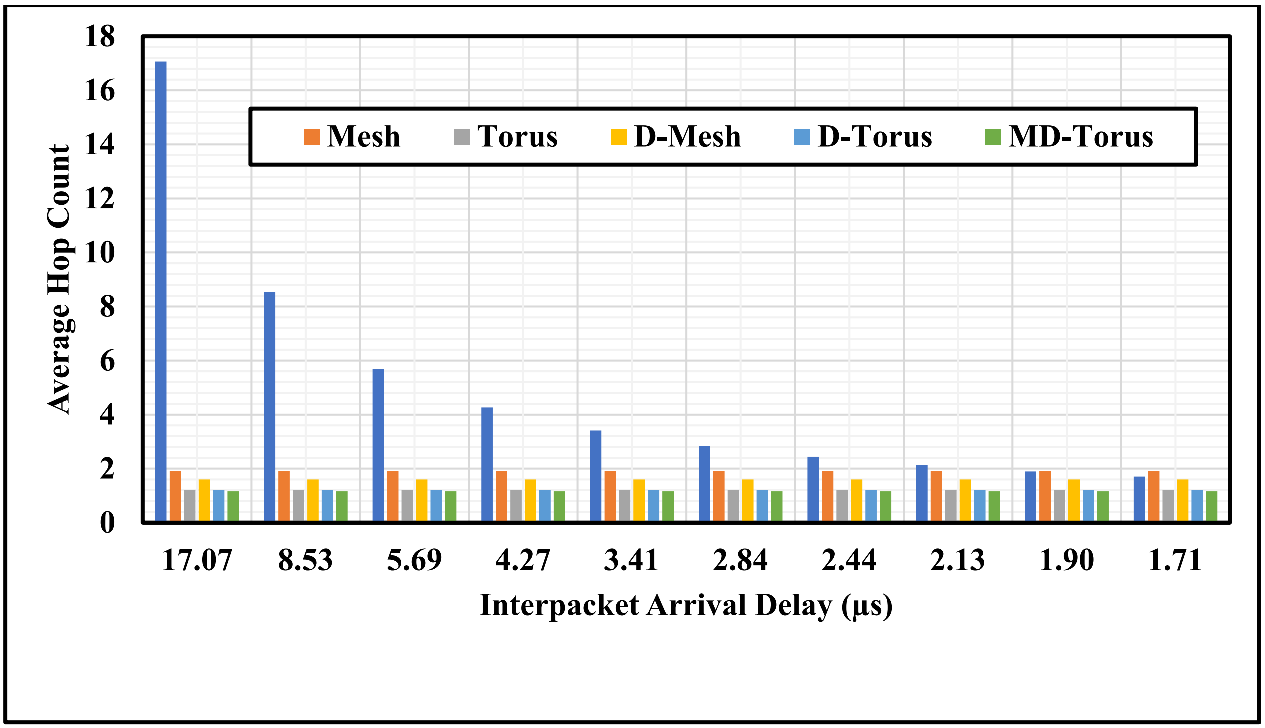

Figure 6.14 Average Hop Count under Neighbor Traffic

Figure 6.14 provides the information about the average hop count in the Mesh, Torus, D-Mesh, D-Torus and MD-Torus based on data points (Interpacket Arrival Delay) provided in Table 6.15. Overall, Mesh, Torus, D-Mesh, D-Torus and MD-Torus experienced a constant trend throughout the time. The average hop count of Mesh, Torus, D-Mesh, D-Torus, and MD-Torus was 1.92, 1.2, 1.6, 1.2 and 1.16 respectively. Hop count of MD-Torus topology was smaller than other mentioned topologies and found to be 1.16 throughout the time.

Table 6.16 Average Hop Count under Tornado Traffic

Interpacket Arrival Delay(us)	Mesh	Torus	D-Mesh	D-Torus	MD-Torus
17.07	3.98713	3.19875	2.83674	2.43916	2.02102
8.53	3.36801	3.01152	2.81508	2.42826	1.98749
5.69	3.07285	2.81700	2.76124	2.41138	1.97825
4.27	2.91139	2.68167	2.72318	2.40092	1.97375
3.41	2.81101	2.58443	2.69570	2.39441	1.97077
2.84	2.74287	2.51120	2.67489	2.39013	1.96860
2.44	2.69375	2.45418	2.65857	2.38721	1.96691
2.13	2.65661	2.40799	2.64542	2.38522	1.96555
1.90	2.62774	2.37030	2.63458	2.38382	1.96441
1.71	2.60471	2.33879	2.62549	2.38286	1.96345

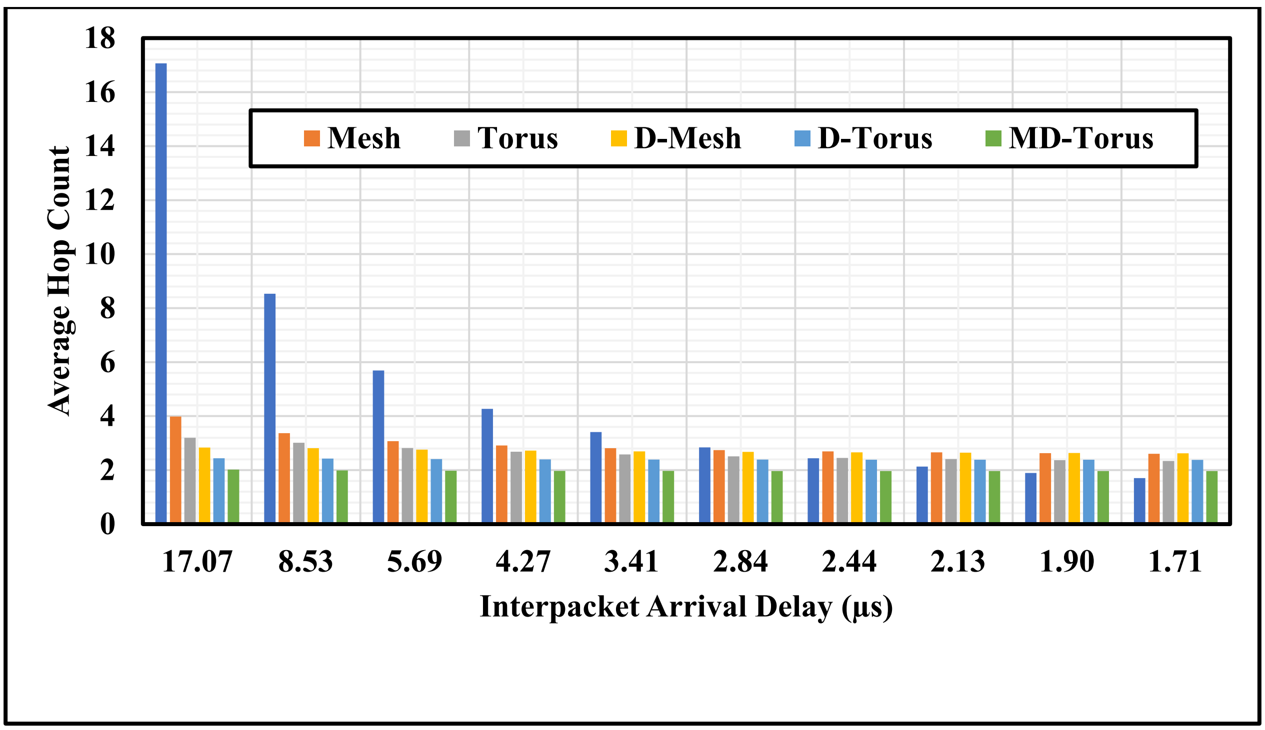

Figure 6.15 Average Hop Count under Tornado Traffic

Figure 6.15 indicates the average hop count in the Mesh, Torus, D-Mesh, D-Torus and MD-Torus based on data points (Interpacket Arrival Delay) provided in Table 6.16. In general, Mesh, Torus, D-Mesh, D-Torus and MD-Torus experienced a downwards trend throughout the time. MD-Torus hop count was 2.02102 at 17.07 being lower than Mesh, Torus, D-Mesh and D-Torus topologies' hop count and continued to decline steadily to the end of the time, reaching around 1.97077 at 3.41 and hitting low point of 1.96345 at 1.71 interpacket arrival delay. Same pattern is also observed for other mentioned topologies.

Table 6.17 Average Hop Count under Hotspot Traffic

Interpacket Arrival Delay(us)	Mesh	Torus	D-Mesh	D-Torus	MD-Torus
17.07	3.16822	2.44535	2.25256	1.83810	1.69599
8.53	2.86186	2.37409	2.22940	1.82674	1.68335
5.69	2.56708	2.30082	2.10675	1.81470	1.67511
4.27	2.32875	2.19983	1.98160	1.79677	1.66934
3.41	2.14647	2.08187	1.87919	1.76661	1.66189
2.84	1.99732	1.97991	1.79267	1.73159	1.65015
2.44	1.87563	1.88959	1.71772	1.69163	1.63304
2.13	1.77394	1.80953	1.65289	1.65247	1.61269
1.90	1.68587	1.73884	1.59695	1.61631	1.59089
1.71	1.60838	1.67602	1.54842	1.58296	1.56817

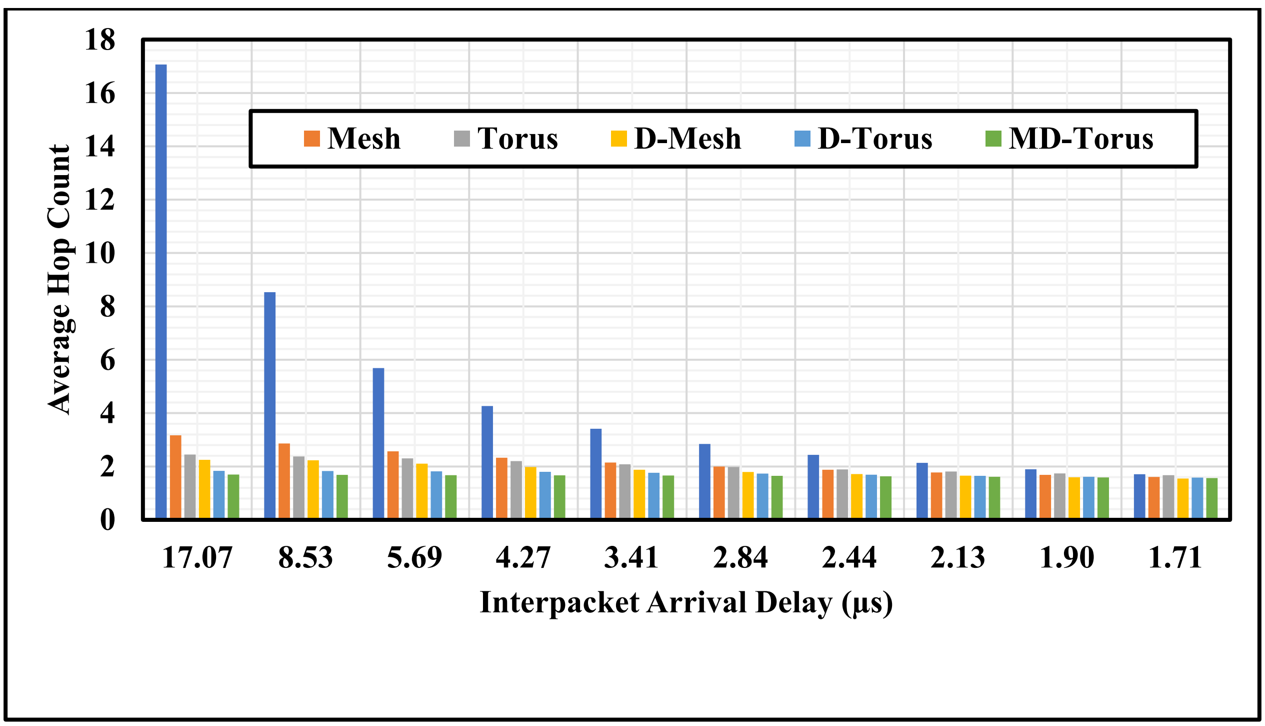

Figure 6.16 Average Hop Count under Hotspot Traffic

Figure 6.16 illustrates the average hop count in the Mesh, Torus, D-Mesh, D-Torus and MD-Torus based on data points (Interpacket Arrival Delay) provided in Table 6.17. Overall, Mesh, Torus, D-Mesh, D-Torus and MD-Torus experienced a downwards trend throughout the time. MD-Torus hop count was 1.69599 at 17.07 being lower than Mesh, Torus, D-Mesh and D-Torus topologies' hop count and continued to decline steadily to the end of the time, reaching around 1.66189 at 3.41 and hitting low point of 1.56817 at 1.71 interpacket arrival delay. Same pattern is also observed for other mentioned topologies.

6.4 Results and Discussions

This chapter presented an improved version of D-Torus topology, called as MD-Torus and is suitable for parallel processing system. Based on results obtained in section 5.3, the proposed topology revealed better performance in comparison to Mesh, Torus, D-Mesh, and D-Torus topologies. Maximum reduction in end to end delay was obtained i.e. 98.71% under neighbor traffic. There was a major improvement in average throughput i.e. 57.99% under tornado traffic while average hop count was reduced by 23.08% under the influence of bit complement traffic. The proposed topology provided better path diversity that can handle fault tolerance of network in an efficient way. This topology inherits the properties of D-Mesh as well as D-Torus topologies. In terms of degree, diameter, path diversity, average distance and bisectional bandwidth the proposed topology provided better performance. Due to its enhanced feature and performance it could be a better option for the large scale of parallel computing system. Conclusively, the studied topology could further be explored to attain high performance in various quality of service parameter because of efficient inter processor communication.

CHAPTER 7
CONCLUSIONS AND FUTURE SCOPE

The study focused on the influence of the topology on the performance of the parallel computing system as there are several interconnection topologies. The exhaustive survey on interconnection network prompted us to design efficient topologies to achieve the optimal performance. At present, several interconnection network topologies in the parallel computing system exist, therefore it was a difficult task to select one of them for a specific objective. It was observed that the selection of topology depends on topological properties as well as quality of services for an application. Therefore, the key objective of this thesis was to perform comprehensive analysis of existing topologies and design new efficient topologies to improve the performance of parallel computing systems.

In chapter 1, we have focused on the introduction of interconnection networks. The chapter also contains research gaps, problem reflecting objectives.

In chapter 2, we have presented the empirical information via comprehensive literature search presented background of various interconnection networks, approach for analyzing data, and calculations corresponding to topological properties.

The chapter 3 described the design of Modified X-Torus topology as the enhanced version of X-Torus topology and its topological properties were calculated. We evaluated its performance using OMNeT++ simulator under various traffic patterns such as uniform, bit complement, neighbor, tornado and hotspot. Proposed topology was compared with Mesh, Torus, and X-Torus topologies. Simulation work was performed to evaluate its performance in the real-time applications such as weather forecasting, earthquake predictions and other applications which require real time data. In addition, it was found that proposed topology was more fault tolerant due to better path diversity compared to other topologies. It also provided minimal average latency and average hop count as well as maximum average throughput.

In chapter 4, we proposed a new topology, called as Center Concentrated X-Torus and it is the improved version of X-Torus topology where center nodes were connected to the vertical and horizontal edge nodes. Various topological properties were calculated such as node degree, diameter, bisection width, average, and path diversity. Quality of services were also evaluated and found that its performance was comparatively better.

Similarly, chapter 5 and 6 also report to design two efficient topologies. These were Hexagonal X-Torus and Modified Diagonal Torus topologies. After calculating topological properties and evaluating quality of services, it was concluded that the proposed topologies improved the performance of the parallel computing system to a large scale. As the proposed topologies have high path diversity so it could provide better fault tolerance in the network in case of node, as well as link failure. Ideally these findings revealed it as a better option to improve the performance of parallel computing systems.

Overall, present thesis objectives were broadly focused on the comprehensive study of various interconnection network topologies and its topological properties to enhance the path diversity and improving the quality of services for the topologies. However, some new topologies were designed which have prominent topological attribute. In the study of topologies it was identified that vertex in MX-Torus topology were not vertex symmetric. If the topology is not vertex symmetric this means that design cost is wasted which can be utilized for improving path diversity and performance of the topology. The key challenge was to identify these nodes and define link between them. These links are to be represented by the mathematical formulation, which should be true for all the topologies for the even and odd dimensions. Secondly, in order to improve the path diversity additional links were added to center node to X-Torus topology and termed as center concentrated X torus topology with the objective to provide the alternate path that can help in improving the performance of the system. In order to improve the performance to most important parameter is the internode distance. The hexagonal links in the Hexagonal X-Torus and diagonal link introduce in modified diagonal torus topologies helped in reducing the hop count of topologies. Even though the topologies were designed but key challenge in this exploration is automatic exploration of links based on multiple parameter. This work can be further extending using machine learning algorithms.

LIST OF PUBLICATIONS

Journal Publications:

Published:

1. Dinesh Kumar, Vivek Kumar Sehgal, Nitin, “A Modified X-Torus Topology for Interconnection Network,” Journal of Telecommunication, Electronic and Computer Engineering (JTEC), 9(3-6), 49-54.

2. Dinesh Kumar, Vivek Kumar Sehgal, Nitin, “Center Concentrated X-Torus Topology,” Journal of Telecommunication, Electronic and Computer Engineering (JTEC) 9(3-6), 55-60.

Communicated:

3. Dinesh Kumar, Vivek Kumar Sehgal, Nitin, “A Hexagonal X-Torus Topology for Interconnection Network”, Springer Computing.

Conference Publications:

Paper Presented:

4. Dinesh Kumar, Vivek Kumar Sehgal, Nitin, “A Modified X-Torus Topology for Interconnection Network,” RICSIT 2017 HPU Shimla.

5. Dinesh Kumar, Vivek Kumar Sehgal, Nitin, “Center Concentrated X-Torus Topology,” RICSIT 2017 HPU Shimla.

6. Dinesh Kumar, Vivek Kumar Sehgal, Nitin, “Performance Evaluation of Routing Algorithms on Network on Chips,” INDIA Com – 2018 (Paper will appear in IEEE Xplore).

REFERENCES

[1] W. J. Dally and B. Towles, *Principles and Practices of Interconnection Networks*. Elsevier, 2004.

[2] J. Najarian, “Introduction to parallel computing,” *Control Eng. Pract.*, 1993.

[3] C. H. Hsu and V. Salapura, “Network and Parallel Computing,” *International Journal of Parallel Programming*, 2016.

[4] L. Eeckhout, “Computer Architecture Performance Evaluation Methods,” *Synth. Lect. Comput. Archit.*, 2010.

[5] R. Jain, *The art of computer systems performance analysis: techniques for experimental design, measurement, simulation, and modeling*. John Wiley & Sons, 1990.

[6] M. J. Flynn, “Computer organization and architecture,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 1978.

[7] W. Stallings, *Computer Organization and Architecture Designing For Performance*. 2010.

[8] B. Wilkinson, *Computer architecture: design and performance*. Prentice-Hall, Inc., 1991.

[9] 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,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 3984 LNCS, 2006, pp. 149–157.

[10] R. Duncan, “A survey of parallel computer architectures,” *Computer (Long. Beach. Calif).*,

vol. 23, no. 2, pp. 5–16, 1990.

[11] W. J. Dally and J. W. Poulton, *Digital Systems Engineering*, no. April. Cambridge: Cambridge University Press, 1998.

[12] W. J. Dally and B. Towles, “Route packets, not wires: on-chip interconnection networks,” in *Proceedings of the 38th Design Automation Conference (IEEE Cat. No.01CH37232)*, 2001, pp. 684–689.

[13] 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.

[14] 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.

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

[16] Z. Cvetanovic, “Performance analysis of the Alpha 21364-based HP GS1280 multiprocessor,” in *30th Annual International Symposium on Computer Architecture, 2003. Proceedings.*, 2003, pp. 218–228.

[17] S. L. Scott, G. M. Thorson, and E. Al, “The Cray T3E Network: Adaptive Routing in a High Performance 3D Torus,” *Proc. Hot Interconnects IV Symp.*, pp. 147–156, 1996.

[18] IBM Blue Gene team, “The IBM Blue Gene project,” *IBM J. Res. Dev.*, vol. 57, no. 1/2, p. 0:1-0:6, Jan. 2013.

[19] 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.

[20] 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.

[21] 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*, 2010, pp. 1–4.

[22] J. M. Camara *et al.*, "Twisted Torus topologies for enhanced interconnection networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 21, no. 12, pp. 1765–1778, 2010.

[23] A. Asaduzzaman, S. R. Chaturvedula, and R. Pendse, "A novel folded-torus based network architecture for power-aware multicore systems," *Comput. Electr. Eng.*, vol. 39, no. 8, pp. 2494–2506, Nov. 2013.

[24] 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.

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

[26] J. Balfour and W. J. Dally, "Design tradeoffs for tiled CMP on-chip networks," in *ACM International Conference on Supercomputing 25th Anniversary Volume*, 2014, pp. 390–401.

[27] J. Camacho, J. Flich, J. Duato, H. Eberle, and W. Olesinski, "Towards an efficient NoC topology through multiple injection ports," in *Proceedings - 2011 14th Euromicro Conference on Digital System Design: Architectures, Methods and Tools, DSD 2011*, 2011, pp. 165–172.

[28] J. Camacho, J. Flich, J. Duato, H. Eberle, and W. Olesinski, “A power-efficient network on-chip topology,” in *Proceedings of the Fifth International Workshop on Interconnection Network Architecture On-Chip, Multi-Chip - INA-OCMC ’11*, 2011, pp. 23–26.

[29] 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.

[30] 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.

[31] 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.

[32] A. W. Yin, T. C. Xu, P. Liljeberg, and H. Tenhunen, “Explorations of honeycomb topologies for network-on-chip,” in *NPC 2009 - 6th International Conference on Network and Parallel Computing*, 2009.

[33] C. Decayeux and D. Semé, “3D hexagonal network: Modeling, topological properties, addressing scheme, and optimal routing algorithm,” *IEEE Trans. Parallel Distrib. Syst.*, 2005.

[34] K. W. Tang and S. A. Padubidri, “Diagonal and toroidal mesh networks,” *IEEE Trans. Comput.*, vol. 43, no. 7, pp. 815--826, 1994.

[35] M. H. Furhad and J. M. Kim, “An extended diagonal mesh topology for network-on-chip architectures,” *Int. J. Multimed. Ubiquitous Eng.*, 2015.

[36] M. Tahghighi, M. Mousavi, P. Khadivi, and K. Bazargan, “A new hybrid topology for network on chip,” in *ICEE 2012 - 20th Iranian Conference on Electrical Engineering*, 2012.

[37] F. N. Sibai, “A two-dimensional low-diameter scalable on-chip network for interconnecting thousands of cores,” *IEEE Trans. Parallel Distrib. Syst.*, 2012.

[38] H.-J. Cho and L.-Y. Hsu, “Generalized honeycomb torus,” *Inf. Process. Lett.*, vol. 86, no. 4, pp. 185–190, May 2003.

[39] 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.

[40] S. Yadav and C. R. Krishna, “CCTorus: A New Torus Topology for Interconnection Networks,” in *International Conference on Advanced Computational Technologies and Creative Media (ICACTCM’2014) August 14-15, 2014 Pattaya (Thailand)*, 2014, pp. 8–14.

[41] 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.

[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, Oct. 2011.

[43] 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 *2012 IEEE 26th International Parallel and Distributed Processing Symposium Workshops & PhD Forum*, 2012, pp. 2061–2068.

[44] Y. Liu, M. Zhu, L. 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.

[45] 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, Mar. 2012.

[46] A. Bakhoda, J. Kim, and T. M. Aamodt, "Throughput-effective on-chip networks for manycore accelerators," in *Proceedings of the Annual International Symposium on Microarchitecture, MICRO*, 2010.

[47] J. Duato, S. Yalamanchily, and L. M. Ni, *Interconnection Networks: an Engineering Approach*. Morgan Kaufmann, 2003.

[48] N. Economides, "The economics of networks," *Int. J. Ind. Organ.*, 1996.

[49] D. Wentzlaff *et al.*, "On-chip interconnection architecture of the tile processor," *IEEE Micro*, 2007.

[50] R. Peñaranda, M. E. Gómez, P. López, E. Gunnar Gran, and T. Skeie, "A fault-tolerant routing strategy for k-ary n-direct s-indirect topologies based on intermediate nodes," in *Concurrency Computation*, 2017.

[51] N. S. Contractor, L. A. DeChurch, J. Carson, D. R. Carter, and B. Keegan, "The topology of collective leadership," *Leadersh. Q.*, 2012.

[52] V. Ramamurthi, A. Reaz, D. Ghosal, S. Dixit, and B. Mukherjee, "Channel, capacity, and flow assignment in wireless mesh networks," *Comput. Networks*, 2011.

[53] W. Chang, L. Yubai, and C. Song, "Design and simulation of a Torus topology for network on chip," *J. Syst. Eng. Electron.*, vol. 19, no. 4, pp. 694–701, Aug. 2008.

[54] D. Kumar, V. K. Sehgal, and N. Nitin, "A Modified X-Torus Topology for Interconnection Network," *J. telecomunication Electron. Comput. Eng.*, vol. 9, no. 3, pp. 49–54, 2017.

[55] F. Jordán and I. Scheuring, “Network ecology: Topological constraints on ecosystem dynamics,” *Physics of Life Reviews*. 2004.

[56] N. Kashtan, S. Itzkovitz, R. Milo, and U. Alon, “Topological generalizations of network motifs,” *Phys. Rev. E - Stat. Physics, Plasmas, Fluids, Relat. Interdiscip. Top.*, 2004.

[57] P. B. M. and D. B. Jing-Quan Li, “The k -Degree Cayley Graph and its Topological Properties,” *Networks*, vol. 47, no. 1, pp. 26–36, 2006.

[58] A. Varga, “Using the OMNeT++ discrete event simulation system in education,” *IEEE Trans. Educ.*, vol. 42, no. 4 [+CDROM], p. 11 pp., 1999.

[59] S. Knight, H. X. Nguyen, N. Falkner, R. Bowden, and M. Roughan, “The internet topology zoo,” *IEEE J. Sel. Areas Commun.*, 2011.

[60] T. yun Feng, “SURVEY OF INTERCONNECTION NETWORKS.,” *Computer (Long. Beach. Calif).*, 1981.

[61] W. J. Dally, “Virtual-Channel Flow Control,” *IEEE Trans. Parallel Distrib. Syst.*, 1992.

[62] M. Bhardwaj, “C2 Torus New Interconnection Network Topology Based on 2D Torus,” *Am. J. Networks Commun.*, vol. 4, no. 3, p. 1, 2015.

[63] V. Chaudhary and J. K. Aggarwal, “A Generalized Scheme for Mapping Parallel Algorithms,” *IEEE Trans. Parallel Distrib. Syst.*, vol. 4, no. 3, pp. 328–346, 1993.

[64] T. Hoefler and M. Snir, “Generic topology mapping strategies for large-scale parallel architectures,” in *Proceedings of the international conference on Supercomputing - ICS ’11*, 2011.

[65] Bhuyan and Agrawal, “Design and Performance of Generalized Interconnection Networks,”

IEEE Trans. Comput., vol. C-32, no. 12, pp. 1081–1090, Dec. 1983.

[66] A. Bhatele *et al.*, “Mapping applications with collectives over sub-communicators on torus networks,” in *International Conference for High Performance Computing, Networking, Storage and Analysis, SC*, 2012.

[67] W. Gropp, E. Lusk, N. Doss, and A. Skjellum, “A high-performance, portable implementation of the MPI message passing interface standard,” *Parallel Comput.*, 1996.

[68] D. Kumar and V. K. Sehgal, “Center Concentrated X-Torus Topology,” *J. Telecommun. Electron. Comput. Eng.*, vol. 9, no. 3, pp. 55–60, 2017.

[69] P. Architectures, “Introduction to MPI,” *Introd. to MPIs*, 2001.

[70] A. Asadi, Q. Wang, and V. Mancuso, “A survey on device-to-device communication in cellular networks,” *IEEE Commun. Surv. Tutorials*, 2014.

[71] Y. N. Patt, S. J. Patel, M. Evers, D. H. Friendly, and J. Stark, “One billion transistors, one uniprocessor, one chip,” *Computer (Long. Beach. Calif).*, 1997.

[72] T. Opsahl, F. Agneessens, and J. Skvoretz, “Node centrality in weighted networks: Generalizing degree and shortest paths,” *Soc. Networks*, 2010.

[73] M. O. Jackson, “Average distance, diameter, and clustering in social networks with homophily,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2008.

[74] J. Han, D. Watson, and F. Jahanian, “Topology aware overlay networks,” in *Proceedings - IEEE INFOCOM*, 2005.

[75] R. Teixeira, K. Marzullo, S. Savage, and G. M. Voelker, “In search of path diversity in ISP

networks," in *Proceedings of the conference on Internet measurement conference - IMC '03*, 2003.

[76] Yulu 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, Jul. 2001.