

# **Comparative Study of Topologies in 3D NoC**

Project report submitted in partial fulfillment of the requirement  
for the degree of

**Master of Technology**  
**in**  
**Computer Science and Engineering**

By

**Suchi Johari (132208)**

Under the Supervision of

**Dr. Vivek Kumar Sehgal**



**Department of Computer Science and Engineering**  
**Jaypee University of Information and Technology**  
**Waknaghat, Solan – 173234, Himachal Pradesh, India**

# Certificate

This is to certify that project report entitled **Comparative Study of Topologies in 3D NoC**, submitted by **Suchi Johari** in partial fulfillment for the award of degree of Master of Technology in Computer Science and Engineering to Jaypee University of Information Technology, Wagnaghat, Solan has been carried out under my supervision.

This work has not been submitted partially or fully to any other University or Institute for the award of this or any other degree or diploma.

DATE:

SUPERVISOR:

**Dr.Vivek Kumar Sehgal**

**Associate Professor**

**Department of CSE and IT**

# Acknowledgements

I **Suchi Johari** would like to thank **Prof. Dr. RMK Sinha (Dean CSE and IT)** and **Prof. Dr. S.P Ghrera (Head, Dept. of CSE)** for all the support and guidance they have provided to me during my M.Tech programme. I am thankful for his continuous motivation and encouragement provided to me at every point in time.

I would like to thank **Dr. Vivek Kumar Sehgal (Associate professor, Department of CSE and IT)** for offering me the opportunity to do my post graduation project under his supervision at Jaypee University of Information Technology. In the meetings and discussions, he always gave me right advice and he directed my research in the right direction.

I would also like to thank **Dr. Pardeep Kumar (Associate M.Tech Research Coordinator)** for his valuable guidance and motivation. I am thankful to my seniors and friends for their support and help provided during the completion of this thesis. I would sincerely like to thank all the librarian for their support and help by providing me the study material.

Finally I would like to express my profound thanks to my parents for teaching me how to soar. I would not have made it this so far without their unbounded love, guidance, support and most importantly their prayers.

DATE:

SUCHI JOHARI(132208)  
**M.Tech C.S.E. 2<sup>nd</sup>yr**

# Declaration

I hereby declare that the thesis entitled "**Comparative Study of Topologies in 3D NoC**" submitted by me for the award of degree of Master of Technology in Computer Science and Engineering, Jaypee University of Information Technology, Waknaghat is original and it has not submitted previously to this or any other university for any degree or diploma. Signature of

Student :

Name of Student :

Date :



# Abstract

Networks on Chip is a communication subsystem on an integrated circuit (commonly known as "chip"). As the number of cores and IP blocks integrated on a single chip are increasing day by day, there is a need to design such a topology which proves to be best for communication, satisfying all the quality of service (QoS) parameters such as latency, throughput, link-utilization, loss probability and energy consumption at constant bandwidth required. Till now the topologies are optimized on the basis of cost and design space. Now the target is to minimize the global communication traffic, along with the cost and space. Throughput, latency and energy consumption are the basic parameters to consider communication between modules. Shorter the communication path, lesser is the latency. Power consumption also depends upon the length of wires and data activity. For this reason, 3D Networks-on-Chip is gaining more popularity as compared to 2D Networks-on-Chip. In order to reduce space and cost of the chip, Networks-on-Chip tries to reduce the number of connections. But reduction in number of connections lead to lack in functionality during traffic and noise, as lesser the number of communication path between any two nodes more chances of bottleneck, leading to the lack of performance. Keeping all these issues in mind, the main concern is to reduce communication delay keeping size and cost constant. The aim of this research work is to do comparative studies of all the topologies being developed for 3D networks-on-chip, and designing a full fledged new topology which proves to be better than the existing ones in all the senses. This report summarizes the concepts of the existing topologies for the on chip networks and gives overview of how different topologies are beneficial over the other.

# Contents

<b>Certificate</b>	<b>i</b>
<b>Acknowledgements</b>	<b>ii</b>
<b>Declaration</b>	<b>iii</b>
<b>Abstract</b>	<b>iv</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Why 3D NoCs? . . . . .	2
1.2 Flit size or Phit Size . . . . .	4
1.3 Routing . . . . .	5
1.3.1 Distributed Routing . . . . .	5
1.3.2 Source Routing . . . . .	5
1.3.3 Hybrid Routing . . . . .	6
1.3.4 Centralized Routing . . . . .	6
1.4 Implementation of Routing Algorithm . . . . .	6

1.4.1	Deterministic routing algorithms . . . . .	6
1.4.2	Oblivious routing algorithms . . . . .	7
1.4.3	Adaptive routing algorithms . . . . .	7
1.5	Switching . . . . .	7
1.5.1	Circuit Switching . . . . .	7
1.5.2	Store-and-forward (SF) switching or Packet Switching . . . . .	8
1.5.3	Virtual cut-through (VCT) switching . . . . .	8
1.5.4	Wormhole (WH) switching . . . . .	9
1.6	Router Architectures . . . . .	12
<b>2</b>	<b>Literature Review</b>	<b>14</b>
2.1	Research Papers . . . . .	14
2.1.1	Key Research Problems in NoC Design: A Holistic Perspective . . . . .	14
2.1.2	3D Network-on-Chip Architectures Using Homogeneous Meshes and Heterogeneous Floorplans [43] . . . . .	15
2.1.3	Topology Optimization for Application-Specific Networks-on-Chip . . . . .	17
2.1.4	3-D Topologies for Networks-on-Chip [45] . . . . .	17
2.1.5	Energy and Latency Evaluation of NoC Topologies [46] . . . . .	18
2.1.6	Throughput-Oriented NoC Topology Generation and Analysis for High Performance SoCs [47] . . . . .	19
<b>3</b>	<b>Problem Statement</b>	<b>22</b>
3.1	Problem Formulation . . . . .	22

---

<b>4</b>	<b>Proposed Approach</b>	<b>23</b>
4.1	Cluster Formation . . . . .	24
4.2	Structure of Router and Core for implementation of CBCT . . . . .	26
4.3	Master based Routing Algorithm . . . . .	30
4.4	CBCT : Communication based Cluster Topology . . . . .	30
4.5	3D CBCT topologies for NoC . . . . .	36
4.6	Quality of Services . . . . .	40
<b>5</b>	<b>Simulator Tool</b>	<b>45</b>
5.1	OMNET++ Simulator . . . . .	45
5.2	Orion 2.0 Simulator . . . . .	46
<b>6</b>	<b>Design</b>	<b>48</b>
<b>7</b>	<b>Experimental Results</b>	<b>54</b>
7.1	Experimental results of 3D NoC CBCT topology . . . . .	64
7.1.1	Results of 3D Mesh topology . . . . .	64
7.1.2	Results of 3D Hybrid topology . . . . .	64
7.1.3	Results of 3D Layered topology . . . . .	67
7.1.4	Comparison of Results . . . . .	67
7.1.5	Results for energy consumption . . . . .	74
<b>8</b>	<b>Conclusion</b>	<b>80</b>

---

<b>9 Future Work</b>	<b>81</b>
<b>Bibliography</b>	<b>82</b>
<b>List of Publications</b>	<b>86</b>

# List of Figures

1.1	Shared Bus Architecture . . . . .	2
1.2	Point to Point Architecture . . . . .	3
1.3	Networks on Chip Architecture . . . . .	3
1.4	Phit and Flit . . . . .	5
1.5	Circuit Switching . . . . .	8
1.6	Store-and-forward (SF) switching or Packet Switching . . . . .	9
1.7	Virtual cut-through switching . . . . .	10
1.8	Wormhole switching . . . . .	11
1.9	Router Architecture . . . . .	13
2.1	Floorplans showing the combined use of homogeneous floorplan and heterogeneous floorplans. . . . .	16
2.2	Various NoC Topologies . . . . .	18
2.3	ACP and CRG . . . . .	19
2.4	Example of switch merger. . . . .	20
2.5	Example of switch partitioning process. . . . .	20

---

2.6	Topology generation Algorithm. . . . .	21
4.1	Communication between Cores . . . . .	26
4.2	Intra-cluster and inter-cluster communication in CBCT . . . . .	27
4.3	Topology based cluster graph . . . . .	27
4.4	Flowchart for generating CBCT topology . . . . .	28
4.5	Structure of 2D router used for CBCT . . . . .	29
4.6	Structure of 2D core used for CBCT . . . . .	29
4.7	5X5 2D Mesh Topology . . . . .	32
4.8	Cluster of most communicating cores . . . . .	33
4.9	Communication based Cluster topology . . . . .	34
4.10	Flowchart for creating CBCT topology and routing packets from source to destination . . . . .	35
4.11	5X5 2D Mesh Topology . . . . .	37
4.12	Heterogeneous and hybrid Clustered topology . . . . .	38
4.13	Clustered topology geometrically equivalent to mesh topology . . . . .	39
5.1	Flow chart representation of execution of steps in OMNET++ . . . . .	46
6.1	Design of 5X5 Mesh Topology . . . . .	49
6.2	Design of Cluster based Hybrid Topology . . . . .	50
6.3	5X5X4 (100 processing elements) 3D Mesh . . . . .	51
6.4	Hybrid CBCT topology . . . . .	52

---

---

6.5	Layered CBCT topology . . . . .	53
7.1	End-to-End Latency . . . . .	57
7.2	Network Latency . . . . .	58
7.3	Packet Latency . . . . .	58
7.4	Sink Bandwidth . . . . .	59
7.5	Link Utilization . . . . .	59
7.6	Energy consumption on core $c_4$ . . . . .	61
7.7	Energy consumption on core $c_9$ . . . . .	61
7.8	Energy consumption on core $c_{14}$ . . . . .	62
7.9	Energy consumption on core $c_{17}$ . . . . .	62
7.10	Energy consumption on core $c_{19}$ . . . . .	63
7.11	Energy consumption on core $c_{24}$ . . . . .	63
7.12	End to End Latency in 3D Mesh Topology . . . . .	64
7.13	Network Latency in 3D Mesh Topology . . . . .	65
7.14	Packet Latency in 3D Mesh Topology . . . . .	65
7.15	Sink Bandwidth in 3D Mesh Topology . . . . .	66
7.16	Link Utilization in 3D Mesh Topology . . . . .	66
7.17	End to End Latency in 3D hybrid CBCT . . . . .	67
7.18	Network Latency in 3D hybrid CBCT . . . . .	68
7.19	Packet Latency in 3D hybrid CBCT . . . . .	68



7.20 Sink Bandwidth in 3D hybrid CBCT . . . . .	69
7.21 Link Utilization in 3D hybrid CBCT . . . . .	69
7.22 End to End Latency in 3D layered CBCT . . . . .	70
7.23 Network Latency in 3D 3D layered CBCT . . . . .	70
7.24 Packet Latency in 3D layered CBCT . . . . .	71
7.25 Sink Bandwidth in 3D layered CBCT . . . . .	71
7.26 Link Utilization in 3D layered CBCT . . . . .	72
7.27 Comparison of End-to-End Latency . . . . .	72
7.28 Comparison of Network Latency . . . . .	73
7.29 Comparison of Packet Latency . . . . .	73
7.30 Comparison of Sink Bandwidth . . . . .	74
7.31 Comparison of Link Utilization . . . . .	75
7.32 Comparison of energy consumption (in pJ) in $c_4$ core (a) 3D Mesh (b) 3D Cluster based Hybrid Topology(clustering of layers of Mesh Topology) . . . . .	75
7.33 Comparison of energy consumption (in pJ) in $c_{57}$ core (a) 3D Mesh (b) 3D Cluster based Hybrid Topology(clustering of layers of Mesh Topology) . . . . .	76
7.34 Comparison of energy consumption (in pJ) in $c_{88}$ core (a) 3D Mesh (b) 3D Cluster based Hybrid Topology(clustering of layers of Mesh Topology) . . . . .	76
7.35 Comparison of energy consumption (in pJ) in $c_{95}$ core (a) 3D Mesh (b) 3D Cluster based Hybrid Topology(clustering of layers of Mesh Topology) . . . . .	77
7.36 Comparison of energy consumption (in pJ) in $c_4$ core (a) 3D Mesh (b) 3D Cluster based Layered Topology(clustering of layers of Mesh Topology) . . . . .	77

- 
- 7.37 Comparison of energy consumption (in pJ) in  $c_9$  core (a) 3D Mesh (b) 3D  
Cluster based Layered Topology(clustering of layers of Mesh Topology) . . . . 78
- 7.38 Comparison of energy consumption (in pJ) in  $c_{12}$  core (a) 3D Mesh (b) 3D  
Cluster based Layered Topology(clustering of layers of Mesh Topology) . . . . 78
- 7.39 Comparison of energy consumption (in pJ) in  $c_{19}$  core (a) 3D Mesh (b) 3D  
Cluster based Layered Topology(clustering of layers of Mesh Topology) . . . . 79

# List of Tables

7.1	Parameters considered for simulation of mesh topology and CBCT. . . . .	55
7.2	Parameters considered for calculation of Energy Consumption in mesh topology and CBCT. . . . .	55
7.3	Simulation details for mesh topology and CBCT. . . . .	56
7.4	Energy consumption of Link (in pJ). . . . .	60
7.5	Energy consumption of Router (in pJ). . . . .	60

# Chapter 1

## Introduction

System-on-chip has performance issues due to interconnection networks. Basically while concentrating over the issues such as scalability, high bandwidth and low latency system on chip tries to become as smaller as possible by shrinking the size of the chip. But this instead of making thing better, making things appear more worst. As in this the performance degrade due to bottleneck and congestion, basically because of shrinking geometry. If the traditional bus topology is considered, it doesn't prove to be a good alternative due to less scalability provided in this, lack of parallelism, high latency involved, low throughput and more energy consumption along with high power dissipation. For providing the best solution to the existing system-on-chip NoC proves to be the best solution. In NoC higher scalability is provided along with the concurrency of execution of multiple tasks by processors working concurrently. And hence Network-on-chip can be considered as a solution to all the problems being faced in the System-on-chip architecture, and these problems are treated at nano scale level in Networks-on-chip [1, 2, 3, 4]. In last few years a large number of changes has taken place in the Networks-on-chip architecture such as mesh topology, torus, fat tree, etc and also on other application specific architectures [5, 6, 7, 8, 9, 10, 11, 12].

But instead of providing so many solutions over the system-on-chip architecture, networks-on-chip still is not a perfect alternative due to some limitations being faced by networks-on-chip. The limitations faced by networks-on-chip are high power consumption, large amount of communication cost, and a lower throughput. To deal with these problems a new concept of 3D networks-on-chip architecture came into existence, with low power consumption along with higher speed.

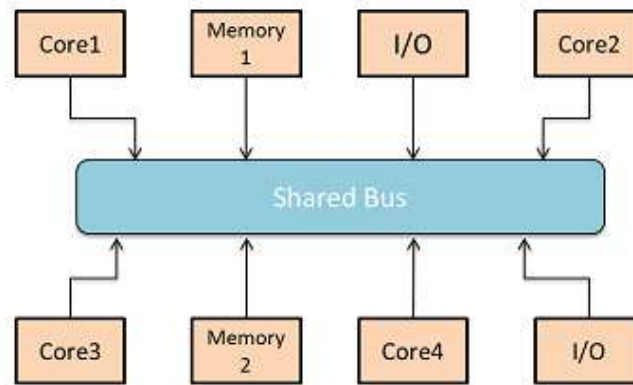


Figure 1.1: Shared Bus Architecture

According to Moore's law the number of transistors are increasing day by day, so to accommodate such a large number of transistors on a single chip, it is required to shrink the geometry of the chip while performance should not be effected negatively. Due to scalability SoC can grow more and more continuously, due to this there is occurrence of many problems such as power dissipation, latency, overheat, management of resources, etc[13, 14]. Interconnection network if considered perform an important role in the performance and power consumption of the chip[15]. Now due to so many problems faced in traditional topologies such as scalability issues and lack of parallelism and concurrency, bus topology and P2P architecture are of no use today[13, 16]. The scalable and parallel connections provided, NoC basically provides an connection between the processors, memory and also provides a facility of custom design. NoC basically provides hop-by-hop communication with the help of the packet switching. Fig.1.1(a) and Fig.1.1(b) show one of the most well-known architectures which are respectively Point-to-Point (P2P) and shared bus systems. As shown in Fig.1.2, NoC architecture connects switch and wires, hence it makes use of the combination of the traditionally build bus and point to point architecture, providing advantages of bot and hence reducing the disadvantages. The basic disadvantages removed using network on chip are large number of long wires as in case of P2P architecture and scalability issues as occurred in shared bus architecture.

## 1.1 Why 3D NoCs?

As the development in the technology the future applications are becoming complex day by day. Due to their complexity they are demanding for the more bandwidth for transaction between memories and cores or for providing communication between the cores of the chip. These

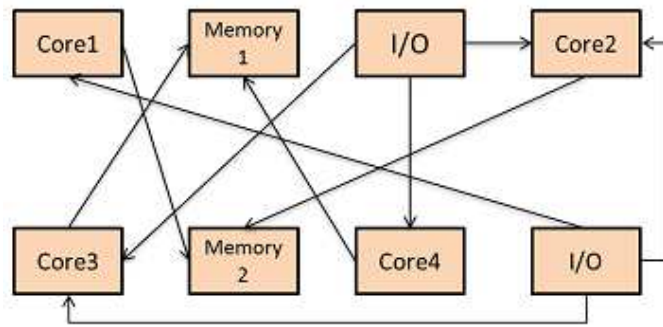


Figure 1.2: Point to Point Architecture

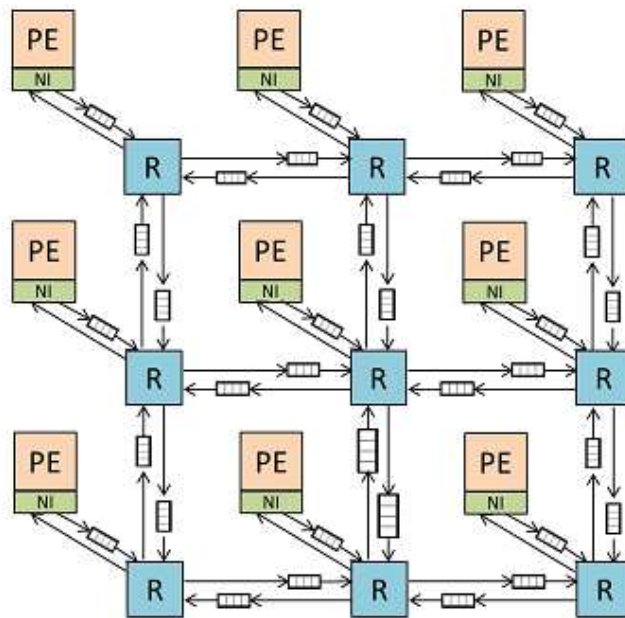


Figure 1.3: Networks on Chip Architecture

factors lead to 2D NoC failure and hence the future is based on the more scalable NoCs such as they are so large that they can accommodate hundreds of cores. 2D NoC faces a high dimension of problems hence an introduction to the 3D NoC has proved to be the best alternative for all problems. 2D NoC could not support high diameter but it can easily be supported by the 3D NoC [17, 18]. Diameter can be defined as the number of hops that a flit, packet or message takes for being transferred from source to the destination. If large distance is covered by the flit to traverse from the source to the destination, the latency due to communication will be very high, due to which throughput is very low. The best solution to this problem is that instead of using 2D NoC topology switch to the 3D NoC architecture. Since there is a third dimension in 3D NoC architecture so the density of 3D is much higher than as compared to the 2D NoC topology architecture. Due to the reduced number of wiring the lower interconnection power is required. The reduction in power is basically due to the reduction in the number of hops, as the flit has to cover a lesser number of hops to reach from source to the destination. Due to less hops lesser amount of buffer access is required along with less switch arbitration and less number of interconnection links leading to less traversal. The results of these factors will lead to the decreases in power consumption.

## 1.2 Flit size or Phit Size

- **Flit:** It is the smaller unit being extracted from information at the link layer, and the information consists of one or several words. There are several types of flit, and it requires several cycles to exchange flits [19, 20].
- **Phit:** At the physical layer the smallest unit of information is basically known as phit, which requires one cycle for transferring of flit over physical layer. Fig.1.3 shows how message is divided into packet which are further split into flit which is a combination of much smaller units called phit [19, 20].

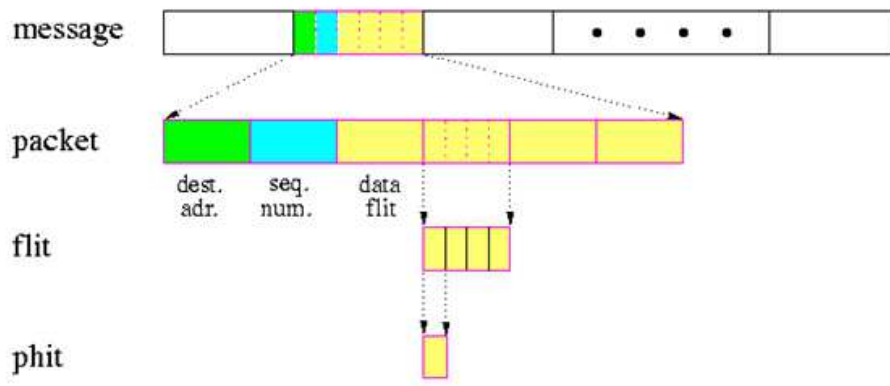


Figure 1.4: Phit and Flit

## 1.3 Routing

### 1.3.1 Distributed Routing

For passing the traffic over the network between routers and nodes there is a requirement of some kind of a routing algorithm which can decide the path followed by the packet or flit to pass from source to the destination [21, 22, 23, 24]. Header associated with packet just contains the address of the destination node. Each router have the information about the address of the neighboring routers. When packet reach to some intermediate router then it is passed to the other router whose address is there in the table of the intermediate router. Passing from router to router and following the path according to the routing table, message is passed from source node to the destination node. If we talk about this routing approach, then it is most favorable in the symmetric and regular topologies, because of the reason that all the routers use same routing algorithm.

### 1.3.2 Source Routing

In this routing algorithm source node will calculate the full routing path to the destination in advance [21, 22, 23, 24]. This path is pre-calculated by the source node even before injecting the packet in the network. Now since the path is pre-calculated so it will be embedded with the header and hence all the routers in the path will just read the header and will calculate the path to reach the destination. So this technique instead of calculating the routing path at each node



partially, the full path is determined and is send along with the packet in the header.

### **1.3.3 Hybrid Routing**

This routing algorithm can be considered as the combination of the distributed and source routing algorithms. In this instead of calculating the full path from the source node to the destination, only the path to some of the intermediate nodes is pre-calculated. Then with the help of these intermediate paths and the paths from the routing table of each node the precise path between the routers are calculated to reach the destination node. As full path is not calculated at once, and the path is calculated in distributed manner and hence it is called distributed routing algorithm.

### **1.3.4 Centralized Routing**

In this the whole routing algorithm relies on the controller. This controller is responsible basically for determining the path from source to the destination node to pass the packet.

## **1.4 Implementation of Routing Algorithm**

### **1.4.1 Deterministic routing algorithms**

In this routing algorithm there would be a routing path which is dedicated and will be used to pass the packet from source to the destination [25, 26]. Every time the packet has to follow the same path between the source and the destination and hence a dedicated path is allotted for transferring packet from source to the destination. It is most simple as only one dedicated path has to be determined for routing packets, and hence faster too as not much calculations required.

### 1.4.2 Oblivious routing algorithms

In this routing algorithm the determination of the routing path between the two nodes is basic concern and what is happening all around is not need to be considered. S in this routing algorithm no extra information is required to be maintained. So here only the address of the node is considered and other details are of no use. So the information such as the congestion details are not maintained here. Due to this traffic problems such as congestion cannot be resolved.

### 1.4.3 Adaptive routing algorithms

In this basic concentration is on the avoidance of the congestion. For reducing congestion, in this algorithm the full information about the network traffic is concerned [27, 28]. This information include the details such as network traffic, channel status, faulty regions etc. This technique can be applied and best suitable for the conditions where the traffic status is constant and doesn't change too fast or very frequently. If the traffic pattern change too frequently then it would be costlier to monitor them. Here the network congestion on every intermediate node is calculated. During the occurrence of congestion alternate path are followed which will provide better throughput and lower latency. But here the problem of message arrival out of order can be faced very frequently.

## 1.5 Switching

### 1.5.1 Circuit Switching

This is the switching algorithm which is basically consisting of two phases which include the circuit establishment phase and the message transmission phase. There is a probe message which will go towards destination node, via a reserved physical links [23, 29, 30]. These physical links are reserved as the probe passes through the intermediate routers (see Fig.1.4(a)). Once the probe has reached to the destination the full path is set up. After the establishment of the path a flit is sent back over this path to acknowledge that the path between source and destination are set. When the acknowledgment is received on the sender's side then the sender is capable of transmitting complete message at the full bandwidth. The path forms a dedicated circuit that is reserved for full time till message transmission take place (see Fig.1.4(c)).

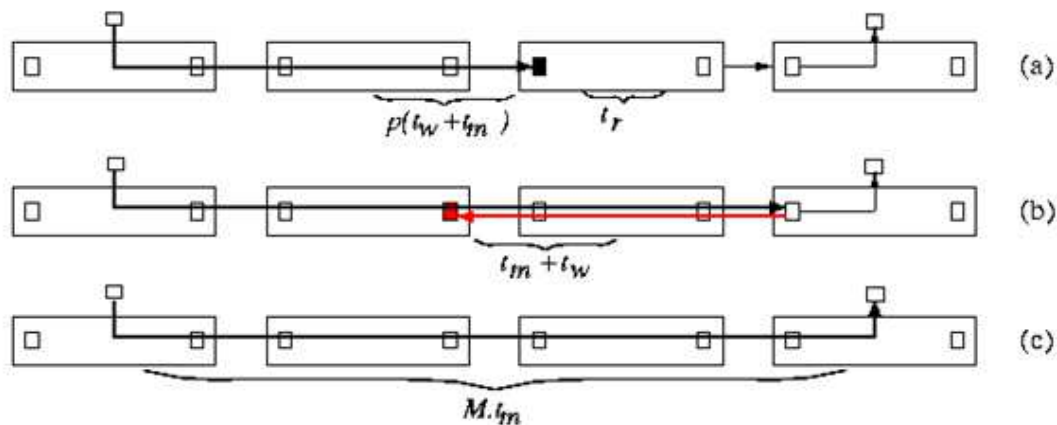


Figure 1.5: Circuit Switching

### 1.5.2 Store-and-forward (SF) switching or Packet Switching

Here in this algorithm the message is split into different fixed length packets. These packets consist of flits, along with the header flit [23, 31, 32]. There are input and output buffers for each distinct packet over the network channel. From source to the destination every packet is treated as an unique identity and hence routed individually. The steps termed in the stop and forward switching is called a hop. In this the packet is copied from the output buffer of one node to the input buffer of the other node. Once the full packet is buffered in the input buffer of the node then the node is responsible for further routing decision. In Fig.1.5 Store-and-forward switching of a packet is shown. Fig.1.5(a) Routing decision for the first router is being considered. Fig.1.5(b) shows the packet taking the first hop, after being copied to output buffer, to the second router. Fig.1.5(c) shows the structure and conditions of the whole packet after occurrence of the second hop.

### 1.5.3 Virtual cut-through (VCT) switching

This technique is the most costlier techniques over the other techniques discussed till now. Like in store and forward here also the message is broken down into packets and there buffers present in each router to store these packets [23, 33]. Header flit will be removed from the packet at the next router as instant as the routing decision is taken, conditioned output channel should be free. Whenever the new flit reaches the router it is being buffered. And it is further passed to the immediate neighboring router if the output channel is free. If the packet cannot be further

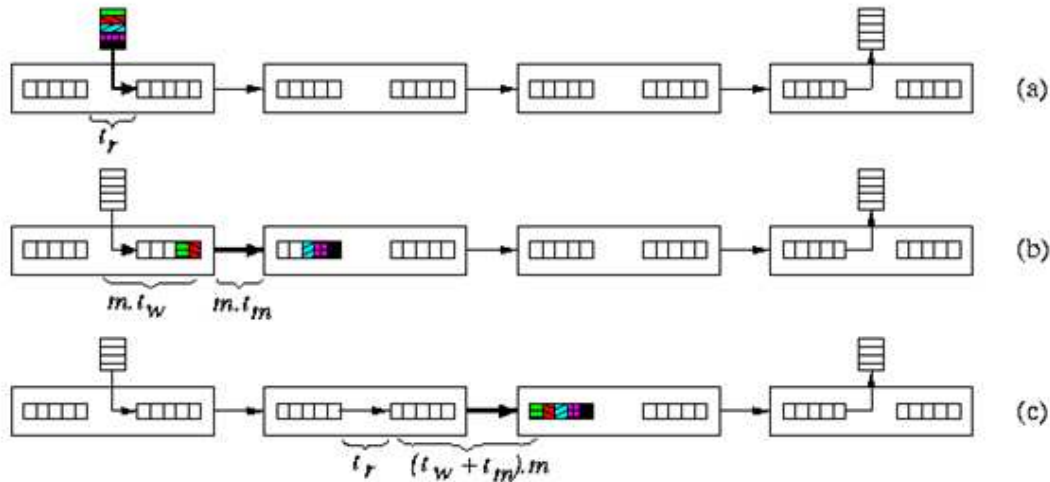


Figure 1.6: Store-and-forward (SF) switching or Packet Switching

processed then it will keep on waiting in the current router for its chance to come, till the other flits not releasing the channels occupied by them. But if the channel is free and there is no congestion then the packet will be divided into flits and these flits will pass over the channel in the continuous flow. Till then all the other buffers in the routing path are blocked to take over more packets. In Fig.1.6 Virtual cut-through switching of a packet. Fig.1.6(a) shows the moving of the header flit to the output buffer of the first packet. Fig.1.6(b) shows the distribution of the header over the second router whereas the subsequent flits will continue to follow their paths. Fig.1.6(c) shows that even after pulling the flits in a chain the header is cut through, and its output buffer is reserved for further communication. Fig.1.6(d) gives details that how the chained packets are contracted to form the message again and then the packet will be buffered in the first router. While doing this all the previously allocated links will be released. Fig.1.6(e) shows the pipe-lining of the flits for moving towards the destination node.

#### 1.5.4 Wormhole (WH) switching

Like conflict-free virtual circuit switching, the packets are split into flits and are pipelines then passed over the route. In this algorithm also different packets cannot be interleaved or multiplexed on the channel without providing it an additional support for architecture [23, 34, 35]. The major difference between the two techniques is that in this there is no concept of buffers for the packets. Still there is a provision of buffers and here every router has a small size of buffer to store one or a very few number of flits. Header flit builds or initialize a path from source to

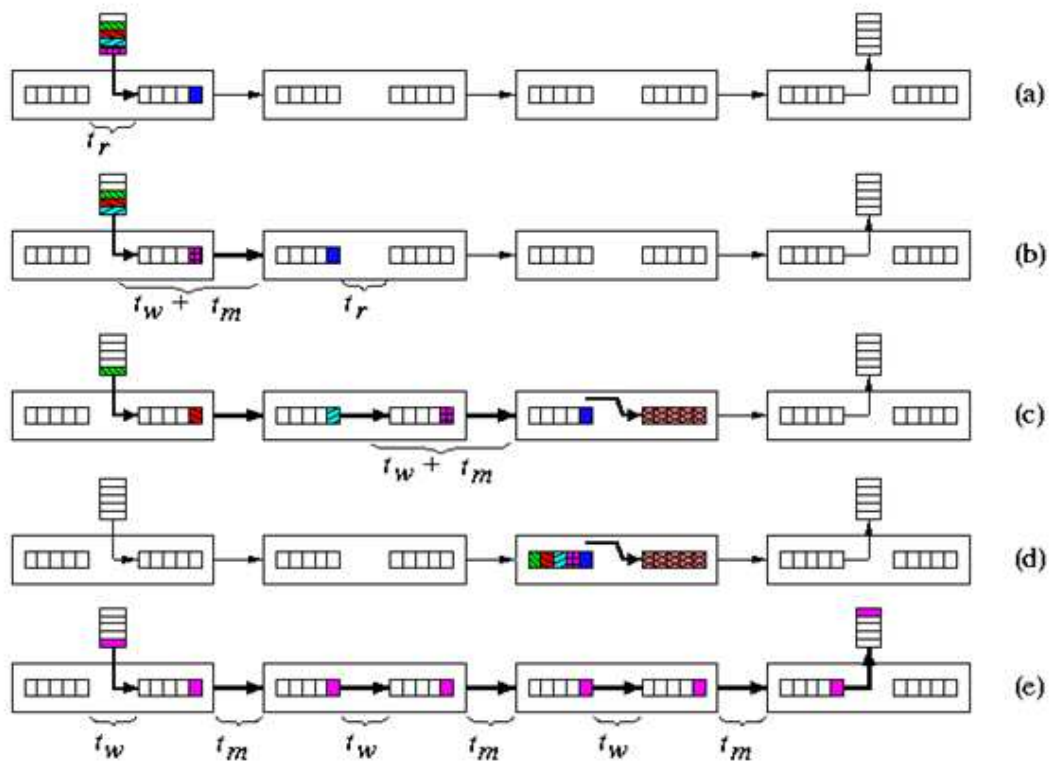


Figure 1.7: Virtual cut-through switching

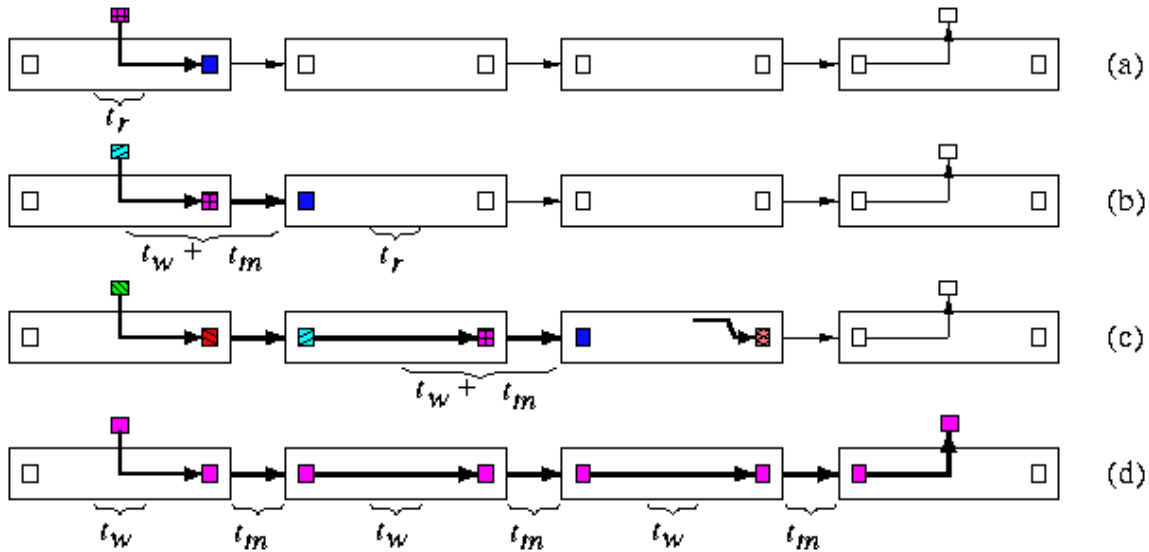


Figure 1.8: Wormhole switching

the destination which is being followed from source to the destination node in a pipeline manner. Here in this a wormhole is formed by the sequence of the buffers and the links. Number of flits in the packet determine the length of the path as they are directly proportional to each other. Whole path is being traversed between the source and the destination. If the channel has congestion due to traffic and the header cannot be processed due to the busy output channel, the whole chain of flits will get stalled. To resolve this issue the flits should occupy the buffers in the routers blocking the other type of communication taking place. Conflict-free communication latency  $t_{WH}$  produces same results as in case of the VCT switching. Due to this it is proportional to the summation including the path length and size of the packet. In Fig.1.7 explanation of how wormhole switching of a packet works is given in a diagrammatic context. Fig.1.7(a) shows the copying of the header to the output buffer once the routing decision is taken. Fig.1.7(b) shows the transferring of the header flit to the second router along with the other flits following it. Fig.1.7(c) When the header flit is arrived into a router whose output channel is busy, whole flits in the path will get stalled, and all channels will be blocked. Fig.1.7(d) shows that how the flits are pipelined in the case of the conflict-free routing algorithm by the establishment of the wormhole switching across the routers.

## 1.6 Router Architectures

Traditionally used 7X7 3D router which was based on the shared bus architecture for communication interface between different layers are modified and an introduction to the 3D NoC bus based hybrid topology [36, 37, 38, 39, 40, 41]. Due to this the number of ports are reduced from 7 to 6, but still there is an issue as the flits passing from one layer to other has to fight for accessing the shared bus, as it is only responsible for the communication between the layers. High traffic and network congestion will lead to an undesirable performance of the system. If we talk about the 2D router architecture then there are 5 ports which include NORTH, SOUTH, EAST, WEST, LOCAL. If the 3D router architecture is considered then there are 7 ports among which 5 ports are same as mentioned above with two extra ports UP and DOWN ports as shown in Fig. 1.8. From the figure it is clearly visible that inter-layer channels are used to provide linking between the inter-layer channel. But the switches used in different layers are connected to each other using intra-layer links. And each router or switch is connected to a single processing element.

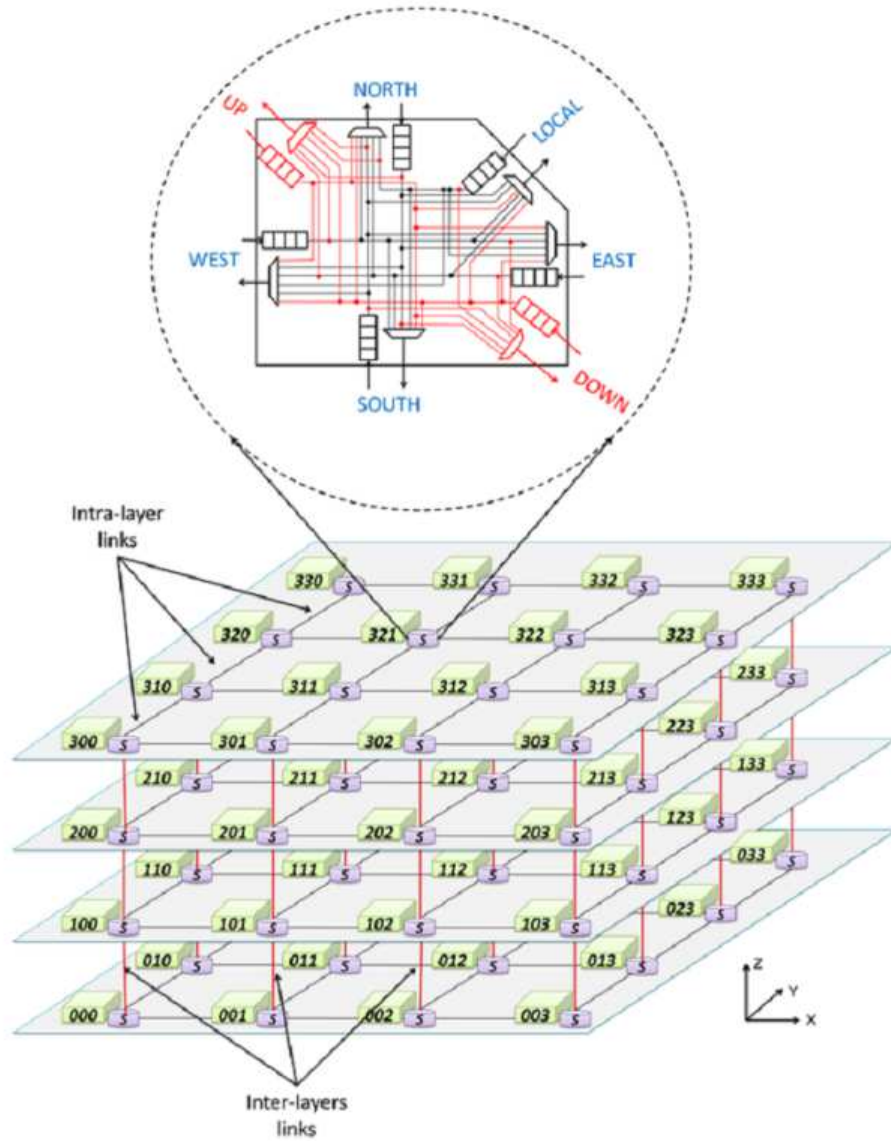


Figure 1.9: Router Architecture



# Chapter 2

## Literature Review

### 2.1 Research Papers

In this section the basic idea is to give the explanation of some of the papers which are being reviewed by me for reference to generate new topology.

#### 2.1.1 Key Research Problems in NoC Design: A Holistic Perspective

In this paper they have defined a 3D design space that involves issues related to communication infrastructure synthesis, communication paradigm selection and application mapping optimization. In the paper to give an explanation about the application and architecture description some of the graphs are considered which include communication task graph and application characterization graph. In the paper eight problem domains are discussed [42]. These problems include:

- Topology Synthesis Problem
- Channel Width Problem
- Buffer Sizing Problem
- Floorplanning Problem

- Routing Problem
- Switching Problem
- Scheduling Problem
- IP Mapping Problem

The main purpose to read the paper is to get through an idea that what are the fields in which improvement is required. After going through the problems defined in this paper and having read about all these problems i am able to decide what to choose as a problem domain for research work.

### **2.1.2 3D Network-on-Chip Architectures Using Homogeneous Meshes and Heterogeneous Floorplans [43]**

In this paper, they have proposed a new 3D architectures for networks-on-chip which are of the form either 2-layer or 3-layer. For these layers a homogeneous regular mesh topology is used along with some of the heterogeneous topologies for the floorplanning layers. The advantages of heterogeneous floorplans and regular mesh networks are combined and the results obtained are compared here in this paper.

In Fig.2.1(a) initial floorplan with no router is shown while Fig.2.1(b) shows two layer architecture, Fig.2.1(c) shows 3 layer architecture and Fig.2.1(d) shows the 2D implementation for each IP/core for calculation of the area which is required in the interfaces of the network, routers along with the wires used for the physical links.

Two NoC architectures are proposed in the paper which include 3D NoC architectures for 2-layer along with the 3-layer architectures. This architecture make use of the homogeneous network on a different layer and that of the heterogeneous floorplans on the different layers. So for the purpose of attaining flexibility and predicting delay the IP/cores can have an arbitrary size. Designing difficulties are resolved in this approach because of the irregularities in size of IP/core that are basically being addressed by some of the specialized algorithms for routing. Author in this paper has proposed the software framework which is versatile in nature for investigating the advantages of the proposed architectures for 3D NoC. In this the integration of the efficient  $B^*$  Tree-based floorplanner with a NoC simulator which is cycle-accurate for maximum performance is used in the results obtained by experiments.

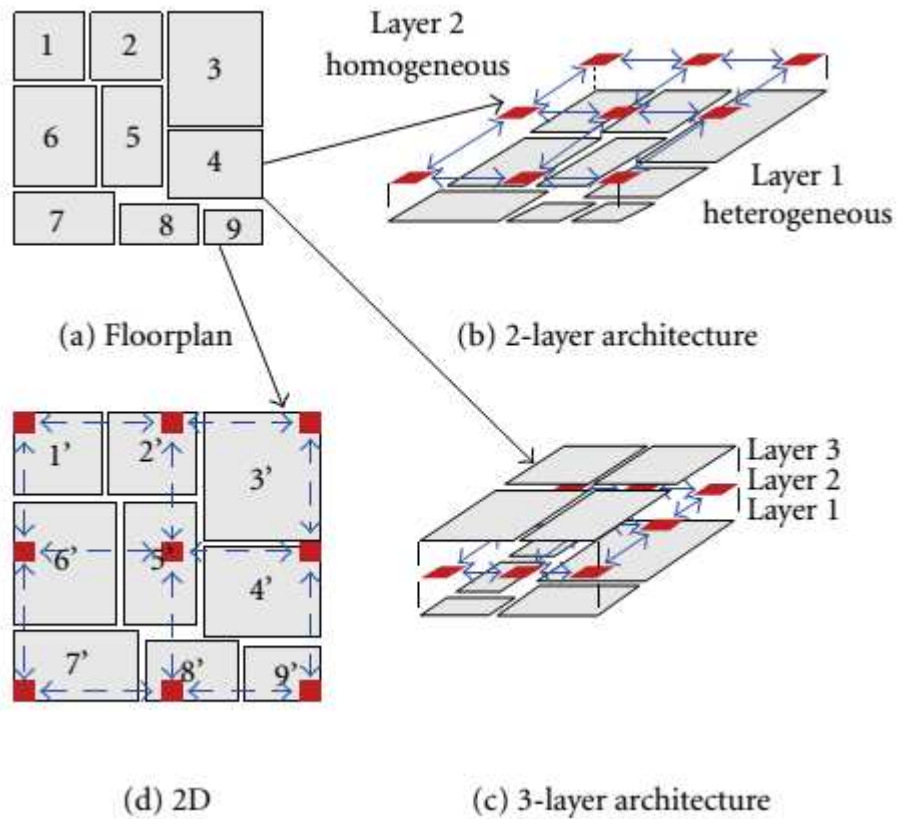


Figure 2.1: Floorplans showing the combined use of homogeneous floorplan and heterogeneous floorplans.

### 2.1.3 Topology Optimization for Application-Specific Networks-on-Chip

This paper has proposed that due to the shrinking of the geometries of circuits the on-chip interconnection networks have become the most prominent for the performance bottleneck. As more functionality is integrated into a single chip, interconnection requirements become complex. Customization aims at satisfying operating speed requirements while meeting the power budget. This is achieved if one can put the interconnections which are very critical as short as possible with the help of partitioning of the system placement of the block in the layout. According to the paper minimization of global communication traffic is concentrated, along with cost and design space. Heavily communicating task should be mapped to the same PE. As shorter the path low is the delay and less power consumption. This paper has also proposed that an Iterative approach should be followed during the development of topologies for 3D NoC. If design fails to meet the particular specification at any point then previous stage is resumed [44].

### 2.1.4 3-D Topologies for Networks-on-Chip [45]

In this paper the author has concentrated upon the those planes which has the property to be stacked vertically and there exist some of the symmetry between the horizontal and vertical communication channels of the network, as the basis for speed and power consumption in the models. This paper has also considered diversity occurring among the number of nodes that is being utilized in the 3D, which reduces the average number of hops traversed by a packet. In this paper it is mentioned that NoC offers high flexibility and also the structure of the network is regular, which is capable of supporting an interconnect models which is simple and can handle a large amount of fault tolerance. The canonical interconnect backbone of the network combined with appropriate communication protocols enhance the flexibility of such systems. They have considered Multidimensional interconnection networks, and these networks are analyzed for multiple constraints, which include a constant bisection-width and the other parameter as pin-out constraints. NoC has no similarity with the generic interconnection networks and there lies a great difference between the two. This paper puts a great emphasis on the Optimization of the topology and also minimize the zero-load latency and plays an important role in the optimization of power consumption in a network. The optimization of topologies greatly depends on the number of parameters used to characterize the router and the channel for communication, for example the number of ports involved in the router, the communication channel length, and the other characteristics or properties of the interconnection networks such as impedance. Fig.2.2 shows the various NoC topologies which include (a) 2-D IC–2-D NoC (b) 2-D IC–3-D NoC (c)

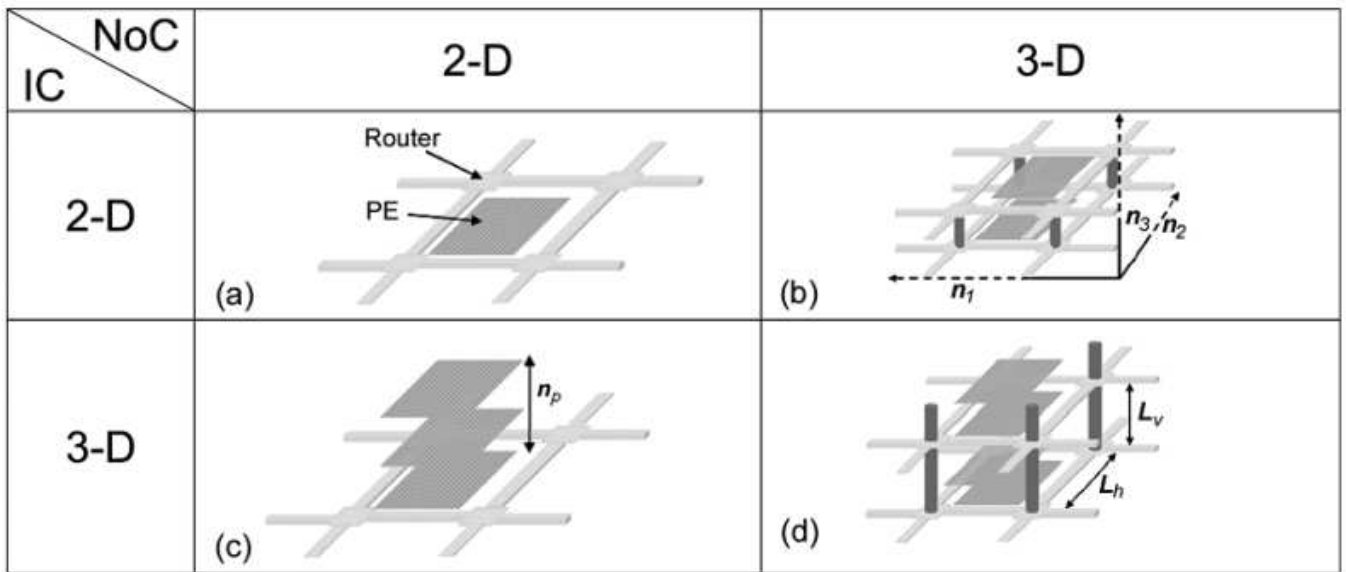


Figure 2.2: Various NoC Topologies

3-D IC-2-D NoC (d) 3-D IC-3-D NoC.

### 2.1.5 Energy and Latency Evaluation of NoC Topologies [46]

This paper concentrates on finding the topology, among regular and irregular NoC topologies, that better fits the application requirements. SUNMAP, this tool selects best topology for given application and generates mapping of core onto that topology. This paper has justified that fat-tree topology is a strongly fulfills the latency constraints for many applications, while mesh topology achieves less energy consumption. In this paper mapping applications onto different networks-on-chip (NoCs) topologies is done such that the mapping of the cores on to the ports of routers, one should consider the requirements such as the latency involved and energy consumption. This paper include two graphs are being used and they include ACP (Application Communication Pattern) graph, CRG (Communication Resource Graph) and CTG (Communication Task Graph).

For each message in the application, it is a necessity to find out a path between the vertexes of CRG which involve sender and the receiver. In order to verify that the bandwidth required by the paths must be similar to the one which is necessary for the application. So in Fig.2.3 ACP and CRG relation ate shown diagrammatically.

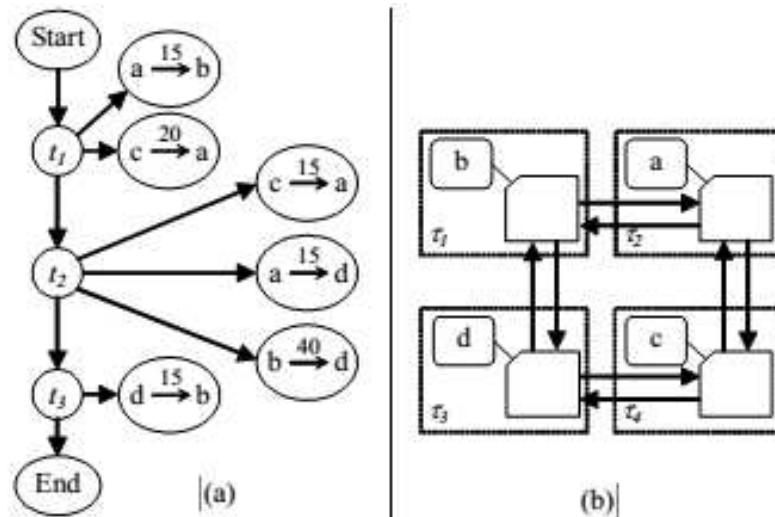


Figure 2.3: ACP and CRG

### 2.1.6 Throughput-Oriented NoC Topology Generation and Analysis for High Performance SoCs [47]

Predictive analysis for estimation of degree of contention is the basic idea of researcher in this paper, instead of performing detailed simulation. This paper has justified that irregular topologies can also provide better performance if correctly built. In this paper packet-switching is considered as an extension of wormhole switching. Topology generation mechanism is carried out in this by using two techniques Switch Merger Technique and switch Partitioning Technique. In Fig.2.4 the switches which have larger number of nodes attached to them and participate in communication to each other the maximum number of times are merged to each other to form a single switch. This merging technique causes the less delay in communication and hence energy and cost are saved as heavily communicating nodes are at the same switch. In Fig.2.5 Suppose a switch has too many number of nodes attached to it is broken into two parts causing the traffic to be distributed over multiple switches and hence controlling congestion. Fig.2.6 shows an algorithmic representation of creating a new topology using iterative approach which is proposed in this paper.

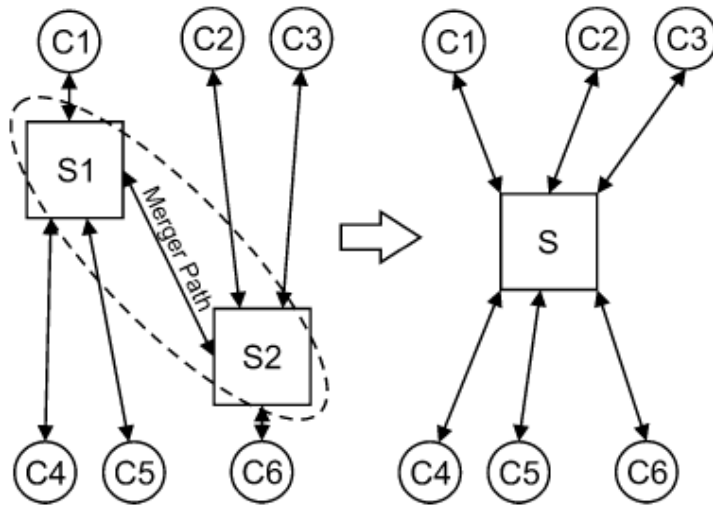


Figure 2.4: Example of switch merger.

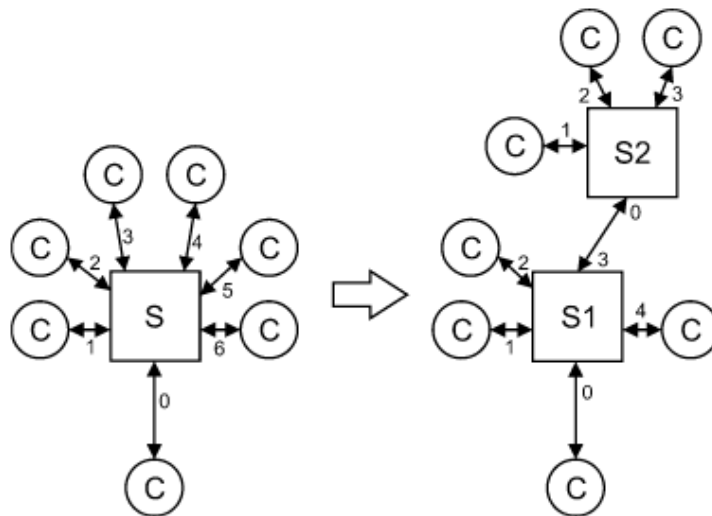


Figure 2.5: Example of switch partitioning process.

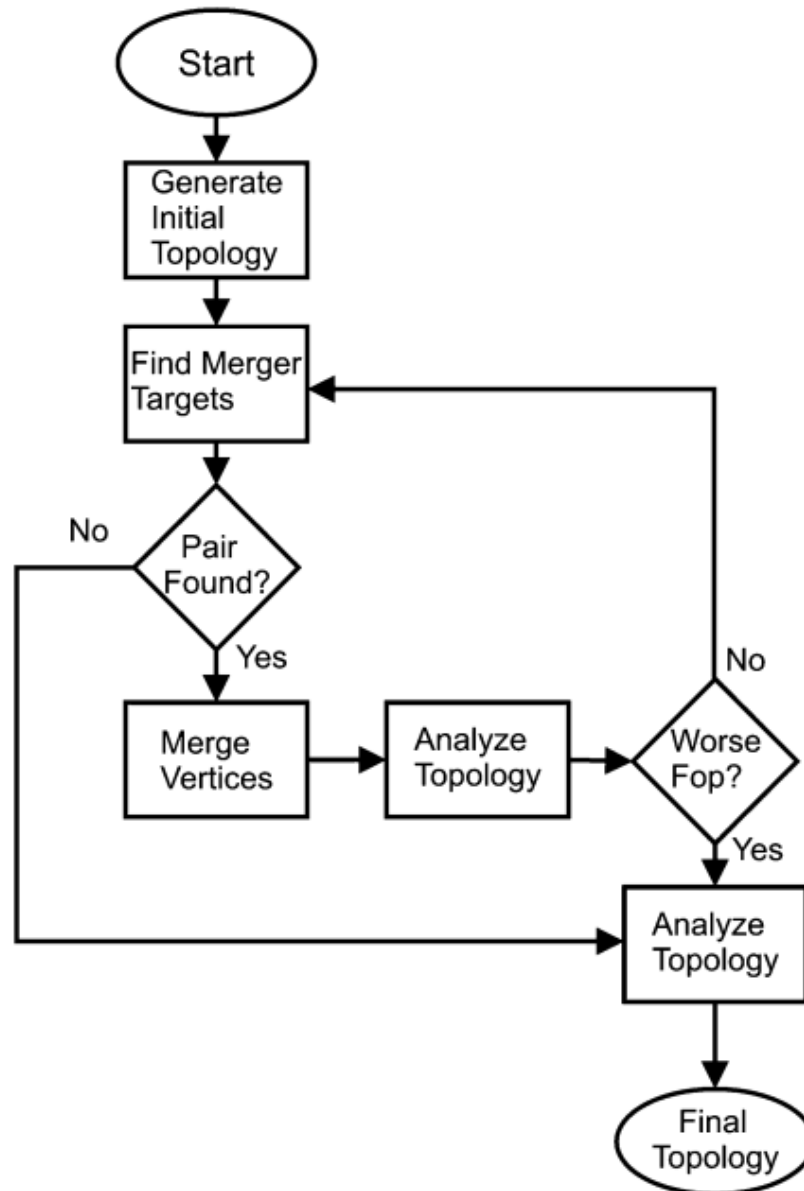


Figure 2.6: Topology generation Algorithm.



# Chapter 3

## Problem Statement

For reduction of design space, communication delay and cost involved, NoC tries to reduce the number of connections. Reduction in the number of communication channels leads to lack in functionality during traffic and noise. Lesser number of channels between any two nodes leads to the situation of bottleneck, due to which performance is also reduced. So the main concern of this research is to reduce the end to end delay, packet latency, network latency and link utilization, keeping area and bandwidth constant. For this purpose, the comparative studies of all the existing NoC topology is done and then providing a new topology which will be better in every perspective, than the already existing one's, in terms of QoS parameters.

### 3.1 Problem Formulation

**Given:** A core communication graph  $G$  (or  $G$ ).

**Find:** The topology generated should optimize  $O(A,G)$ , subject to the constraints specified by  $Const(A,G)$ .

- $O(A,G)$  can be a metric for reliability and performance of the network, such as end-to-end latency, network latency, packet network latency, loss probability, link utilization etc.
- $Const(A,G)$  can be considered as the amount of needed resources (e.g. area, wiring, etc.), cost and bandwidth imposed by the application.

# Chapter 4

## Proposed Approach

First of all, determine the cores having the maximum amount of communication among themselves. Communication among the cores can be determined on the basis of communication task graph. In this graph the cores are considered as the vertices and the links between them determines the communication between the cores. Links are assigned some weights and these weights determines how much communication is there among the cores. Links having maximum weights determine the cores having maximum communication whereas the links having minimum weights determines the cores having least amount of communication. Cores with maximum communication should be in the same cluster so that they take least time for communication, and hence speed up the process. Core communicating lesser amount of times can be in different clusters because if time taken by these cores is more, even then much difference doesn't occur. Here for the simplicity we consider the router and its corresponding core as a node. Among the nodes within a clusters one or more nodes are chosen as the master node. Master node act as an intermediate between the two clusters. It is the node responsible for the inter-core communication. The clusters here can have any of the NoC topology as per the communication requirements among the cores within the cluster. Since single cluster can communicate to the multiple clusters so there can be more than one master node within the cluster. Here if we consider the amount of traffic then the maximum traffic is found to be at the links joining the two master nodes. As this is the single links which is responsible for the inter cluster communication. To reduce the amount of traffic on these links, it is required that some special configuration of these links are required. Configuration of the links means the delay, bandwidth and the type of wire used in the links. In these links varying the bandwidth of the links have made it possible to simulate the required communication among the cores.

## 4.1 Cluster Formation

Cluster in CBCT is formed using the communication based cluster graph (CBCG). CBCG is a directed connected graph which can be represented as  $CBCG = (C, I_l)$  where,  $C$  represents the clusters and  $I_l$  represents the inter-cluster links. In CBCG, the clusters of most communicating cores are formed, the corresponding routers attached to the cores are also considered in the same cluster. Fig.4.1 gives the communication between the cores. The weight of edges describe the amount of communication taking place between the two cores. More the edge weight, more is the communication. According to the amount of communication clusters can be defined, and communication based cluster graph is obtained. The clusters are shown in the Fig.4.2 as  $C_1, C_2, C_3, C_4$  and the links inside each cluster represents the communication between the cores in the cluster. In Fig.4.2, the values associated with the links represent the communication between the cores. These links between the cores of the cluster are called intra-cluster links and the links through which the clusters are connected together are called the inter-cluster links. Cores of two distinct clusters have minimum communication between them. The cores in the CBCG as shown in Fig.4.2 can be connected to each other using any topology within a single cluster such that most communicating cores lie within same cluster. Algorithm 1 gives the details for the formation of cluster. In the algorithm 1, 'N' gives the total number of cores present,  $T_h$  defines the threshold that how many clusters to be formed, and array V stores the details of the cores, already visited during the formation of clusters.

CBCG graph can further be reduced in more generalized graph TBCG.  $TBCG = \langle T, t \rangle$  where, T represents the topology used to arrange the cores within the cluster and t represents the topology used to connect these clusters. In TBCG as shown in Fig.4.3,  $T_i$  represents the topology in cluster  $C_i$  and  $M_i$  represents the master router for each cluster. Master router is responsible for the inter-cluster communication. From each cluster, a master router is chosen such that it provides connectivity to the outside world i.e. other clusters. The topologies used within the cluster can vary for each cluster depending upon the requirements. We are dealing with heterogeneous networks, so for proposed approach, we have considered different topology for each cluster. In Fig.4.3, the clusters are connected to each other using the mesh topology, but any other topology such as star, bus etc. can also be used to connect the different clusters. Flowchart shown in Fig.4.4 describe the steps to be followed for the creation of the CBCT topology.

**Algorithm 1:** Cluster formation for CBCG graph

**Data:**  $N$  = number of cores,  $\{1, \dots, n\}$ ,  $T_h$  = maximum possible clusters,  $V[N]$  = visited cores,  $j=0$ ,  $i=0$ ;

**Result:** Cluster formation

**while** ( $j \neq T_h$ ) **do**

**if**  $i=0$  **then**

**while**  $sizeof(V) \neq N$  **do**

      Let  $(x,y)$  be highest cost edge from  $E$ ;

**if**  $((x \in V) \&\&(y \in V))$  **then**

$E - (x,y)$ ;

        Continue;

**else**

**if**  $x \in V$  **then**

          Assign core  $y$  to the cluster to which  $x$  belongs;

$E - (x,y)$ ;

**else**

**if**  $y \in V$  **then**

            Assign core  $x$  to the cluster to which  $y$  belongs;

$E - (x,y)$ ;

**else**

            Assign both cores  $x$  and  $y$  to new cluster  $C_i$ ;

$E - (x,y)$ ;

$i++$ ;

    Put cores  $x$  and  $y$  in  $V$  whichever is not present in  $V$ ;

**else**

**if**  $j < T_h$  **then**

      Take edge  $(x,y)$  having least cost where  $x,y \in C_i$ ;

      Assign both cores  $x$  and  $y$  to new cluster  $C_i$ ;

$i++$ ;

**else**

**if**  $j > T_h$  **then**

        Take edge  $(x,y)$  having highest cost where  $x \in C_i$  and  $y \in C_j$ ;

        Merge the clusters  $C_i$  and  $C_j$ ;

$i-$ ;

$j=i$ ;

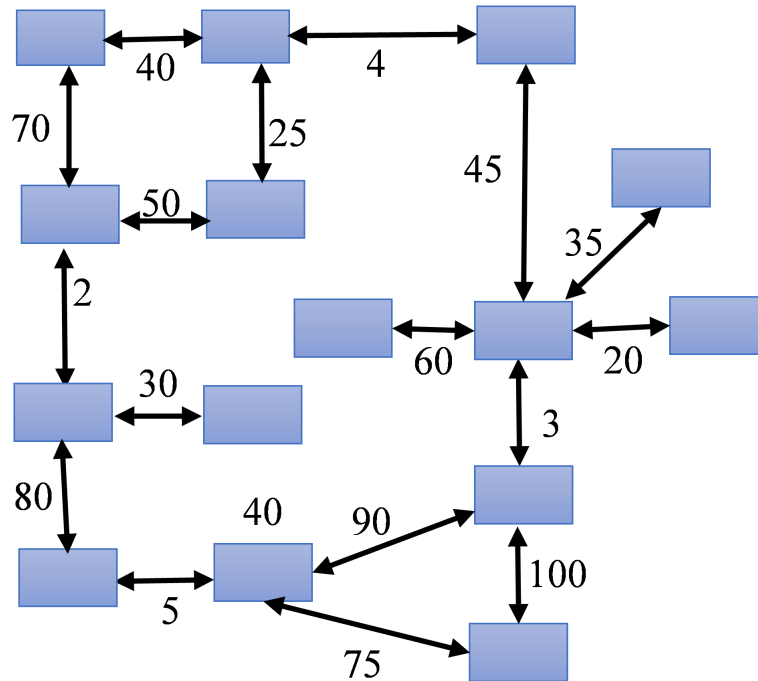


Figure 4.1: Communication between Cores

## 4.2 Structure of Router and Core for implementation of CBCT

Router, used to implement proposed approach CBCT, is a 2D router which consists of 5 in-ports and 5 out-ports. In-ports are used by node (router and core) to receive the packet whereas, out-port is used to send the packets to destination. There are 4 ports which are referenced as northPort, southPort, eastPort, westPort to communicate with the routers being connected in the north, south, east and west direction respectively. Along with these ports, there is one more port i.e. corePort. This port provides the connectivity among the router and the core. Fig.4.5 gives the diagrammatic representation of the 2D router being used in the CBCT approach. In Fig.4.5, port[0], port[2], port[4], port[1] represents the northPort, southPort, eastPort, westPort respectively, and port[3] represents corePort. Direction of arrow shows the flow of packets and communication among connected routers. If the structure of the core is considered, then for the implementation of the CBCT, the core is considered as source as well as the sink as shown in Fig.4.6. The two cores involved in the communication act as source and the sink. Core which transmit the packet to other core act as the source, whereas the core receiving the packets is referred as the sink.

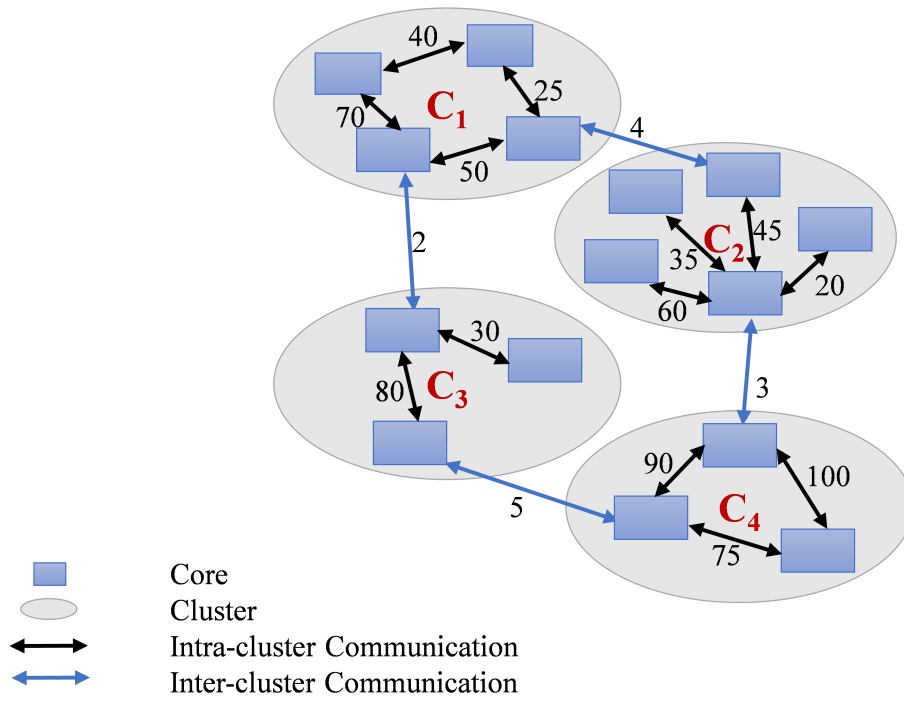


Figure 4.2: Intra-cluster and inter-cluster communication in CBCT

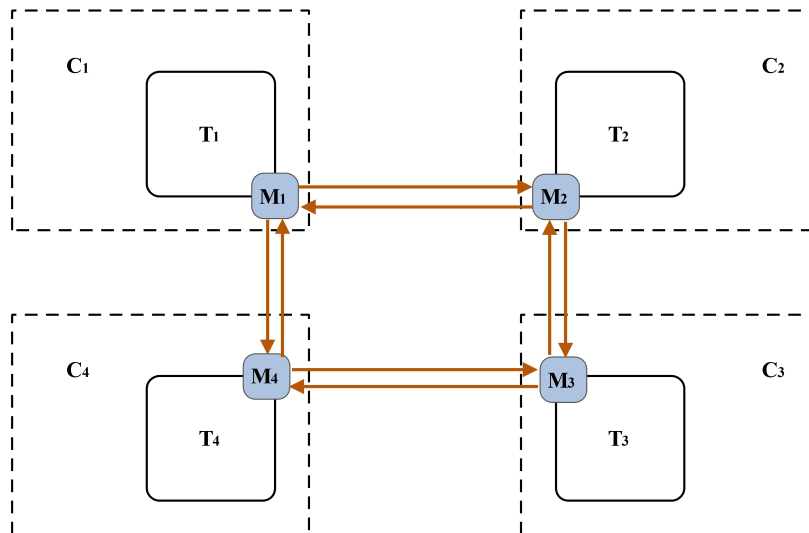


Figure 4.3: Topology based cluster graph

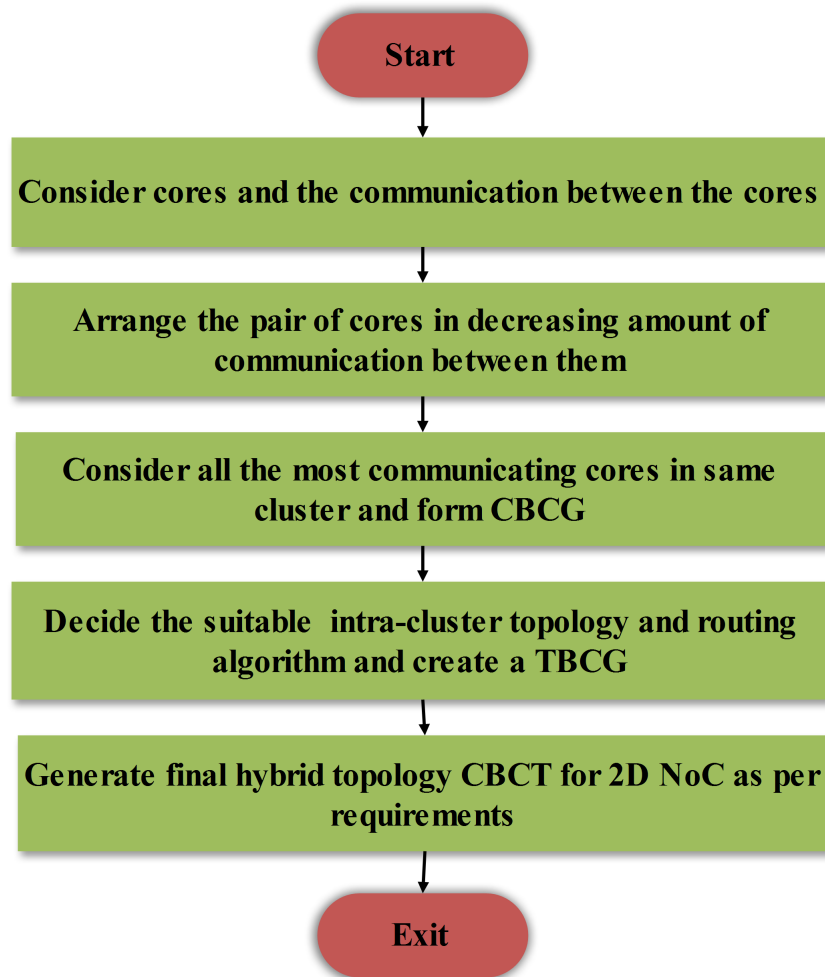


Figure 4.4: Flowchart for generating CBCT topology

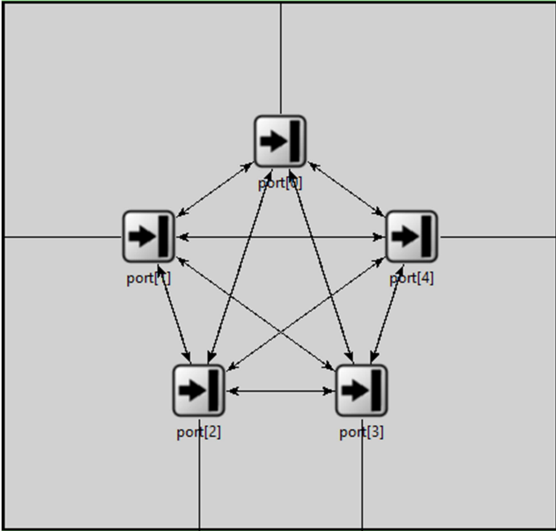


Figure 4.5: Structure of 2D router used for CBCT



Figure 4.6: Structure of 2D core used for CBCT



### 4.3 Master based Routing Algorithm

Existing routing algorithms doesn't prove to be an optimized solution for the implementation of the proposed approach, so we have introduced a routing algorithm i.e. master based routing algorithm. In master based routing algorithm, the routers which are the intermediate between two cluster topologies act as the master routers. As shown in the Fig.4.9 yellow colored routers are the master routers as they are intermediates and they are responsible for the communication between the clusters. These clusters are the networks, which consists of the most communicating components within same cluster, and lesser communicating in different cluster. As shown in the Fig.4.9, considers that there are 3 different topologies used in the 3 different clusters. Cluster  $C_1$  consists of mesh topology, cluster  $C_2$  consist of torus topology and cluster  $C_3$  uses bus topology for connecting different components for communication. For the purpose of routing the packets from source to destination, suitable routing algorithm corresponding to the topology is used like XY routing is best suited to mesh topology, if both source and destination belongs to same cluster, whereas if the source and destination belong to different clusters, then master based routing algorithm comes into active state. In master based routing algorithm, master has all the information of the routers and their corresponding cores belonging to the same cluster. Master also keeps record of the other master router directly or indirectly attached to it. Routers, other than master, also have the record to which cluster that router belongs to and have the information about their master. Suppose source and destination belongs to different clusters, then the packet is passed from the source to the master (corresponding to that cluster to which the source belongs). The master will communicate the other masters for checking to which cluster the destination core belongs to, then it will pass the packet to that master either directly or indirectly. Upon reaching to the master of destination core (following the corresponding routing algorithm associated with the topology of cluster), packet will route to the destination core.

### 4.4 CBCT : Communication based Cluster Topology

In order to evaluate the performance of CBCT topology, we compared the result of CBCT topology with mesh topology. We have considered 5X5 2D mesh topology as shown in Fig.4.7, in which the most communicating cores are represented by same colored cubes. Routers are represented as gray colored circles with cross.

For simulation of CBCT, we have considered a CBCG as shown in Fig.4.8. In Fig.4.8,

**Algorithm 2:** Master based routing**Data:** Communication based cluster graph (CBCG)**Output:** Cluster formation with suitable topology within and between clustersInitialize  $B_{inter}$  and  $B_{intra}$ ;Set  $B_{intra} = \text{constant}$ ;

$$B_{inter} = \frac{(n_t - n_m) * B_{intra}}{n_m} ;$$

**while** cores are not assigned to some cluster **do**

**while** most communicating cores, belonging to same group in the CBCG, are not all assigned to some cluster **do**

Cluster[k] = core[i];

Choose master core from each cluster and Master[i] = core[j];

Assign suitable topology to be used within the clusters and between the clusters according to the requirements;

**if** both source and destination belong to same cluster **then**

Pass packet from source to destination directly;

**else**

Pass the packet to the Master core;

Master core will further pass it to the other master core and in the same way to other master cores till the cluster having destination core is not reached;

Pass packet successfully to the destination;

there are 4 clusters with most communicating cores in the same cluster. These clusters are represented as  $C_a, C_b, C_c$  and  $C_d$ . Cluster  $C_a$  is a combination of most communicating cores  $\langle c_0, c_3, c_4, c_6, c_{14}, c_{17}, c_{20}, c_{23}, c_{24} \rangle$ ,  $C_b$  consists of  $\langle c_9, c_{10}, c_{11}, c_{12} \rangle$ ,  $C_c$  is a collection of cores  $\langle c_2, c_5, c_{22} \rangle$  and  $C_d$  consists of  $\langle c_1, c_7, c_8, c_{13}, c_{15}, c_{16}, c_{18}, c_{19}, c_{21} \rangle$ . If Fig.4.7 and Fig.4.9 are considered, we can conclude that the most communicating cores are far away from each other in mesh topology, and hence the hop count is large. For example, in Fig.4.7 cores 0 and 24 are most communicating cores, and the hop count between these cores is 8, so every time for communication between these two cores, it is required to cover a hop count of 8. This causes an increase in latency with an extra overhead. In CBCT as shown in Fig.4.9, the hop count required for the communication between cores 0 and 24 is 4, which is 50% efficient than the existing torus topology. Using CBCT approach, generated clustered topology can provide 40% to 80% of efficiency over existing topologies. Fig.4.10 shows flowchart for creating CBCT topology and routing packets from source to destination.

If two cores 0 and 1 are considered in Fig.4.7, it is clear that they are least communicating

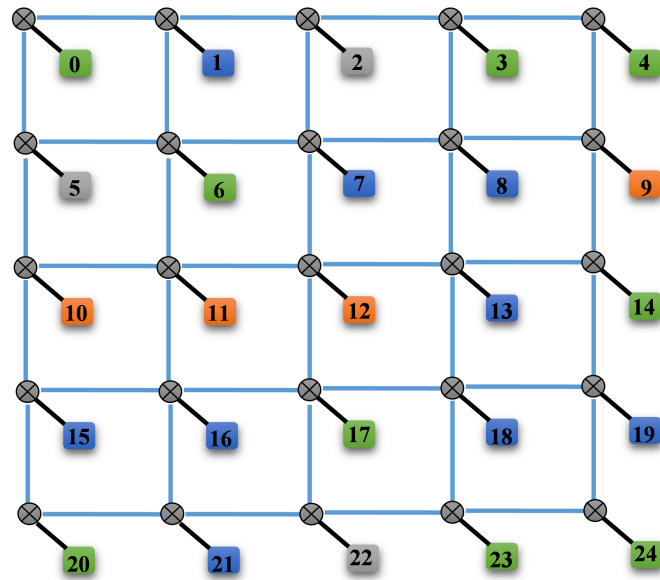


Figure 4.7: 5X5 2D Mesh Topology

cores represented using different colors, so keeping these cores far away will not affect the performance, hence in CBCT as shown in Fig.4.9, in order to bring most communicating cores nearby, these less communicating cores can be placed far from each other. To provide better performance, it is required to optimize latency and overhead. In order to place all the most communicating cores in one cluster, it should be verified that the hop count between these cores of the cluster should not increase as compared to the hop count between the same cores in mesh topology. The hop count between the two most communicating cores in the mesh topology, is considered as the threshold for this pair of cores in the CBCT. While designing the CBCT, it is required to consider that the threshold determined for a particular pair of cores should be minimized and hop count should be reduced. In proposed approach, our basic aim is to produce a hybrid topology, which can follow all the QoS requirements such as minimum latency, minimum bandwidth and maximization of the throughput. Communication among the cores can be determined by using CBCG graph with the weights assigned to the links. Cores with maximum communication should be in the same cluster, so that they take less time for communication, and hence improves processing speed. Less communicating cores can be in different clusters. Among the nodes within a clusters, one or more nodes are chosen as the master node. Master node act as an intermediate between the two clusters. It is the node responsible for the inter-core communication. The clusters can have any of the NoC topology

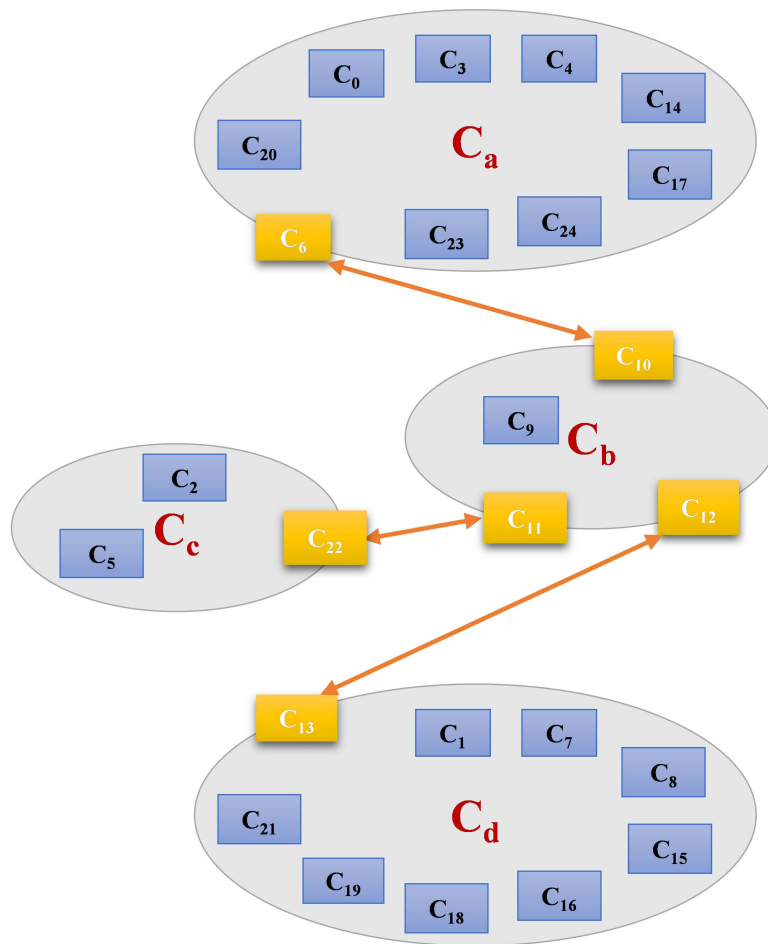


Figure 4.8: Cluster of most communicating cores

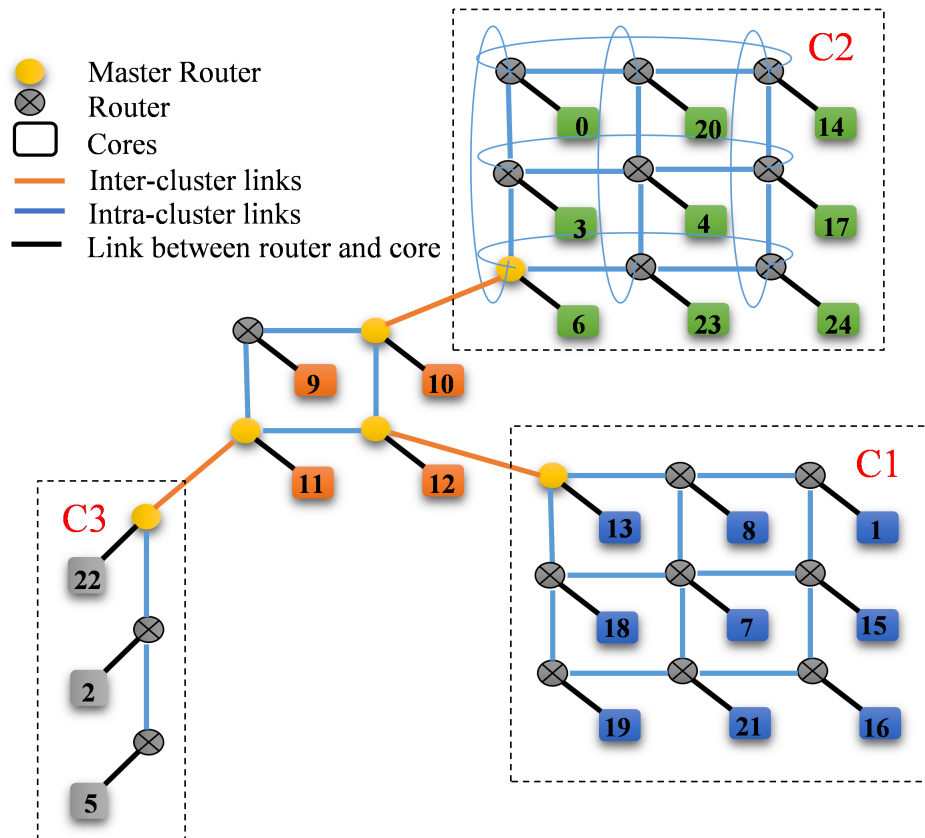


Figure 4.9: Communication based Cluster topology

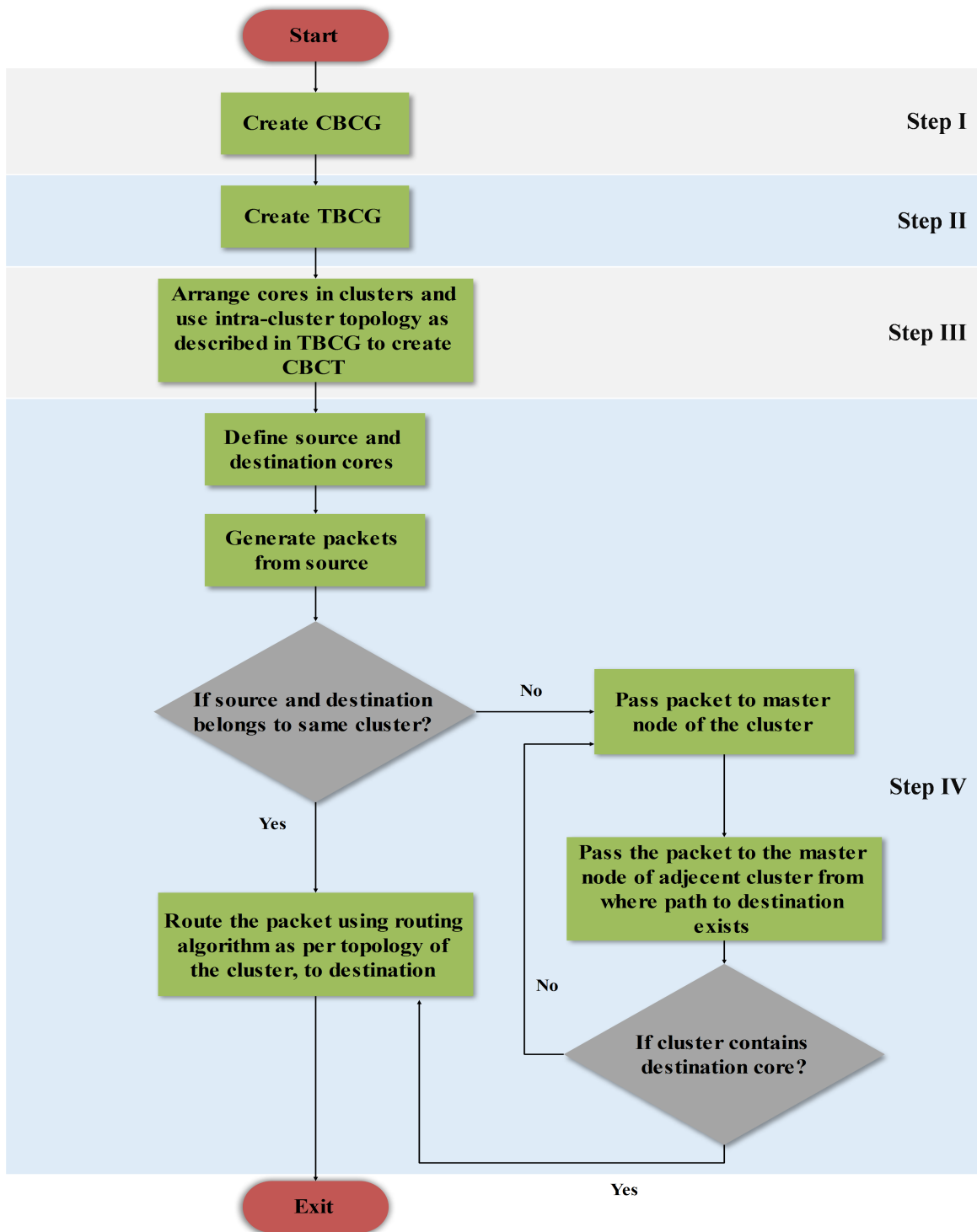


Figure 4.10: Flowchart for creating CBCT topology and routing packets from source to destination

as per the communication requirements among the cores within the cluster. Since single cluster can communicate to the multiple clusters, so there can be more than one master node within the cluster. If we consider the amount of traffic, then the maximum traffic is found to be at the links joining the two master nodes, as this is the single links which is responsible for the inter cluster communication. To reduce the amount of traffic on these inter-cluster links, it is required that some special configuration of these links have to be specified. Configuration of the links means the delay, bandwidth and the type of wire used as the links. In inter-cluster links, varying the bandwidth of the links have made it possible to simulate the required communication among the cores. For this purpose, the mathematical formula used in the proposed approach to set the bandwidth of the inter-cluster links is given as in Equation 4.1:

$$B_{inter} = \frac{N_{nm} \times B_{intra}}{N_m}, \quad (4.1)$$

Where,  $B_{inter}$  is the bandwidth of the inter-cluster links,  $N_{nm}$  is the number of non-master cores,  $B_{intra}$  is the bandwidth of intra-cluster links and  $N_m$  is number of master cores. This bandwidth reduces the traffic as huge amount of bandwidth is allocated to the most congested links. Depending upon the source and sink, the path is determined in the topology and packets are routed from source to destination using Master-based routing algorithm as discussed in section 4.3.

## 4.5 3D CBCT topologies for NoC

In this section the details about the implementation and designing of the CBCT for 3D NoC are discussed. For the purpose of comparing the results of the proposed 3D NoC CBCT topology we have implemented 3D NoC Mesh topology in OMNET++. The configuration of 3D Mesh topology is 5 X 5 X 4. In this mesh topology there are 4 layers each having dimension 4 X 4, i.e., there are 25 cores in each layer. In case of 3D CBCT for NoC we have considered two of the architectures, (i) with most communicating cores on different layers, (ii) most communicating cores on the same layer. In the first case as the most communicating cores are at different layers of mesh topology and hence a lot of time and effort are wasted for them to communicate to each other. But in CBCT the cluster for these most communicating cores are formed and hence the overhead involved in the mesh topology is reduced to a great extent.

Based on the single case we cannot prove that CBCT will produce best results for all the combinations possible. Hence to prove it best in all the cases we have considered the best case of

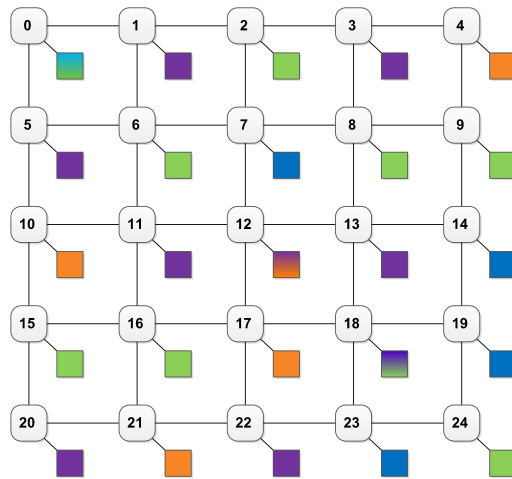


Figure 4.11: 5X5 2D Mesh Topology

the mesh topology. In the best case all the most communicating cores will reside on the same layer of the mesh topology. So for the CBCT the clusters of the cores on the same layer are formed. While considering the mesh topology in its best case also it was concluded from the experimental results that CBCT for 3D NoC proves to be the best.

Now the main focus is to make the CBCT topology geometrically equivalent to original mesh topology. For this first take the mesh topology. Determine the most communicating cores and create the CBCG. Then determine the topology for each cluster using TBCG. Add the links between the most communicating cores and remove the links in between the less communicating or non-communicating cores. Fig. 4.11 shows the given mesh topology. Fig. 4.11 shows the initially provided 5X5 mesh topology. Determine the most communicating cores in the given mesh topology. The most communicating cores have same color for differentiation. Some cores in the Fig. 4.11 has two colors, it determines that it is common to two of the clusters. Fig. 4.12 shows the formation of the clusters and generation of the heterogeneous and hybrid clustered topology. Fig. 4.13 shows that the mesh topology shown in Fig. 4.11 and CBCT shown in Fig. 4.12 are logically equivalent in terms of geometry, and hence the CBCT topology will consume nearly equivalent area as taken by mesh topology.



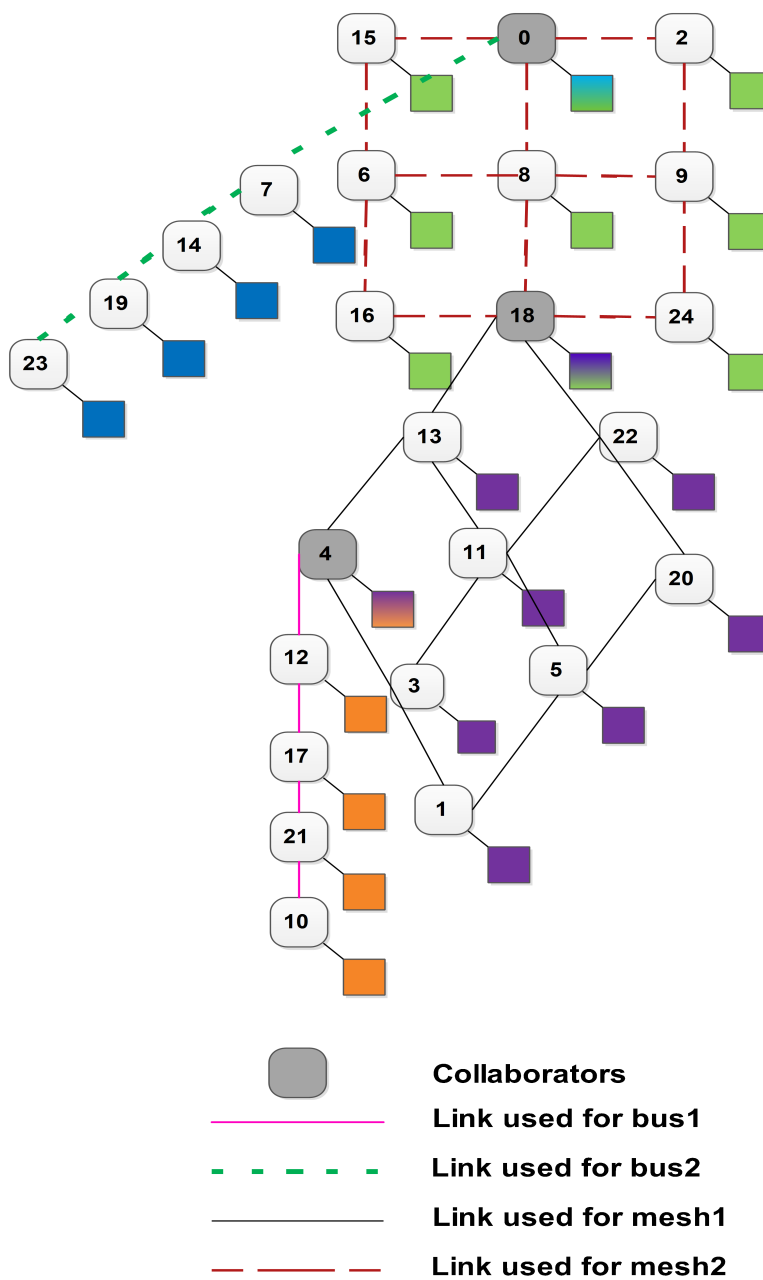


Figure 4.12: Heterogeneous and hybrid Clustered topology

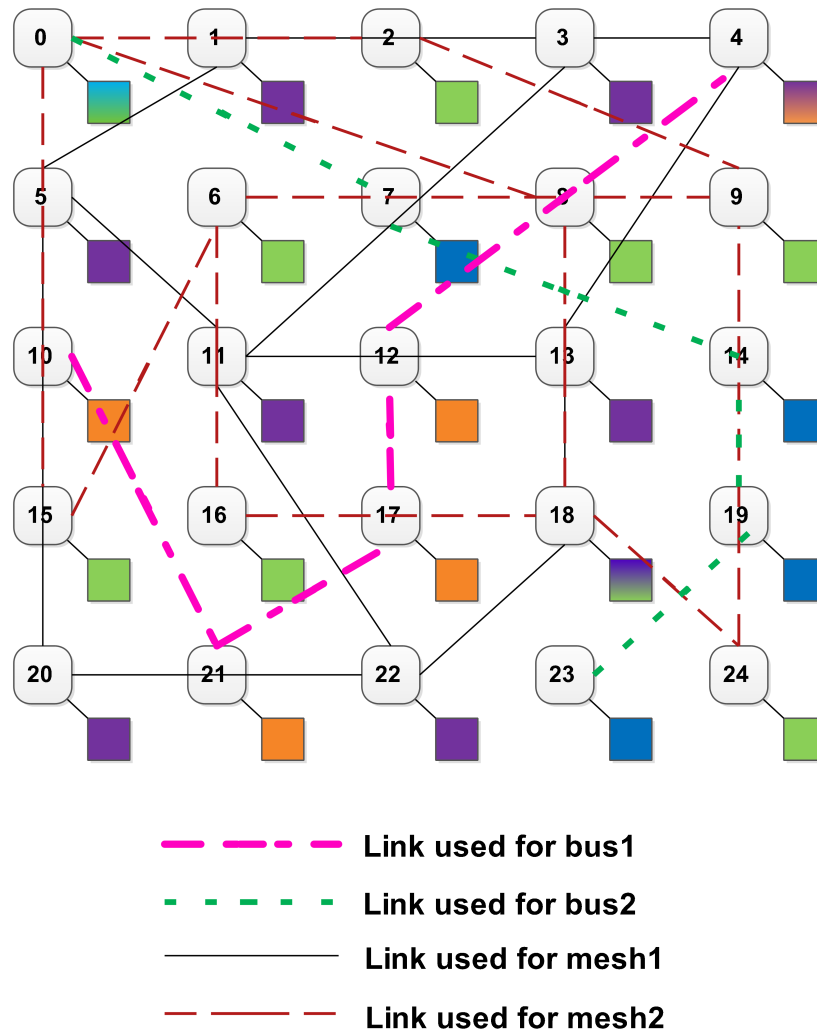


Figure 4.13: Clustered topology geometrically equivalent to mesh topology

## 4.6 Quality of Services

For the NoC topology in the proposed approach, we have multiple communication based clusters. QoS in CBCT topology can be determined by the contribution of the QoS parameters of each cluster individually. We consider the clusters as  $\langle C_1, C_2, \dots, C_n \rangle$ , for each cluster a set of QoS parameters will be defined based on the inter cluster or local topology used in that cluster. Let us consider the QoS parameters as  $\langle Q_{i1}, Q_{i2}, \dots, Q_{in} \rangle$  where,  $i$  denotes the cluster number. These QoS parameters considered can be normalized to give a threshold value by averaging the minimum and maximum QoS value obtained.

$$Q_{i\_min} = \min\{Q_{ij}\} \quad Q_{i\_max} = \max\{Q_{ij}\} \quad (4.2)$$

In Equation 4.2,  $i$  denotes the cluster number and  $j$  denotes the particular parameter for that cluster. We have to consider the mean or the average of two values in order to get the  $Q_{th}$  i.e. QoS threshold value for the particular cluster as calculated in Equation 4.3:

$$\overline{Q_{l\_th}} = \frac{\Sigma(Q_{i\_min}, Q_{i\_max})}{2} \quad (4.3)$$

QoS value for all the parameters of the cluster are considered and the average is calculated as in Equation 4.4:

$$\overline{Q_l} = \frac{\Sigma_{j=1}^n Q_{ij}}{n} \quad (4.4)$$

This average QoS value obtained decides whether the topology developed is best or not. For the experimental results, we have considered that, if  $(\overline{Q_l}) < (\overline{Q_{l\_th}})$ , then the topology generated is not fully optimized but if  $(\overline{Q_l}) > (\overline{Q_{l\_th}})$ , then the topology generated is most optimized. For obtaining a fully optimize global topology, we have introduced the concept of priorities. The parameter which provides  $Q_{i\_max}$  is considered to be at highest priority and the parameter which provides  $Q_{i\_min}$  is at lowest priority. So the QoS values at the local level, i.e. for each cluster is considered as given in Equation 4.5:

$$Q_i = P_{i1} * Q_{i1} + P_{i2} * Q_{i2} + \dots + P_{in} * Q_{in} \quad (4.5)$$

Where,  $P_{ij}$  determines the priority decided for  $i^{th}$  cluster and  $j^{th}$  QoS parameter for that

cluster.  $Q_{ij}$  determines  $j^{th}$  QoS parameter for  $i^{th}$  cluster. The  $Q_{ij}$  obtained for each cluster locally, is considered for calculation of the QoS value for full topology globally. The cluster providing the best QoS results, is considered to be at highest priority over others. The global QoS value of the whole topology is calculated as in Equation 4.6:

$$Q_{total} = \sum_{i=1}^n P_i * Q_i \quad (4.6)$$

There are certain QoS parameters which should be considered during the generation of the topology for the 2D NoC. We have considered different parameters as follows:

- *Latency*: Latency in network-on-chip include the delays involved in the interfaces to pass the packet from router to router, the delays on the routers and the delays due to the traffic, which makes the task to wait inside the core, instead of allowing them to move on the particular core for processing. The formula obtained for latency in the traditional mesh network would be given by Equation 4.7:

$$L_{mesh} = (n_r * Delay_r) + (n_l * Delay_l) + Q_t \quad (4.7)$$

$L_{mesh}$  represents the total latency in the NoC mesh network.  $n_r$  and  $n_l$  represents the number of routers and number of transmission links respectively.  $Delay_r$  and  $Delay_l$  represents the time taken on the router and time consumed during transmission over the transmission links respectively. Latency for the proposed topology can be calculated in the similar way. The latency for each cluster is calculated individually at local level, which is summed up to provide the global latency of the whole network. Local Latency of each cluster as in Equation 4.8:

$$L_c = (n_{(c,r)} * Delay_{(c,r)}) + (n_{(c,l)} * Delay_{(c,l)}) + Q_t \quad (4.8)$$

where  $L_c$  is the latency of the cluster,  $c$  used in the above equation represents the cluster number,  $r$  represents router and  $l$  represents the transmission links. Global latency of the topology as given by Equation 4.9:

$$L_t = \sum_{c=1}^n L_c + \sum_{j=1}^m Delay_j \quad (4.9)$$

Global latency of full network will be the sum of local latencies of all clusters combined with the delay on the inter cluster communication links.  $m$  shows the number of inter cluster links and  $Delay_j$  shows the delay involved on these inter cluster links.

- **Bandwidth:** It is the rate at which data transfers. Network bandwidth should be set such that the maximum data can be transferred either using lesser bandwidth or constant bandwidth. By this a huge amount of data can be passed using comparatively lesser bandwidth, along with minimum latency and energy involved. To get the most optimized results without increasing the amount of bandwidth required, we have to optimize the bandwidth at the inter-cluster links as they are most prone to the traffic and bottleneck. Keeping the bandwidth same at all intra-cluster links as that of the mesh topology, we have increased the inter-cluster link bandwidth proportionally in relation to the intra-cluster links. Let  $B_{intra}$  be the bandwidth of the intra-cluster link which same as the links used in the mesh topology.  $B_{inter}$  is given as in Equation 4.10:

$$B_{inter} = \frac{(n_t - n_m) * B_{intra}}{n_m} \quad (4.10)$$

Where  $n_t$  is the total number of cores,  $n_m$  be the master cores. So the above relationship is maintained by assigning the bandwidth to the inter-cluster links such that either the total bandwidth required reduces or is constant.

- **Bisection Width and Bisection Bandwidth:** The number of links which are broken or removed such that the two networks are obtained, which are equal in size or nearly equal. To obtain multiple paths between these sub-networks obtained, it is required to have maximum bisection width. More the bisection width, more is the bisection bandwidth. Bisection width in case of  $2^n \times 2^n$  2D mesh is taken as  $B_{w\_mesh} = 2^n$ . The bisection bandwidth of whole network is calculated as given in Equation 4.11:

$$B_{b\_mesh} = B_{w\_mesh} * Channel_{bandwidth} \quad (4.11)$$

Here,  $B_{b\_mesh}$  is the bisection bandwidth,  $B_{w\_mesh}$  is the bisection width of the network and  $Channel_{bandwidth}$  is the bandwidth of a channel in the network. While calculating the bandwidth in case of the proposed approach, we should divide the network on the inter-cluster links, which will correspond to a network with sub-networks as clusters. So, the bisection bandwidth for the CBCT with 5 interconnected clusters would be 4 and the 5 sub-networks are obtained. So the bisection width of the CBCT topology would be  $B_{w\_CBCT} = l$ , where  $l$  is the number of inter-cluster links. In CBCT the bisection width is less but bisection bandwidth is maximum, hence proves to be better over other existing

topologies. As it is considered that in order to pass a large amount of traffic over the inter-cluster links as they are maximum prone to bottleneck condition, hence to avoid this bottleneck situation, we have considered these links to have more bandwidth over other links. The bisection bandwidth of the whole network would be considered in this case as in Equation 4.12:

$$B_{b\_CBCT} = B_{w\_CBCT} * ICL_{bandwidth} \quad (4.12)$$

In the equation  $B_{b\_CBCT}$  is the bisection bandwidth,  $B_{w\_CBCT}$  is the bisection width of the network and  $ICL_{bandwidth}$  is the bandwidth of inter cluster link in the network. Suppose that in order to control the traffic in the network, the bandwidth of the inter-cluster links are taken different. In that case the formula in Equation 4.12 is given as in Equation 4.13:

$$B_{b\_CBCT} = \sum_{l=1}^k ICL_{bandwidth} \quad (4.13)$$

Where,  $k$  is the number of inter-cluster link,  $ICL_{bandwidth}$  is the bandwidth of the  $l_{th}$  inter-cluster link.

- **Loss Probability:** Loss probability is the calculation of the lost packet in the stream over the total packets passed. It can be found by dividing the number of packets lost in the network from source to destination by the total number of packets to be send over the network from source to destination is given in Equation 4.14 :

$$L_p = \frac{n_t - n_s}{n_t} \quad (4.14)$$

where,  $L_p$  is the loss probability,  $n_t$  is the number of total packets to be passed and  $n_s$  is the packets passed successfully over the network. As there are multiple paths in the topology generated using proposed approach and also the links having maximum congestion are provided higher bandwidth, in order to provide less loss probability. During experimental result, we have concluded that the loss probability of the proposed approach is lesser than that of the traditional mesh topology. In mesh topology the loss probability was calculated as 0.76 whereas in CBCT it is computed to be 0.54.

- **Energy :** For defining energy consumption for a NoC topology, combination of cores and routers together are considered as a tile. Energy consumption of packet from tile  $t_i$  to  $t_j$  is represented as  $E_{Packet}^{t_i,t_j}$ .  $E_{Packet}^{t_i,t_j}$  is divided into two parts: (i)  $E_{Link}$ , energy consumed by

link, and (ii)  $E_{Router}$ , energy consumed by router.  $E_{Link}$  is calculated as follows given in Equation 4.15 and 4.16:

$$P_{Link} = (P_{Dynamic} + P_{Leakage}) * Num\_Ports \quad (4.15)$$

$$E_{Link} = \frac{P_{Link}}{Frequency} \quad (4.16)$$

$E_{Router}$ , is a combination of three energies i.e. energy of buffer, crossbar and arbitrator as given by equation 4.17.  $E_{Arbiter}$  is calculated as given in Equation 4.18.  $E_{Packet}^{t_i, t_j}$  from tile  $t_i$  to  $t_j$  is computed as given in Equation 4.19. Total energy consumption of NoC topology ( $E_{Total}$ ) is calculated in Equation 4.20, where N is total number of messages in NoC topologies.

$$E_{Router} = E_{Buffer} + E_{Crossbar} + E_{Arbiter} \quad (4.17)$$

$$E_{Arbiter} = E_{SW\_Allocator} + E_{VC\_Allocator} \quad (4.18)$$

$$E_{Packet}^{t_i, t_j} = num_{Hops} * E_{Link} + (num_{Hops} - 1) * E_{Router} \quad (4.19)$$

$$E_{Total} = \sum_{i=0, j=0}^N E_{Packet}^{t_i, t_j}, \text{ where } i \neq j \quad (4.20)$$

Hence equation 4.20 is used to evaluate and compare the energy consumption in the case of Mesh topology and CBCT.

# Chapter 5

## Simulator Tool

### 5.1 OMNET++ Simulator

OMNET++ is an C++ simulation library which is component-based and act as a framework, basically used for the deveopment of the network simulators. "Network" is meant in a broader sense that includes wired and wireless communication networks, on-chip networks, queueing networks, and so on. Domain-specific functionality such as support for sensor networks, wireless ad-hoc networks, Internet protocols, performance modeling, photonic networks, etc., is provided by model frameworks, developed as independent projects. OMNET++ offers an Eclipse-based IDE, a graphical runtime environment, and a host of other tools. There are extensions for real-time simulation, network emulation, alternative programming languages (Java, C), database integration, SystemC integration, and several other functions. OMNET++ is free for academic and non-profit use, and it is a widely used platform in the global scientific community. Fig. 5.1 shows the flowchart for the execution in OMNET++.

- HNOCS - Network on Chip Simulation Framework

NoC simulation provides a HNOC tool which is most important for research for NoC architectures resulting from a huge cost involved for VLSI prototypes which uses modern processes for manufacturing. There are a number of NoC simulators which exist, they may be proprietary or are built on some of the infrastructures which are not standard. HNOCS is basically an implementation of an open-source NoC simulator tool which make use of OMNET++ [48].



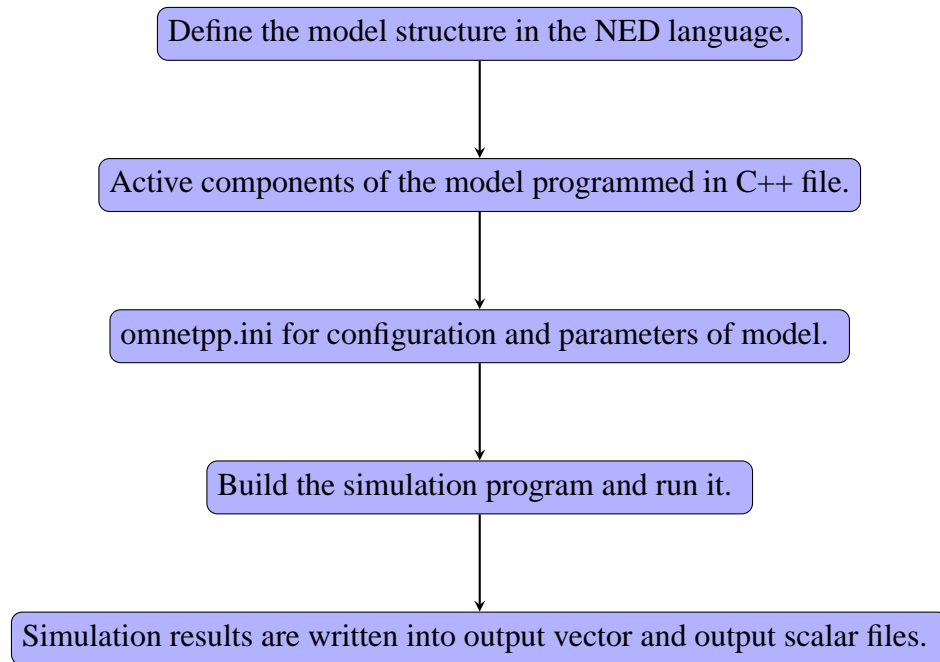


Figure 5.1: Flow chart representation of execution of steps in OMNET++

The HNOCS tool make utilization of the OMNET++ features which can support the run time selection of the modules for simulation from a parameterized component library. Here the op- CalcType parameter is basically set from "XY" to "XY/YX" due to which the simulation make use of the different algorithm for the selection of the output port. Models used here provide a support for the heterogeneous configuration such that it deals with the link capacity and virtual channel numbers. Implementation of the wormhole switching is provided by the HNOCS modules available in todays world, with arbitration done using round-robin. Different router are implemented using HNOCS which are synchronous or asynchronous and a provide a full queuing for virtual output (VOQ) with usage of the FIFO for each of the tuples.

## 5.2 Orion 2.0 Simulator

Orion simulator provides the instruction to embed power model into network [49]. Orion 2.0 simulator includes features such as:-

- Support 90nm, 65nm, 45nm and 32nm technologies.
- It basically supports three operations which include (i) high performance in terms of the

high power, then the second parameter is the normal performance and the third parameter on which it works is the low power and also the low performance.

- In this the routers are added with the router area model.
- It can be used to calculate the link power.
- In this the clock power models are present.

# Chapter 6

## Design

In this chapter the design structure of the different architectures used for the development of 2D and 3D Networks on Chip topology are considered. Fig. 6.1 shows the design structure used during the simulation of the mesh topology in OMNET++ simulator. There are 25 cores being used in this NoC mesh topology. The corresponding proposed 2D Cluster based hybrid topology is shown in Fig.6.2. This CBCT in 2D NoC also have 25 cores in total to be compared with the 2D mesh topology.

The 3D mesh topology considered here has 100 cores as shown in Fig. 6.3. In this the 3D mesh topology with 100 cores is being designed in OMNET++ simulator. Here 4 layered mesh topology is considered where each layer has 25 cores in total. This corresponds to the development of the mesh topology as a 5 X 5 X 4 mesh topology. Fig. 6.5 shows the corresponding 3D CBCT based layered topology. This topology is formed by considering the most communicating layers in the same layer of mesh topology, so now while building the CBCT based hybrid topology the cluster of the cores residing in the same layer of the mesh topology is formed and hence there are three reduced mesh clusters formed with each cluster having 24 cores.

Above considered was the best case of the mesh topology in which the most communicating cores are in the same layer of the mesh topology. But here for explaining the concept of CBCT topology in 3D NoC and prove that the best results are obtained in CBCT hybrid in Fig. 6.4 the average case is considered. Here the most communicating cores are considered to be at different

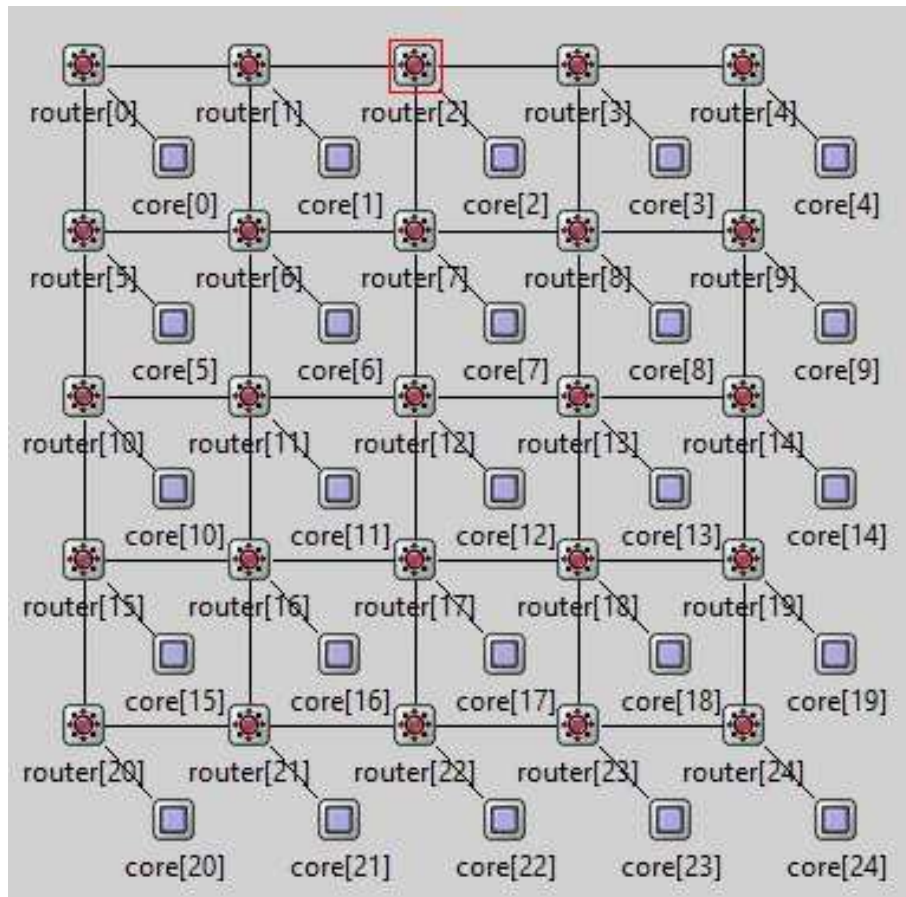


Figure 6.1: Design of 5X5 Mesh Topology

layers of the mesh topology. CBCT based hybrid topology is formed for this approach. Here different topologies are used in different clusters forming an heterogeneous approach.

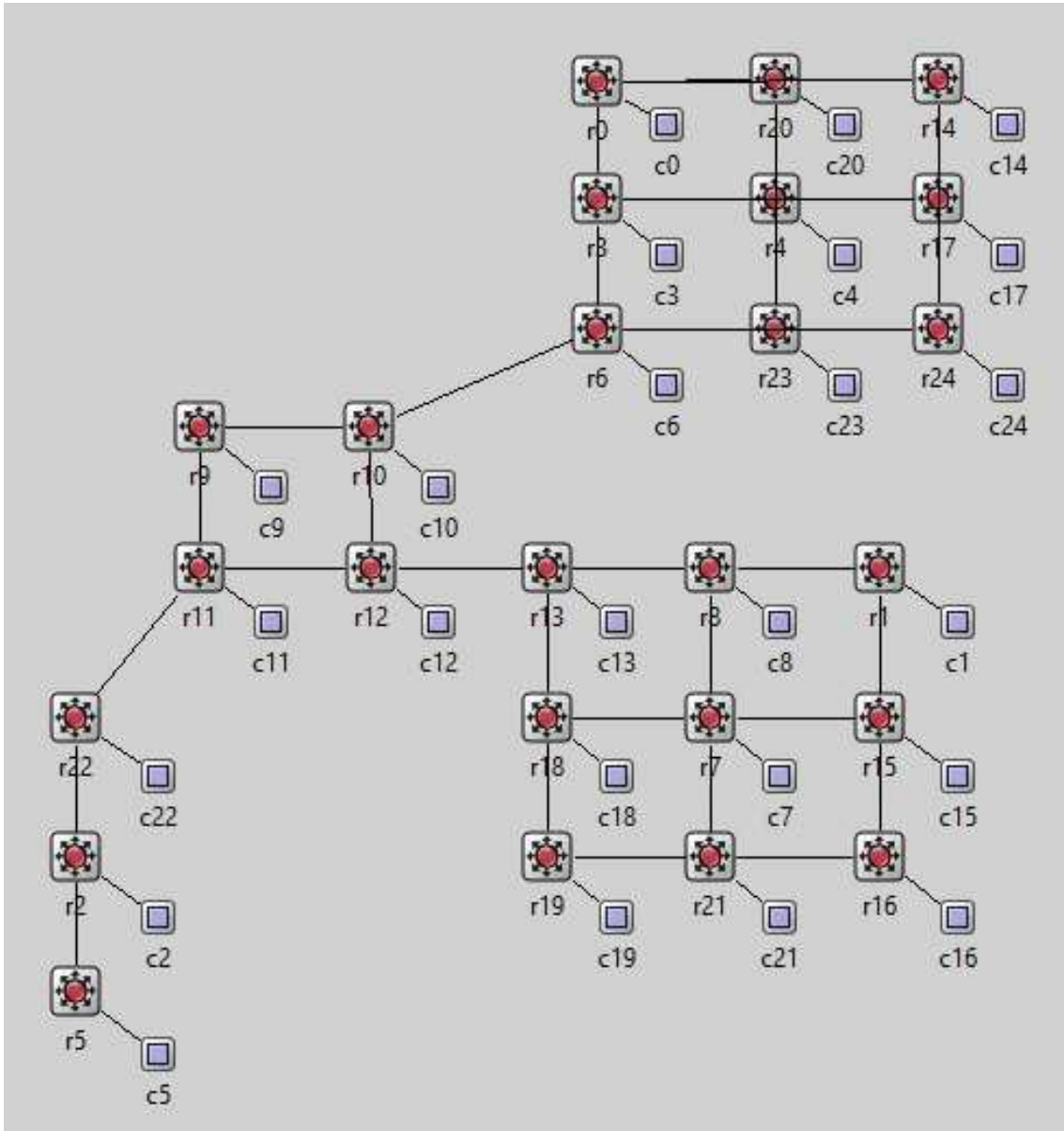


Figure 6.2: Design of Cluster based Hybrid Topology

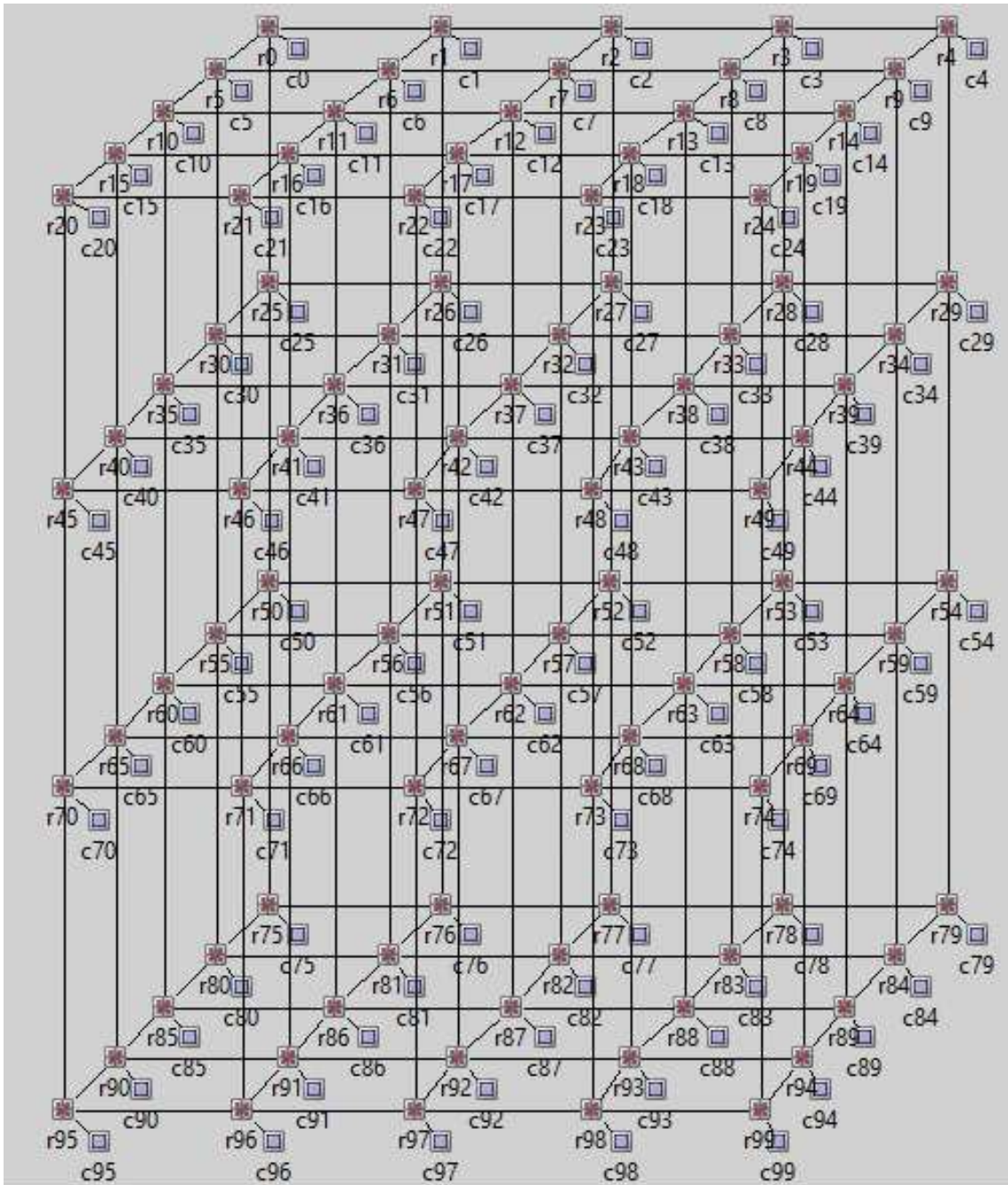


Figure 6.3: 5X5X4 (100 processing elements) 3D Mesh



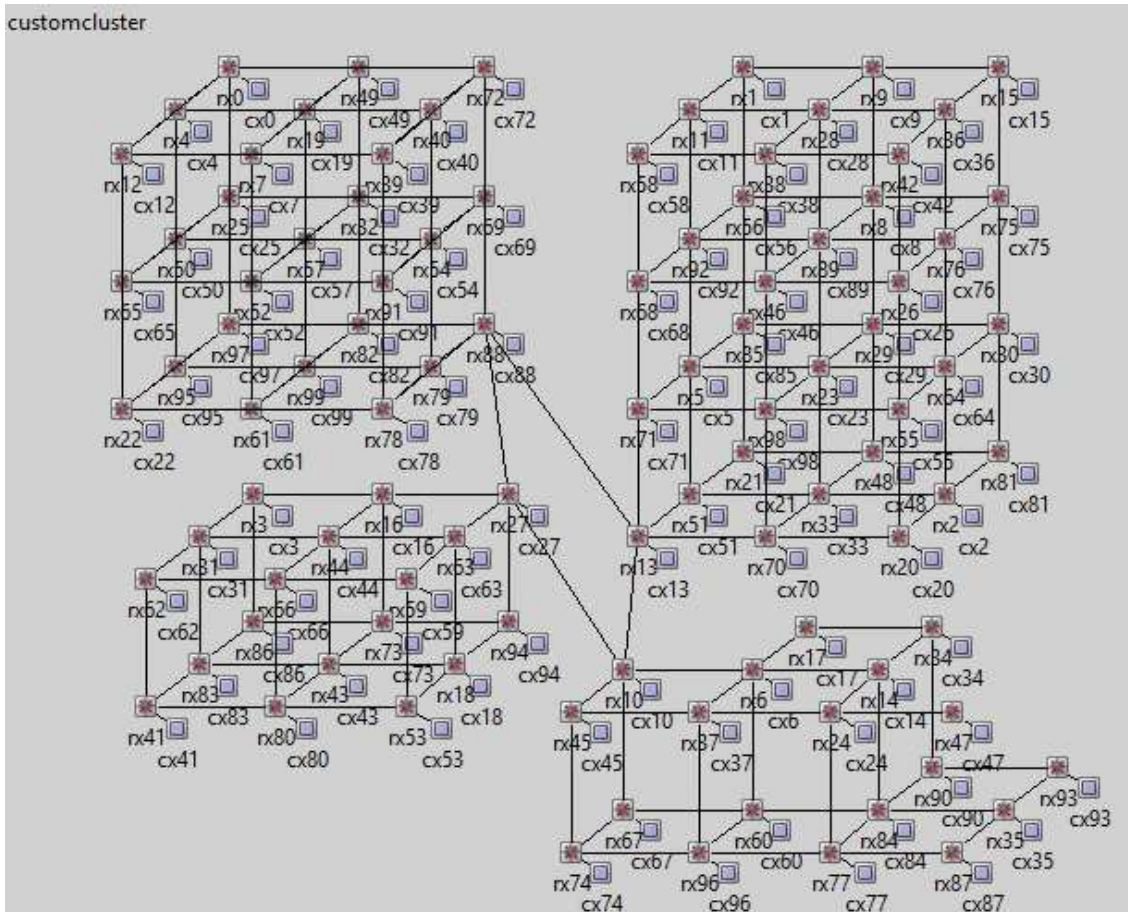


Figure 6.4: Hybrid CBCT topology

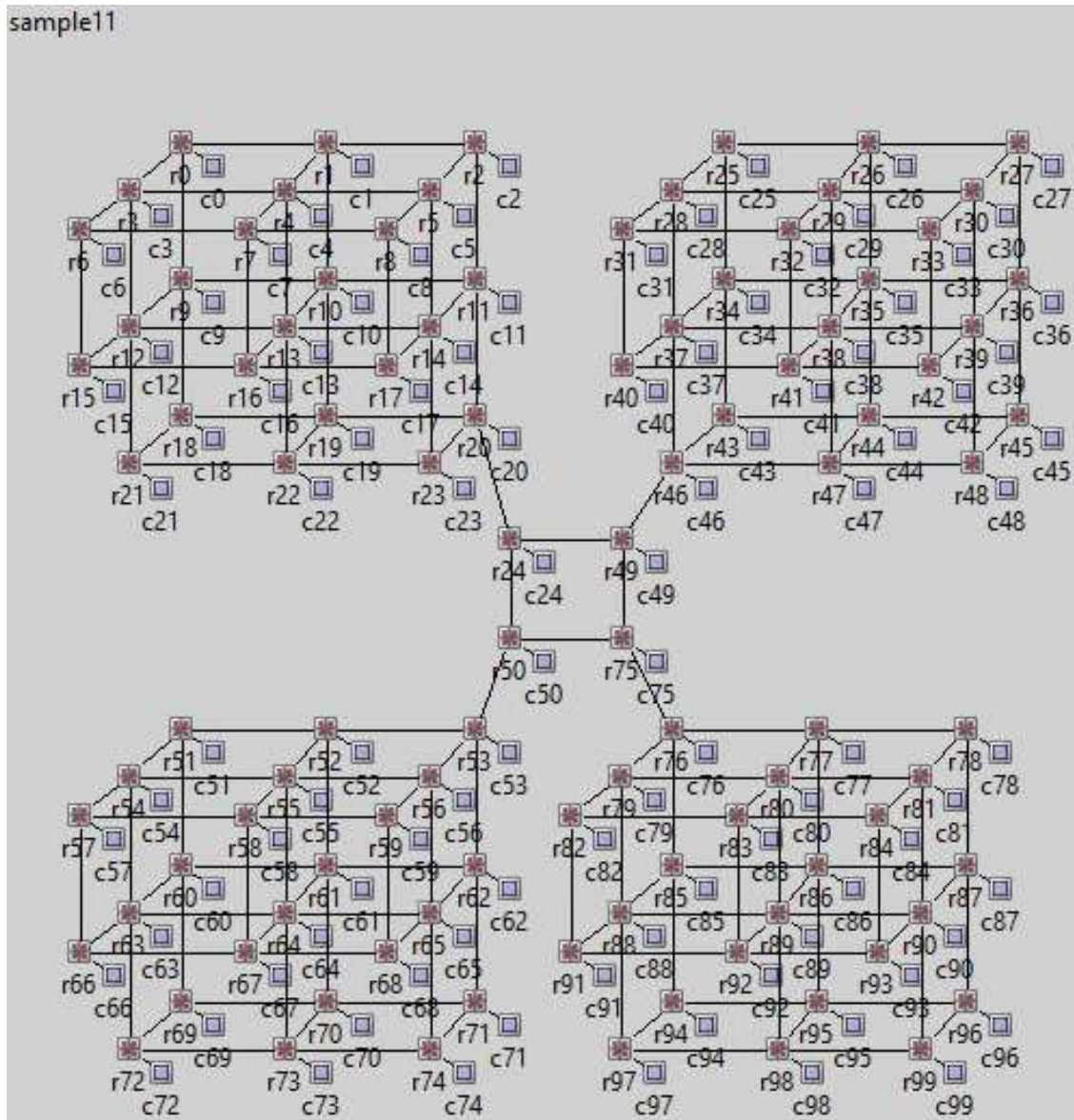


Figure 6.5: Layered CBCT topology



# Chapter 7

## Experimental Results

For the purpose of simulating the CBCT topology for NoC, we have used the OMNET++ simulation tool with HNoC package. We have simulated the hybrid topology (CBCT with 25 nodes) and has compared the results of CBCT topology with 5 X 5 mesh topology. The parameters considered to get the experimental results for end-to-end latency, network latency, packet latency, sink bandwidth, loss probability and link utilization are given in Table 7.1 and parameters to calculate energy consumption are given in 7.2. Table 7.3 gives the detailed description of the values obtained during the simulation of CBCT and Mesh topology in OMNET++. Experimental results in section 6 shows that proposed approach proves to be better than other existing topologies. The main purpose of the proposed approach is to optimize the parameters such as end-to-end latency, network latency, packet latency, loss probability, link utilization and energy consumption of topology at minimum bandwidth required. For experimental results, we have considered 25 core which can be increased to any number of cores as per the requirements. In the Fig.7.1, end-to-end latency of the mesh topology and end-to-end latency of CBCT are compared for different cores keeping source core constant as core 0. Fig.7.1 clearly shows that CBCT approach is better than existing mesh topology and optimizes approximately 50% of end-to-end latency. If we consider the network latency, then the proposed approach proves to be more efficient. To get a clear picture, we can see the graph of network latency for mesh topology and CBCT are compared in Fig.7.2. If the average of the network latency and packet latency in both the cases is considered, then it is concluded that CBCT proves to be more optimized as compared to the mesh topology. These comparisons can be easily visible in the Fig.7.3. While in order to optimize QoS parameters, it is required that bandwidth should not be affected adversely, either bandwidth is optimized or it should remain constant. While simulating the results, keeping the bandwidth constant, we conclude that

Table 7.1: Parameters considered for simulation of mesh topology and CBCT.

C.No	Parameters	Mesh Topology	CBCT
1	Number of Nodes	25	25
2	Rows	5	-
3	Columns	5	-
4	Number of Clusters	-	4
5	Routing	x-y routing	According to topology used in cluster
6	Flit size	4bytes	4bytes
7	Message Length	4	4
8	Packet Length	8 (in flits)	8 (in flits)
9	maximum queued packets	4	4
10	data rate	4Gbps	4Gbps
11	Inter-cluster link	-	3
12	Data rate for Inter-cluster link	-	13Gbps - 16Gbps

Table 7.2: Parameters considered for calculation of Energy Consumption in mesh topology and CBCT.

C_No	Parameters	Values
1	Technology Used	32 nm(nanometer)
2	Transistor Type	LVT
3	Voltage Vdd	1.0 V
4	Frequency	1 * e+9
5	Router INport	5
6	Router Output	5
7	Flit Width	32 bits
8	Virtual Channel Used	2
9	CrossBar Model Used	Multistage Crossbar Switch
10	Buffer Size	4
11	Wire type	LOCAL

Table 7.3: Simulation details for mesh topology and CBCT.

C_No	End-to-End Latency		Network Latency		Packet Latency		Sink Bandwidth	
	Mesh	CBCT	Mesh	CBCT	Mesh	CBCT	Mesh	CBCT
1	264	152	32	112	88	168	22.725	26.7043
2	280	144	48	104	104	160	20.8313	18.9515
3	296	72	64	32	120	88	20.8313	22.3972
4	312	88	80	48	136	104	30.9313	32.7343
5	264	160	32	120	88	176	21.4625	23.2586
6	280	88	48	48	104	104	20.2	15.5057
7	296	152	64	112	120	168	21.4625	17.2286
8	312	136	80	96	136	152	19.5688	21.5357
9	328	112	96	72	152	128	20.8313	23.2586
10	280	96	48	56	104	112	22.725	24.12
11	296	120	64	80	120	136	19.5688	18.09
12	312	112	80	72	136	128	20.8313	22.3972
13	327.9	120	96	80	152	136	18.0301	14.6443
14	344	72	112	32	168	88	20.8313	20.6743
15	296	168	64	128	120	184	18.9375	17.2286
16	312	184	80	144	136	200	16.4125	18.09
17	328	88	96	48	152	104	17.0438	18.09
18	344	136	112	96	168	152	18.3063	18.9515
19	360	152	128	112	184	168	25.8813	21.5357
20	312	72	80	32	136	88	24.2637	25.1968
21	328	168	96	128	152	184	22.0938	20.6743
22	344	128	112	88	168	144	20.2	22.3972
23	360	104	128	64	184	120	18.9375	18.09
24	376	104	144	64	200	120	17.0438	18.09

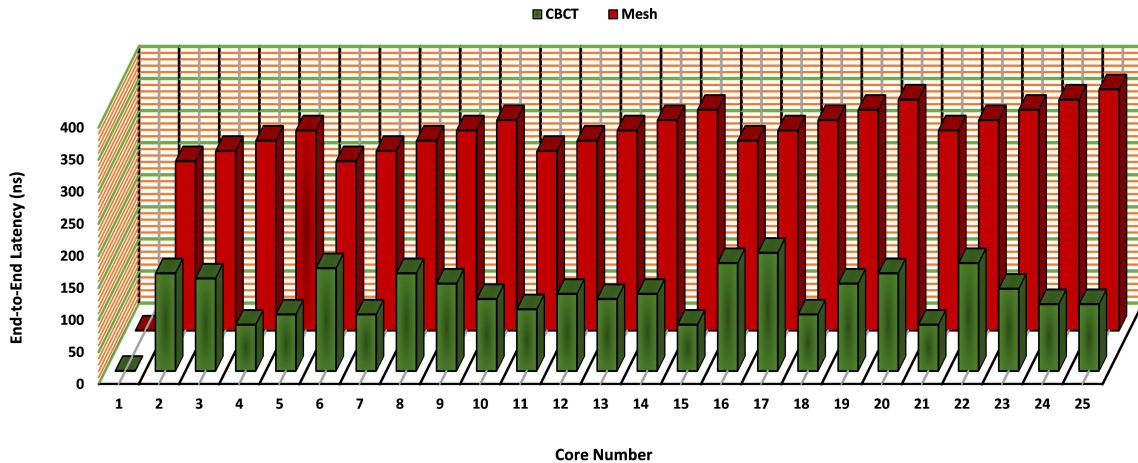


Figure 7.1: End-to-End Latency

the results are optimized in CBCT. On the other hand, optimization of bandwidth to a certain extent can produce even better results. Fig.7.4 shows the bandwidth required in case of mesh topology and CBCT. Link utilization can be defined as the percentage of the total usage of the bandwidth for a particular link. More the link utilization, more is the requirement of bandwidth, leading to a high consumption of energy. Fig.7.5 shows the comparison of the link utilization for 125 links for 25 cores. From Fig.7.5 it is clear that the link utilization in case of CBCT is much lesser as compared to that of mesh topology and hence less amount of bandwidth is required. Readings for different experimental results are taken in OMNET++ and it was observed that hybrid topology CBCT proves to be better. Its performance can further be improved according to the requirements by customizing the topology for the clusters accordingly.

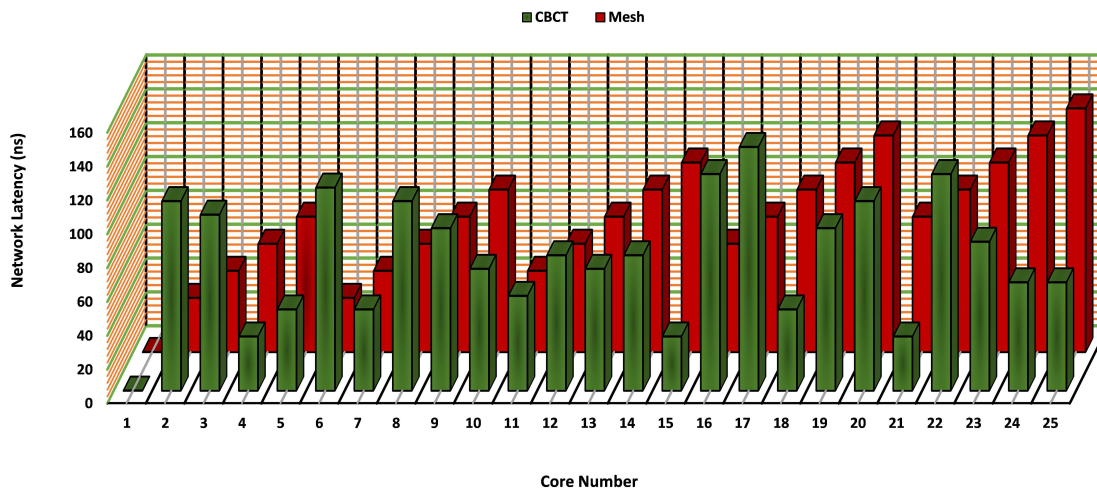


Figure 7.2: Network Latency

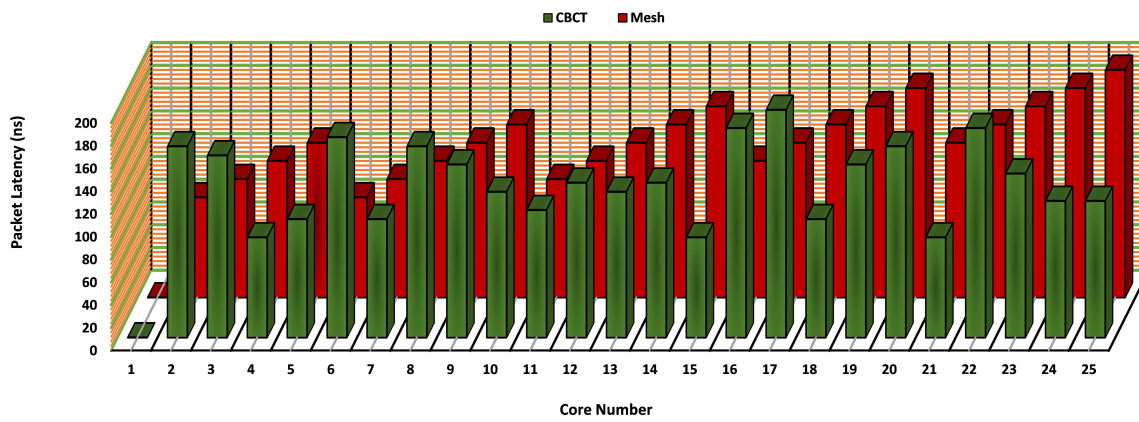


Figure 7.3: Packet Latency

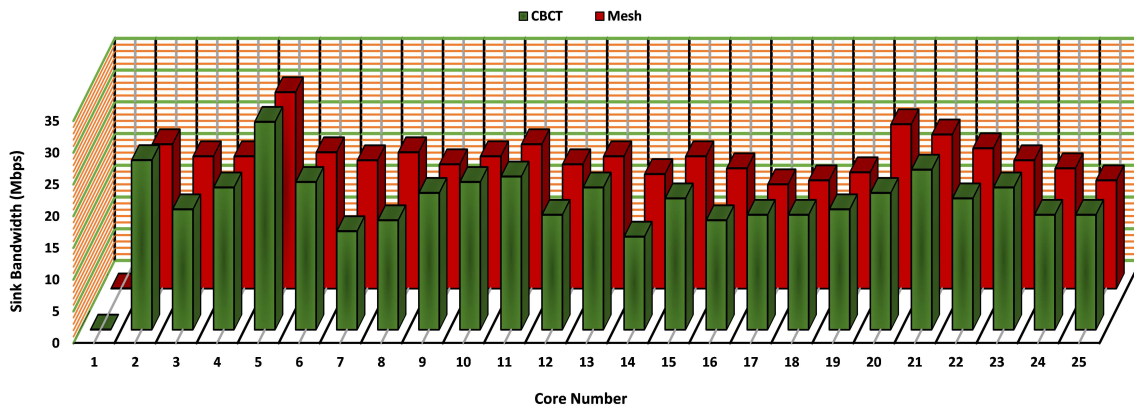


Figure 7.4: Sink Bandwidth

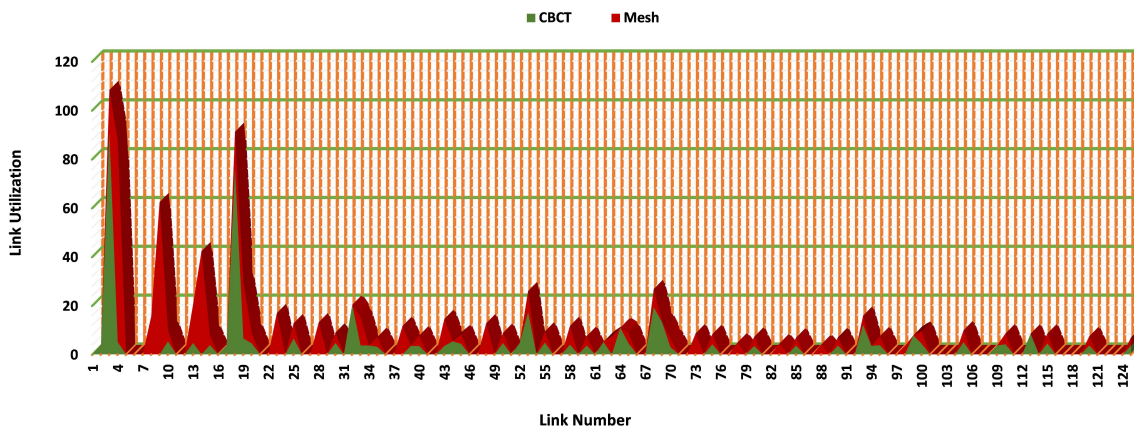


Figure 7.5: Link Utilization

Table 7.4: Energy consumption of Link (in pJ).

Load	Link Length					
	1mm	2mm	3mm	4mm	5mm	6mm
0.2	7.65	15.31	22.97	30.63	38.28	45.94
0.4	12.10	24.20	36.30	48.4	60.50	72.6
0.6	16.54	33.08	49.62	66.17	82.71	99.25
0.8	20.98	41.97	62.95	83.94	104.93	125.91
1	25.42	50.085	76.28	101.71	127.14	152.57

Table 7.5: Energy consumption of Router (in pJ).

Load	Energy consumption
0.2	16.8
0.4	27.1381
0.6	37.4573
0.8	47.7765
1	58.0958

For calculating the energy consumption in topologies as mentioned in this paper, we have used Orion 2.0 simulator.  $E_{Link}$  and  $E_{Router}$  at different load and link length (in mm) are calculated in Orion 2.0 simulator and the detailed overview is given in Table 7.4 and Table 7.5. Using formula as given in Equation 4.16 - 4.20, we compute the energy consumption of mesh and CBCT topology. In Fig.7.11, we compared the energy consumption at different cores in both mesh topology and CBCT. Experimental result shows that there is less energy consumption in CBCT topology as compared to mesh topology. CBCT has high speed of data transfer at less bandwidth required in case of most communicating cores. For simplicity, in order to represent the energy consumption of the topologies we have considered 6 cases. 6 cores are considered randomly to evaluate energy consumption on each one of them. The core selected randomly are  $c_4, c_9, c_{14}, c_{17}, c_{19}, c_{24}$ . Energy consumption for these cores at different link length and load are shown in the Fig. 7.11. From Fig. 7.11 it is clearly visible that energy consumption in CBCT is much lesser as compared to mesh topology for different load and link lengths.

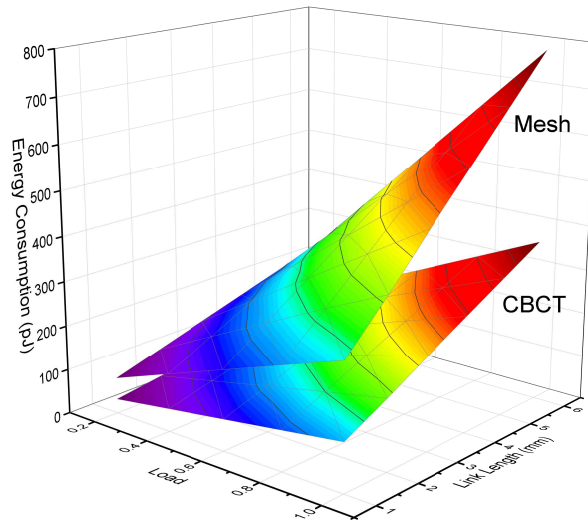


Figure 7.6: Energy consumption on core  $c_4$

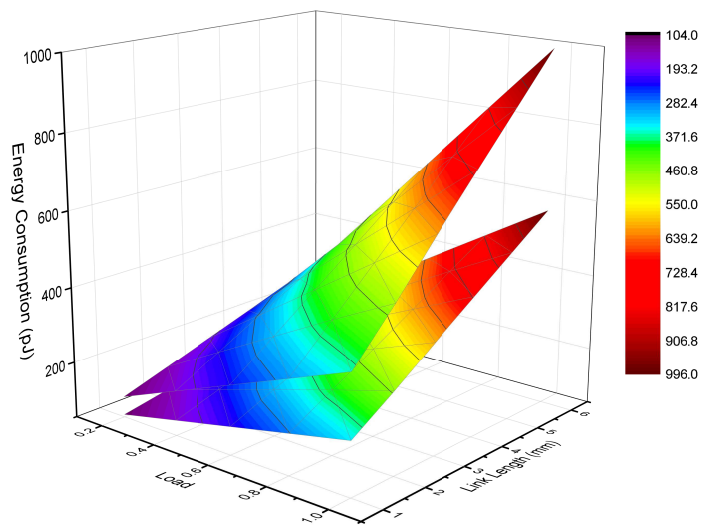


Figure 7.7: Energy consumption on core  $c_9$



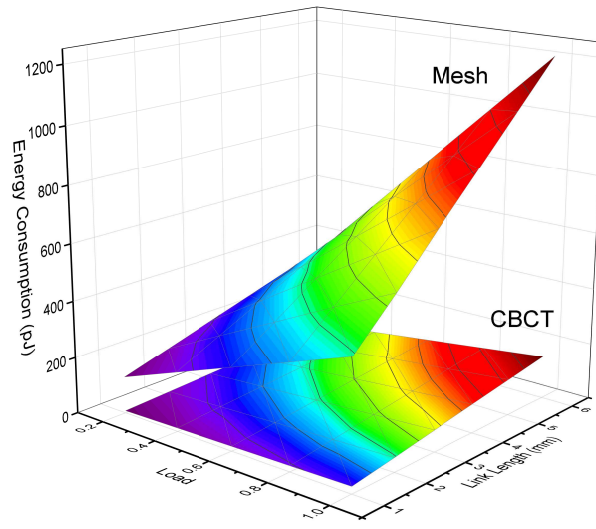


Figure 7.8: Energy consumption on core  $c_{14}$

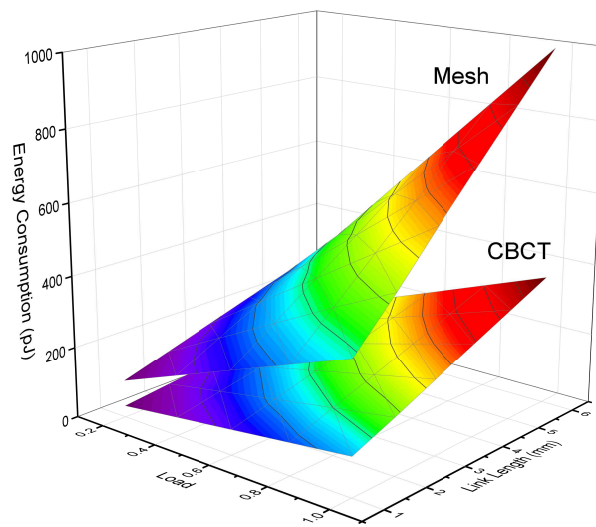


Figure 7.9: Energy consumption on core  $c_{17}$

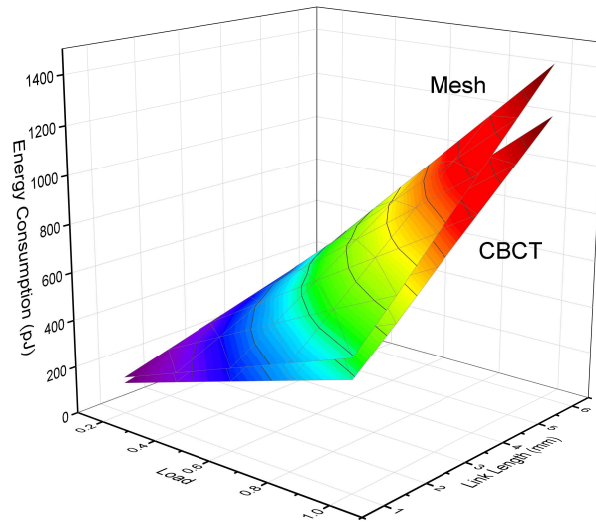


Figure 7.10: Energy consumption on core  $c_{19}$

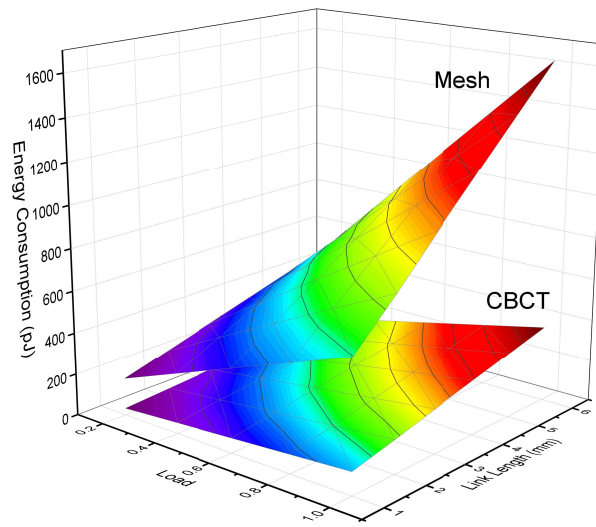


Figure 7.11: Energy consumption on core  $c_{24}$

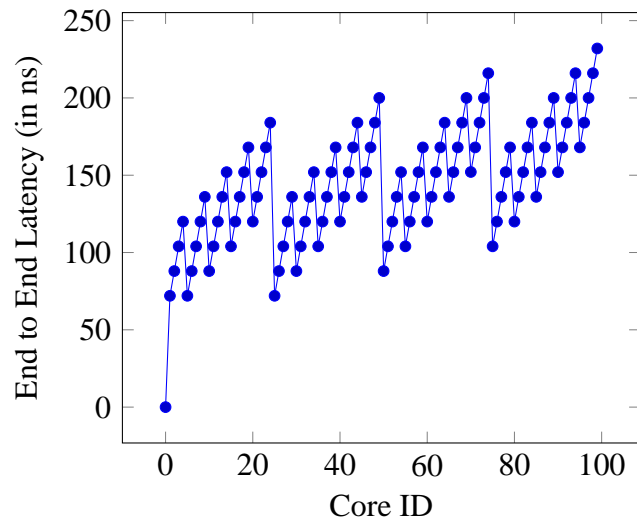


Figure 7.12: End to End Latency in 3D Mesh Topology

## 7.1 Experimental results of 3D NoC CBCT topology

As already discussed that for the proposed approach the CBCT topology is considered in two ways, one in which the most communicating cores lies on the different layers of the mesh topology and the second in which the most communicating cores are on the same layer of the mesh topology. Here the results corresponding to both are given in comparison to the mesh topology.

### 7.1.1 Results of 3D Mesh topology

In Fig.7.12. the detailed description of the end-to-end latency is shown. The results obtained corresponding to the network latency are shown in Fig.7.13. Packet latency results as shown in Fig.7.14 gives the detailed description of the packet latency for 3D Mesh topology. Fig.7.15 gives the sink bandwidth required during the communication between the cores. In Fig.7.16 the detailed description of the link utilization are shown for 3D mesh topology.

### 7.1.2 Results of 3D Hybrid topology

Fig.7.17 gives the detailed description of the end-to-end latency in the 3D hybrid topology for NoC. In Fig.7.18 the network latency of the hybrid 3D NoC is being analyzed. Packet latency of the hybrid 3D NoC is shown in the Fig.7.19. The detailed description

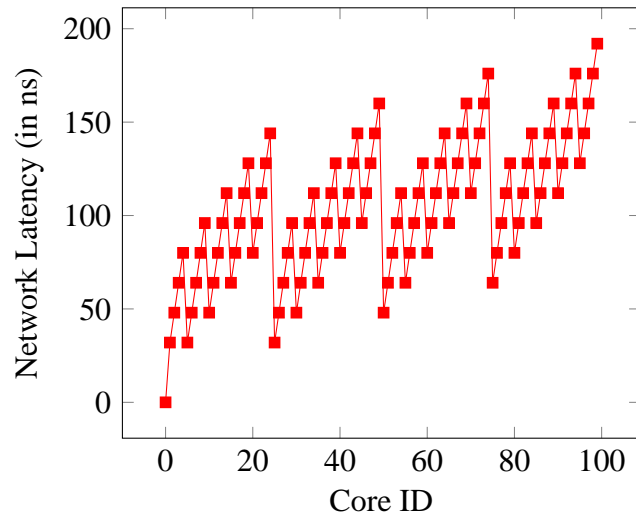


Figure 7.13: Network Latency in 3D Mesh Topology

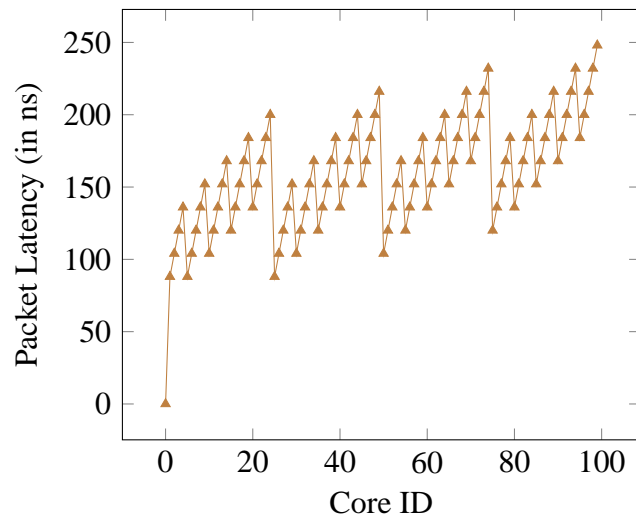


Figure 7.14: Packet Latency in 3D Mesh Topology

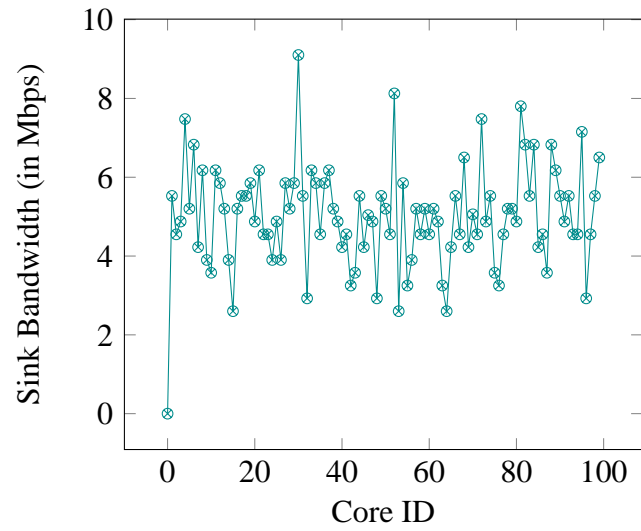


Figure 7.15: Sink Bandwidth in 3D Mesh Topology

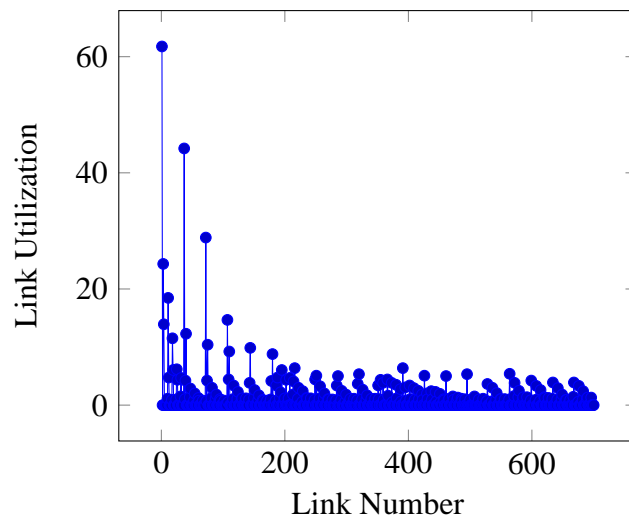


Figure 7.16: Link Utilization in 3D Mesh Topology

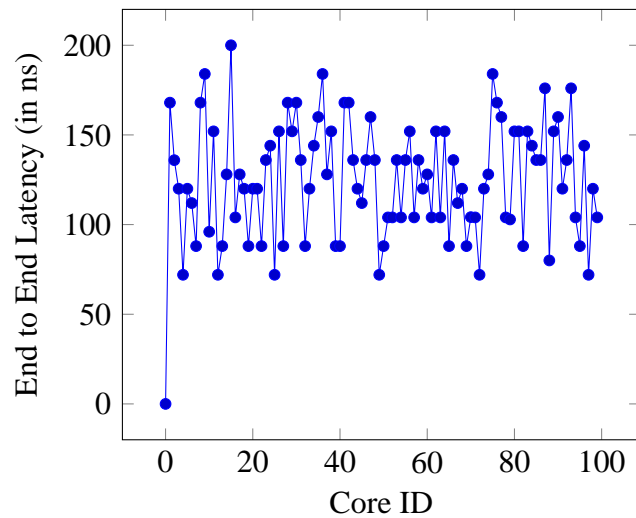


Figure 7.17: End to End Latency in 3D hybrid CBCT

of the sink bandwidth is given in the Fig.7.20 and link utilization of the hybrid CBCT 3D networks on chip topology is given in the Fig.7.21.

### 7.1.3 Results of 3D Layered topology

Fig.7.22 gives the detailed description of the end-to-end latency in the 3D layered CBCT topology for NoC. In Fig.7.23 the network latency of the proposed layered 3D NoC topology is being analyzed. Packet latency of the layered 3D NoC CBCT topology is shown in the Fig.7.24. The detailed description of the sink bandwidth is given in the Fig.7.25 and link utilization of the layered CBCT 3D networks on chip topology is given in the Fig.7.26.

### 7.1.4 Comparison of Results

Fig.7.27 gives the comparison of the results for end-to-end latency in the 3D mesh topology, 3D hybrid topology and 3D layered CBCT topology for NoC. In Fig.7.28 shows the comparison of the results of network latency of the proposed algorithms with 3D mesh topology. Comparison of the packet latency of the three approaches is shown in the Fig.7.29. The detailed comparison of the sink bandwidth is given in the Fig.7.30 and link utilization of the layered CBCT 3D networks on chip topology is given in the Fig.7.31.

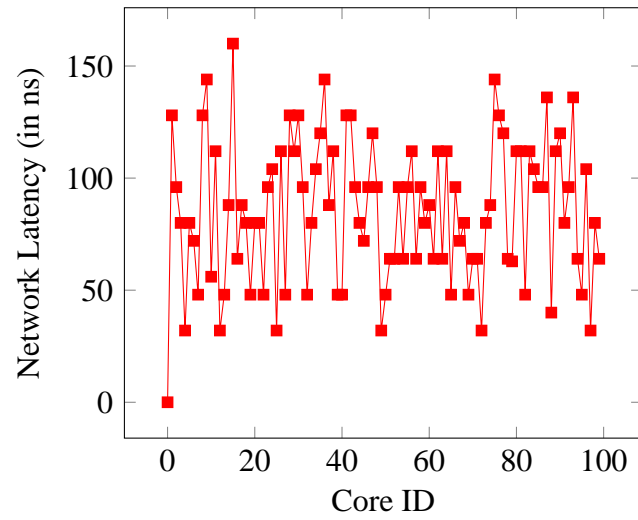


Figure 7.18: Network Latency in 3D hybrid CBCT

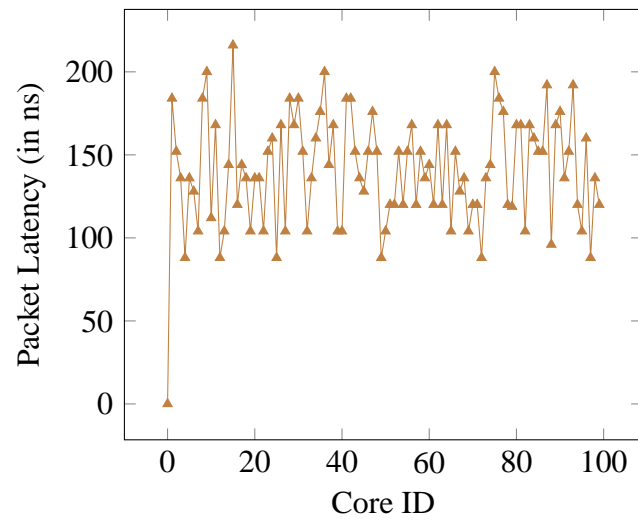


Figure 7.19: Packet Latency in 3D hybrid CBCT

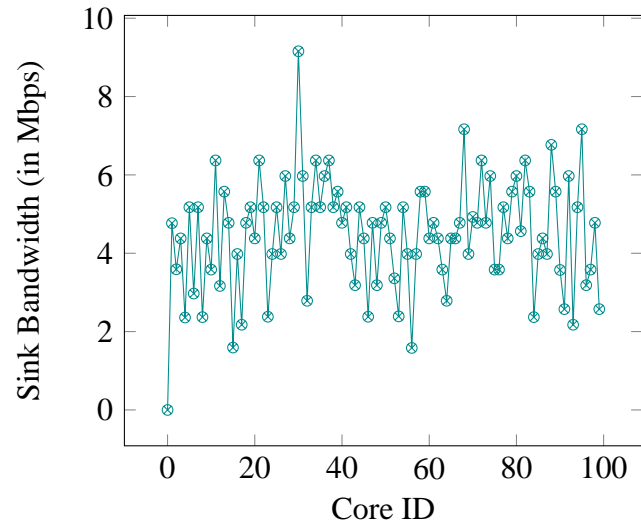


Figure 7.20: Sink Bandwidth in 3D hybrid CBCT

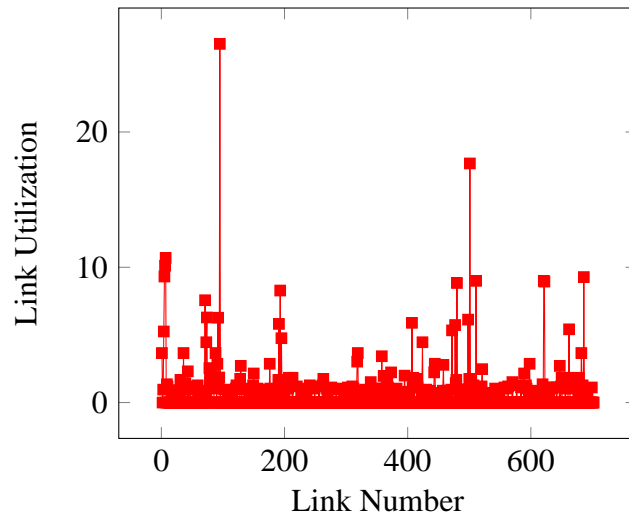


Figure 7.21: Link Utilization in 3D hybrid CBCT



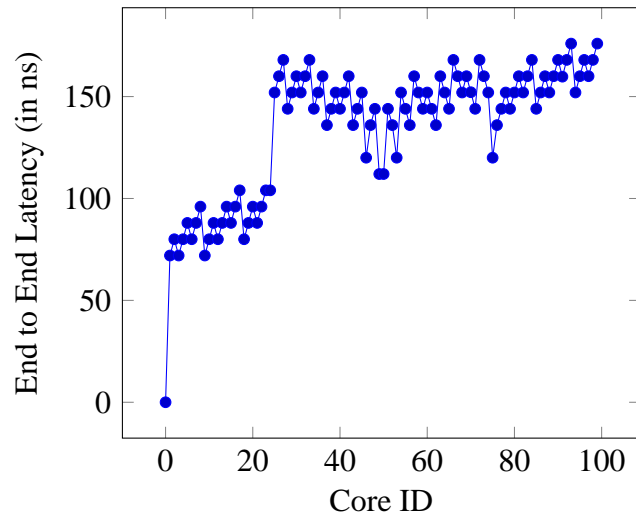


Figure 7.22: End to End Latency in 3D layered CBCT

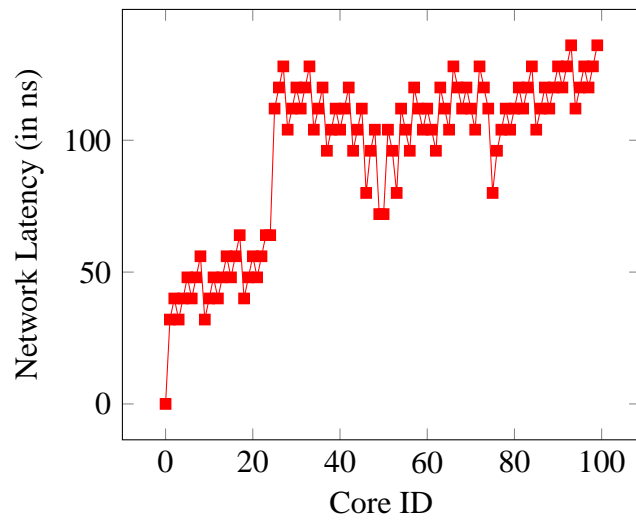


Figure 7.23: Network Latency in 3D 3D layered CBCT

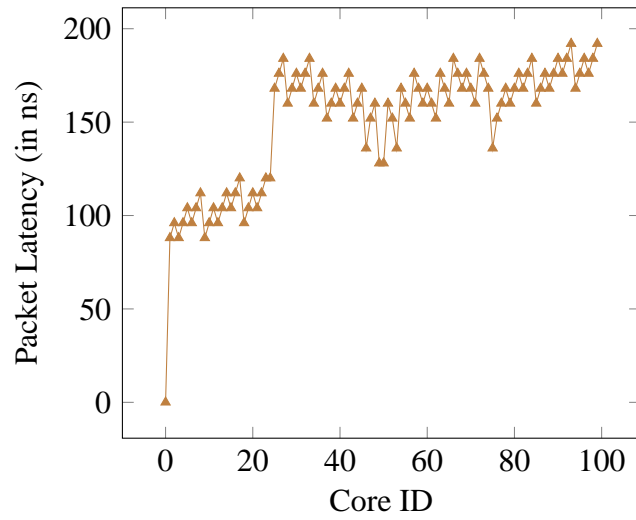


Figure 7.24: Packet Latency in 3D layered CBCT

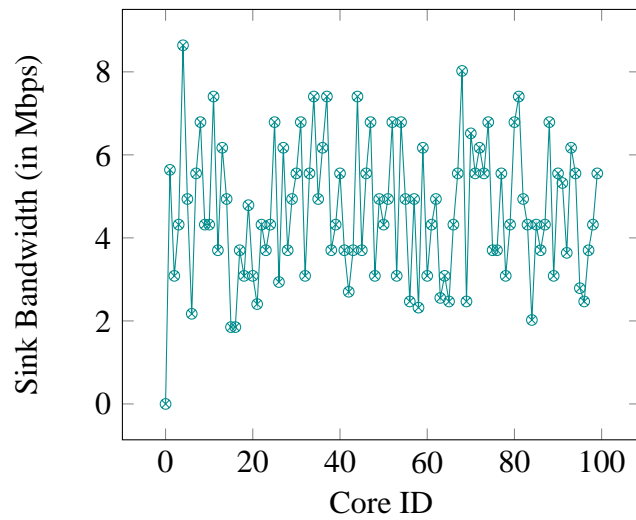


Figure 7.25: Sink Bandwidth in 3D layered CBCT

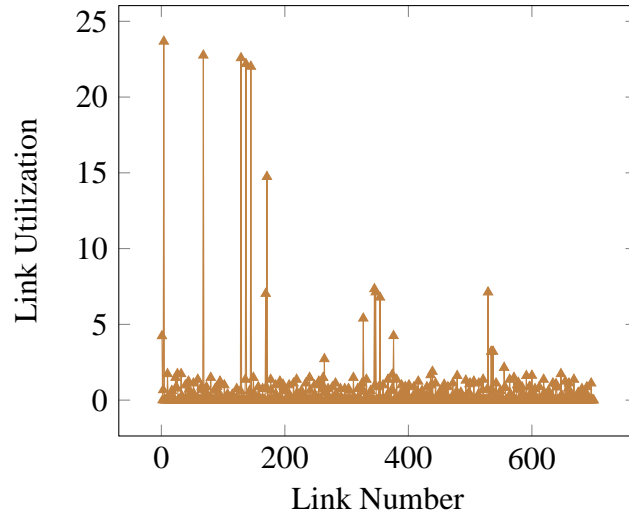


Figure 7.26: Link Utilization in 3D layered CBCT

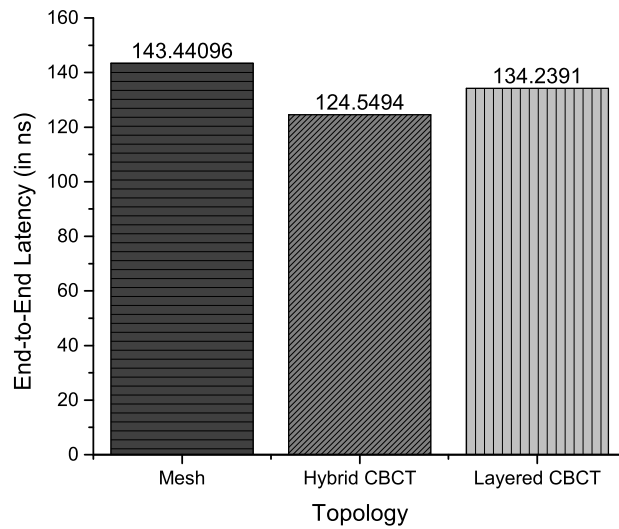


Figure 7.27: Comparison of End-to-End Latency

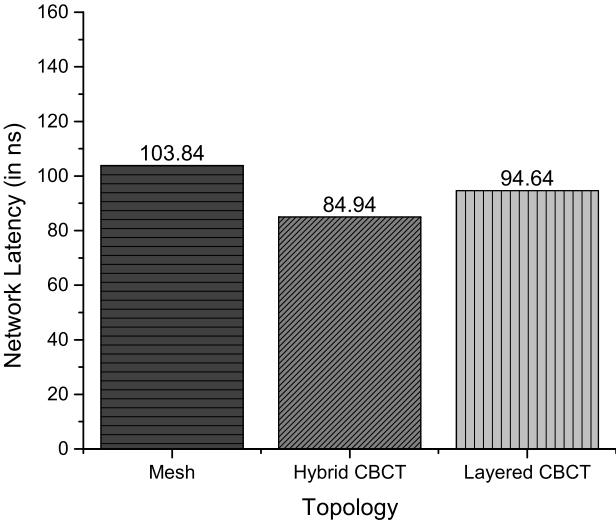


Figure 7.28: Comparison of Network Latency

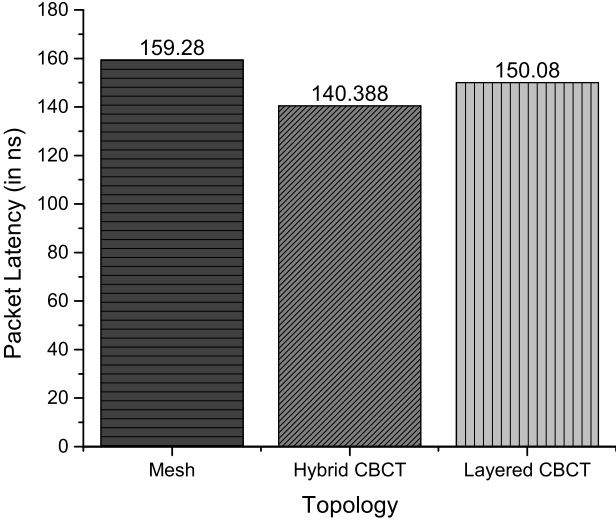


Figure 7.29: Comparison of Packet Latency

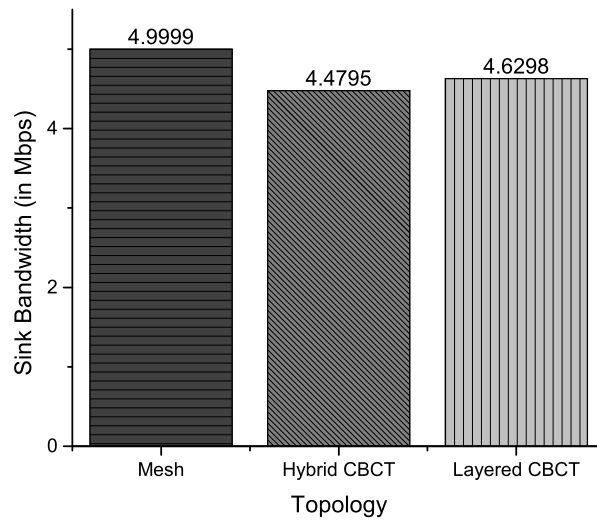


Figure 7.30: Comparison of Sink Bandwidth

### 7.1.5 Results for energy consumption

In this section the comparison of the energy consumption for different cores is being given for the mesh hybrid CBCT topology for 3D NoC and then the comparison of the energy consumption for the mesh topology is being evaluated and shown in the graphical form. Fig.7.32-7.35 shows the energy consumption for cores  $c_4, c_{57}, c_{88}, c_{95}$  in case of mesh compared to hybrid CBCT 3D NoC topologies. Fig.7.36-7.39 shows the energy consumption for cores  $c_4, c_9, c_{12}, c_{19}$  in case of mesh compared to layered CBCT 3D NoC topologies. For comparison these cores are chosen randomly. And any other core irrespective of these cores can be chosen for comparing energy consumption.

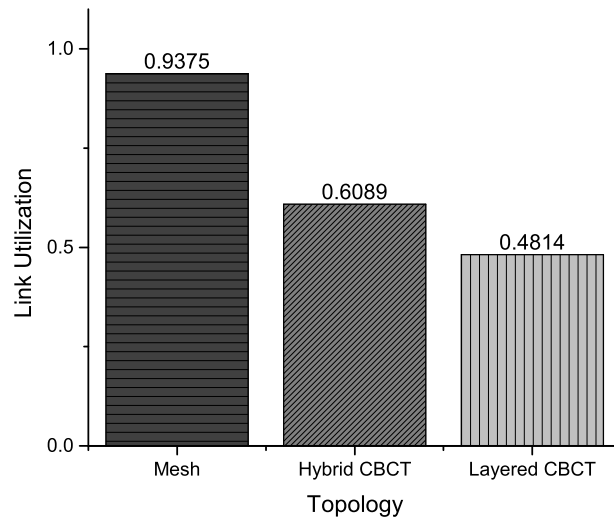
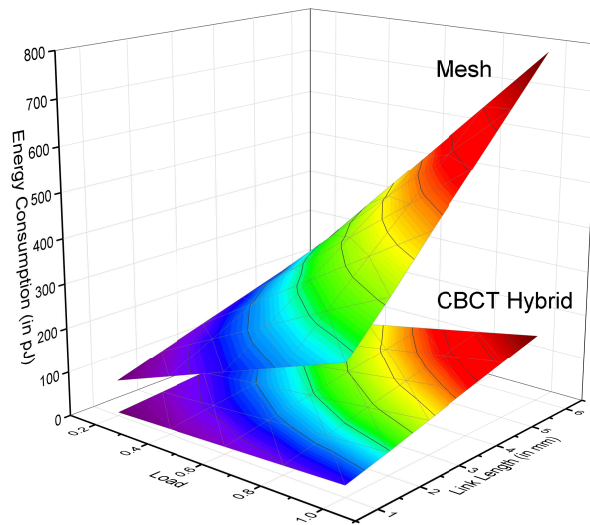


Figure 7.31: Comparison of Link Utilization

Figure 7.32: Comparison of energy consumption (in pJ) in  $c_4$  core (a) 3D Mesh (b) 3D Cluster based Hybrid Topology (clustering of layers of Mesh Topology)

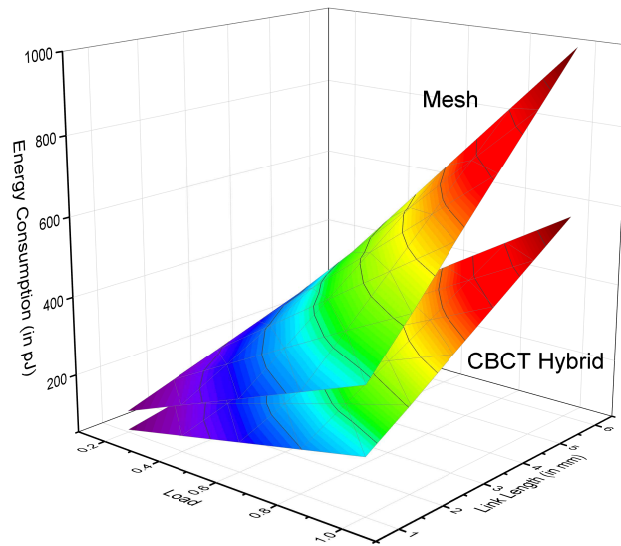


Figure 7.33: Comparison of energy consumption (in pJ) in  $c_{57}$  core (a) 3D Mesh (b) 3D Cluster based Hybrid Topology (clustering of layers of Mesh Topology)

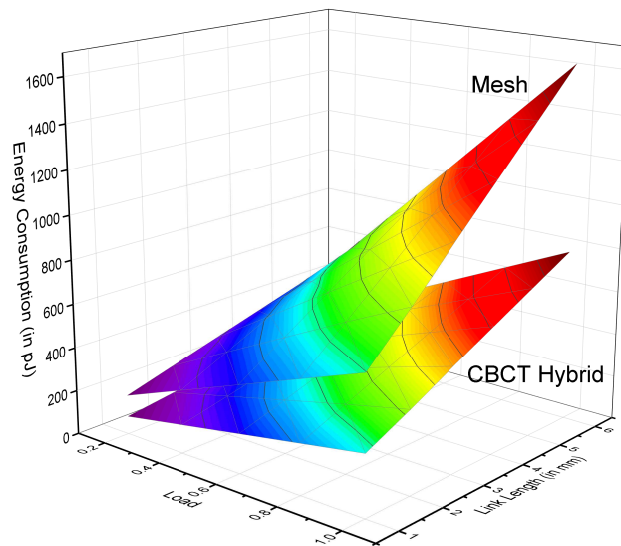


Figure 7.34: Comparison of energy consumption (in pJ) in  $c_{88}$  core (a) 3D Mesh (b) 3D Cluster based Hybrid Topology (clustering of layers of Mesh Topology)

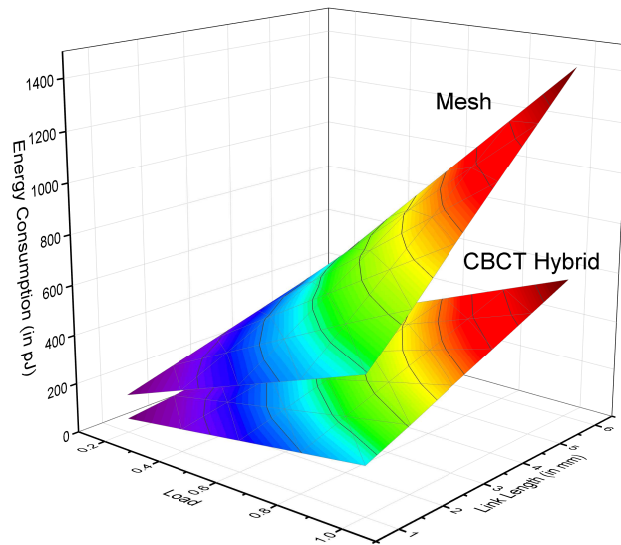


Figure 7.35: Comparison of energy consumption (in pJ) in  $c_{95}$  core (a) 3D Mesh (b) 3D Cluster based Hybrid Topology (clustering of layers of Mesh Topology)

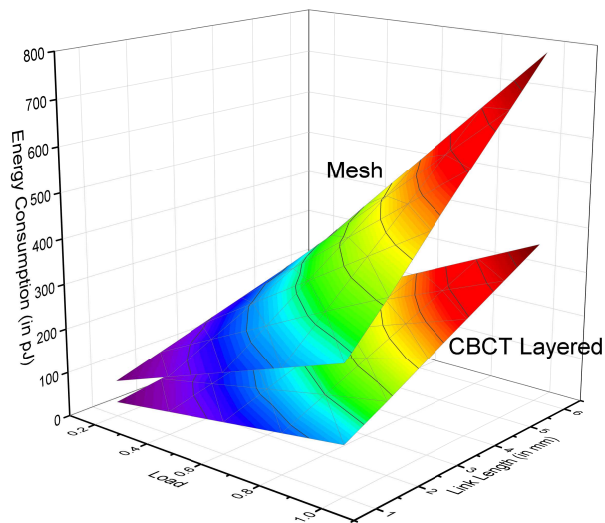


Figure 7.36: Comparison of energy consumption (in pJ) in  $c_4$  core (a) 3D Mesh (b) 3D Cluster based Layered Topology (clustering of layers of Mesh Topology)



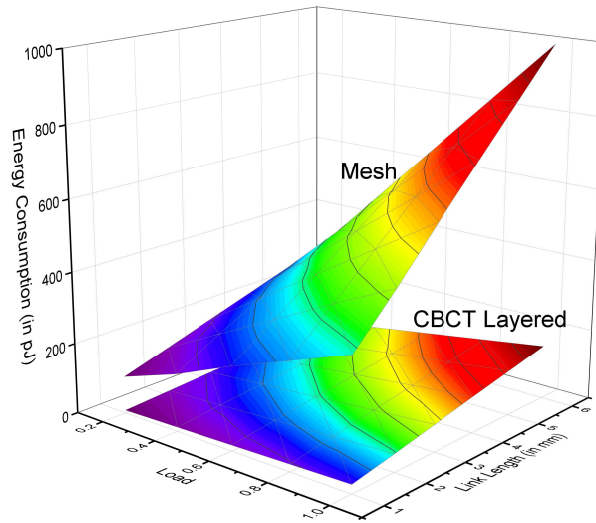


Figure 7.37: Comparison of energy consumption (in pJ) in  $c_9$  core (a) 3D Mesh (b) 3D Cluster based Layered Topology (clustering of layers of Mesh Topology)

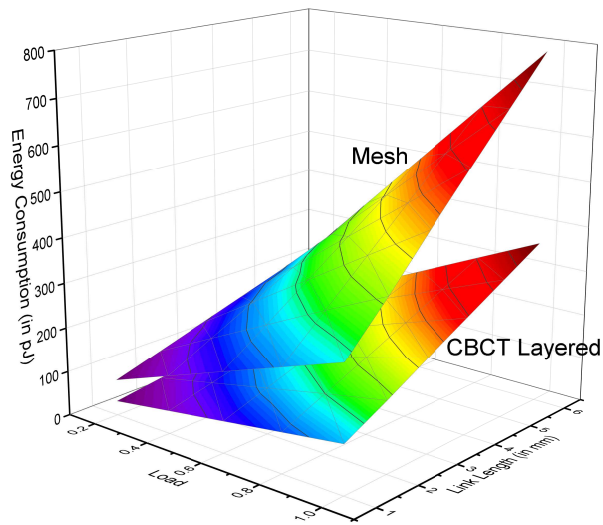


Figure 7.38: Comparison of energy consumption (in pJ) in  $c_{12}$  core (a) 3D Mesh (b) 3D Cluster based Layered Topology (clustering of layers of Mesh Topology)

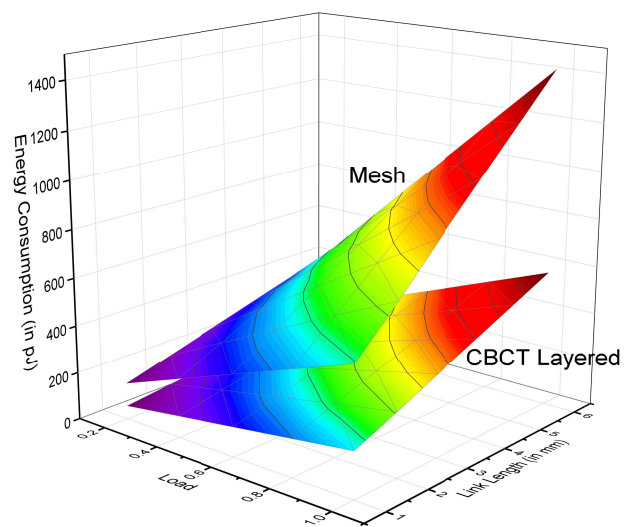


Figure 7.39: Comparison of energy consumption (in pJ) in  $c_{19}$  core (a) 3D Mesh (b) 3D Cluster based Layered Topology (clustering of layers of Mesh Topology)

# Chapter 8

## Conclusion

CBCT approach proves to be better than the mesh topology in terms of end-to-end latency, network latency, packet latency, loss probability, link utilization, energy consumption of topology and processing speed at minimum bandwidth required. Using CBCT, the hybrid topology generated proves to be 40% to 80% more efficient than the existing NoC topologies. CBCT provides the researchers an opportunity to customize the topology according to the requirements of the system. Based on the communication between the cores, most communicating cores are kept in the same cluster. Depending on the number of cores and communication between the cores of the cluster, topology for the cluster is decided. According to the chosen topology, best suited routing algorithm is used for that cluster. If different permutations and combinations of the cores are used to build a cluster, than different level of optimized topology can be obtained. So, this approach provides an opportunity to develop most optimized hybrid and heterogeneous topology as per the conditions and requirements.

## **Chapter 9**

### **Future Work**

- Comparison of results with other existing regular topologies such as Torus.
- Development of an analytical/mathematical model for the proposed approach.

# Bibliography

- [1] N. Viswanathana, K. Paramasivamb, K. Somasundaramc, **Exploring Hierarchical, Cluster based 3D Topologies for 3D NoC**, International Conference on Communication Technology and System Design, Coimbatore, Vol. 30, pp. 606-615, 9 December 2011.
- [2] R.Dafali, J-Ph.Diguët and M.Sevaux, **Key Research Issues for Reconfigurable Network-on-Chip**, International Conference on Reconfigurable Computing and FPGAs, Cancun, pp. 181-186, 3 December 2008.
- [3] W. J. Dally, B. Towles, **Route Packet, Not Wires: On-Chip Interconnection Networks**, Proceedings of the 38th annual Design Automation Conference, Las Vegas, Nevada, USA, pp. 684-689, June 2001.
- [4] L. Benini, G. De Micheli, **Networks on chips: A new SoC paradigm**, IEEE Computer Journal, Vol. 35, No. 1, pp. 70-78, January 2002.
- [5] M. B. Taylor, Jason Kim, Jason Miller, David Wentzlaff, Fae Ghodrât, Ben Greenwald, Henry Hoffman, Paul Johnson, Jae-Wook Lee, Walter Lee, Albert Ma, Arvind Saraf, Mark Seneski, Nathan Shnidman, Volker Strumpfen, Matt Frank, Saman Amarasinghe, Anant Agarwal, **The RAW microprocessor: A Computational Fabric for Software Circuits and General-Purpose Programs**, IEEE Micro Journal, Vol. 22, No. 6, pp. 25-35, March 2002.
- [6] K. Sankaralingam, Ramadass Nagarajan, Haiming Liu, Changkyu Kim, Jaehyuk Huh, Doug Burger, Stephen W. Keckler, Charles R. Moore, **Exploiting ILP, TLP, and DLP with the Polymorphous TRIPS Architecture**, IEEE Micro Journal, Vol. 23, No. 6, pp. 46-51, 2003.
- [7] Jingcao Hu, Radu Marculescu, **Energy Aware Mapping for Tile Based NoC Architecture Under Performance Constraints**, Proceedings of the 2003 Asia and South Pacific Design Automation Conference, pp.233-239, January 2003.

- [8] S. Murali, G. De Micheli, **Bandwidth Constrained Mapping of Cores onto NoC Architectures**, Proceedings of the conference on Design, automation and test in Europe, Vol. 2, pp. 896-901, February 2004.
- [9] K. Srinivasan, K. S. Chatha, G. Konjevod, **Linear Programming Based Techniques for Synthesis of NoC Architecture**, IEEE Transactions on VLSI Systems, Vol. 14, No. 4, pp. 407-420, April 2006.
- [10] S. Murali, Paolo Meloni, Federico Angiolini, David Atienza, Salvatore Carta, Luca Benini, Giovanni De Micheli, Luigi Raffo, **Designing Application-Specific Networks on Chips with Floorplan Information**, Proceedings of the 2006 IEEE/ACM international conference on Computer-aided design, San Jose, CA, pp.355-362, November 2006.
- [11] S. Yan, B. Lin, **Application-specific network-on-chip architecture synthesis based on set partitions and Steiner trees**, Proceedings of the 2008 Asia and South Pacific Design Automation Conference, Seoul, pp. 277-282, March 2008.
- [12] S. Yan, B. Lin, **Custom Networks-on-Chip Architectures with Multicast Routing**, IEEE Transaction on VLSI Systems, Vol. 17, No. 3, pp. 342-355, March 2009.
- [13] A. Habibi, M. Arjomand, H. Sarbazi-Azad, **Multicast-Aware Mapping Algorithm for On-chip Networks**, 19th International Euromicro Conference on Parallel, Distributed and Network-Based Processing, Ayia Napa, pp. 455-462, February 2011.
- [14] G. Leary, Karam S. Chatha, **Design of NoC for SoC with Multiple Use Cases Requiring Guaranteed Performance**, 23rd International Conference on VLSI Design, Bangalore, pp. 205-205, January 2010.
- [15] R. Kumar, V. Zyuban, and D. M. Tullsen, **Interconnections in Multicore Architectures: Understanding Mechanisms, Overheads and Scaling**, Proc. of the 32nd Int. Sym. on Comp. Arch., Vol. 33, No. 2, pp. 408-419, Madison, USA, 2005.
- [16] A. Ben Abdallah, M. Sowa, **Basic Network-on-Chip Interconnection for Future Giga-scale MPSoCs Applications: Communication and Computation Orthogonalization**, Proceedings Of The Tunisia-Japan Symposium on Society, Science and Technology: Partners in Knowledge, pp. 1-7, Dec. 2006.
- [17] Brett Stanley Feero, Partha Pratim Pande, **Networks-on-Chip in a Three-Dimensional Environment: A Performance Evaluation**, IEEE Transactions on Computers, Vol. 58, No. 1, pp. 32-45, 2009.
- [18] K. Tatas, K. Siozios, D. Soudris, A. Jantsch, **Designing 2D and 3D Network-on-Chip Architectures**, Springer-Verlag New York, 2014.

- [19] Giovanni De Micheli, Luca Benini, **Networks on Chips: Technology and Tools**, Morgan Kaufmann Publisher, Editor Luca Benini, Chapter 5, pp. 147-195, 20 Jul 2006.
- [20] Sudeep Pasricha, Nikil Dutt, **On-Chip Communication Architectures :System on Chip Interconnect**, Morgan Kaufmann Publisher, Editor Charles B. Glaser, Chapter 12, pp. 439-466, April 2008.
- [21] **Routing Algorithms in Networks-on-Chip**, Springer-Verlag New York, Eds. Maurizio Palesi, Masoud Daneshtalab, 2014.
- [22] Yongfeng Xu , Jianyang Zhou, Shunkui Liu, **Research and Analysis of Routing Algorithms for NoC**, 3rd International Conference on Computer Research and Development, Shanghai, Vol. 2, pp. 98-102, 11 March 2011.
- [23] Ankur Agarwal, Cyril Iskander, Ravi Shankar, **Survey of Network on Chip (NoC) Architectures and Contributions**, Journal of Engineering, Computing and Architecture, Vol. 3, No. 1, pp. 1-15, 2009.
- [24] A. DeOrio, D. Fick, V. Bertacco, D. Sylvester, D. Blaauw, Hu Jin, G. Chen, **A Reliable Routing Architecture and Algorithm for NoCs** , IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Vol. 31, No. 5, pp. 726-739, 2012
- [25] Dara Rahmati, Hamid Sarbazi-Azad, Shaahin Hessabi, Abbas Eslami Kiasari, **Power-Efficient Deterministic and Adaptive Routing in Torus Networks-on-Chip**, Microprocessors and Microsystems, Vol. 36, No. 7, pp. 571-585, 2012.
- [26] Santanu Kundu, Santanu Chattopadhyay, **Mesh-of-tree Deterministic Routing for Network-on-chip architecture**, Proceedings of the 18th ACM Great Lakes symposium on VLSI, pp. 343-346, 2008.
- [27] Zhu Haibo, P.P. Pande, C. Grecu, **Performance Evaluation of Adaptive Routing Algorithms for achieving Fault Tolerance in NoC Fabrics**, IEEE International Conf. on Application -specific Systems, Architectures and Processors , Montreal, Que., pp. 42-47, July 2007.
- [28] Terrence Mak, Peter Y. K. Cheung, Kai-Pui Lam, Wayne Luk, **Adaptive Routing in Network-on-Chips Using a Dynamic-Programming Network**, IEEE Transactions on Industrial Electronics, Vol. 58, No. 8, pp. 3701-3715, August 2011.
- [29] Liu Jian, L.R. Zheng, H. Tenhunen, **A Circuit-Switched Network Architecture for Network-on-Chip**, IEEE International SOC Conference, pp. 55-58, Sept. 2004.
- [30] Angelo Kuti Lusala, Jean-Didier Legat, **Combining SDM-Based Circuit Switching with Packet Switching in a Router for On-Chip Networks**, International Journal of Reconfigurable Computing, Vol. 2012, Article ID 474765, pp. 1-16, 2012.

- [31] Leonel P. Tedesco, Ney Calazans, Fernando Moraes, **Buffer Sizing for Multimedia Flows in Packet-Switching NoCs**, Journal Integrated Circuits and System, Vol. 3, No. 1, pp. 46-56, 2008.
- [32] T. Pionteck, C. Albrecht, R. Koch, **A Dynamically Reconfigurable Packet-Switched Network-on-Chip**, Design, Automation and Test in Europe, Munich, Vol. 1, March 2006.
- [33] Yogita A. Sadawarte, Mahendra A. Gaikwad, Rajendra M. Patrikar, **Implementation of Virtual Cut-Through Algorithm For Network on Chip Architecture**, International Journal of Computer Applications, pp. 5-8, 2011.
- [34] L. Rooban, S. Dhananjeyan, **Design of Router Architecture Based on Wormhole Switching Mode for NoC**, International Journal of Scientific and Engineering Research, Vol. 3, No. 3, pp. 1-5, March 2012.
- [35] Faizal A. Samman, Thomas Hollstein, Manfred Glesner, **Networks-On-Chip Based on Dynamic Wormhole Packet Identity Mapping Management**, VLSI Design, Vol. 2009, Article ID 941701, pp. 1-15, January 2009.
- [36] S. Swapna, A.K. Swain, K.K. Mahapatra, **Design and Analysis of Five Port Router for Network on Chip**, Asia Pacific Conference on Postgraduate Research in Microelectronics and Electronics, Hyderabad, pp. 51-55, Dec. 2012.
- [37] Anuprita.S. Kale, Prof. M.A. Gaikwad, **Design and Analysis of On-Chip Router for Network on Chip**, International Journal of Emerging Technology and Advanced Engineering, Hyderabad, Vol. 2, No. 1, pp. 143-147, January 2012.
- [38] Seyyed Amir Asghari, Hossein Pedram, Mohammad Khademi, Pooria Yaghini, **Designing and Implementation of a Network on Chip Router Based on Handshaking Communication Mechanism**, World Applied Sciences Journal, Vol. 6, No. 1, pp. 88-93, 2009.
- [39] P. Arivu, N. Shanmugasundaram, C. Kamalanathan, Dr.S. Valarmathi, **Design of Synchronous NoC Router for System-on-Chip Communication and Implement in FPGA using VHDL**, International Journal of Computer Science and Mobile Computing, Vol. 2, No. 12, pp. 308-317, 2013.
- [40] Mike Brugge, Mohammed A. S. Khalid, **Parameterizable NoC Router for FPGAs**, Journal of Computers, Vol. 9, No. 3, pp. 519-529, 2014.
- [41] Rajesh Nema, Teena Raikwar, Prerna Suryavanshi, **Advance NOC Router with LOW Latency and Low Power Consumption by Wormhole Switching**, International Journal of Recent Technology and Engineering (IJRTE), Vol. 1, No. 6, pp. 5-7, 2013



- [42] Umit Y. Ogras, Jingcao Hu, Radu Marculescu, **Key Research Problems in NoC Design: A Holistic Perspective**, Third IEEE/ACM/IFIP International Conference on Hardware/Software Codesign and System Synthesis, Jersey City, NJ, USA, pp. 69-74, Sept. 2005.
- [43] Vitor de Paulo and Cristinel Ababei, **3D Network-on-Chip Architectures Using Homogeneous Meshes and Heterogeneous Floorplans**, International Journal of Reconfigurable Computing - Special issue on selected papers from ReconFig 2009 International conference on reconfigurable computing and FPGAs (ReconFig 2009), Vol. 2010, Article 1, January 2010.
- [44] Tapani Ahonen, David A. Siguenza-Tortosa, Hong Bin, Jari Nurmi, **Topology Optimization for Application-Specific Networks-on-Chip**, Proceedings of the 2004 international workshop on System level interconnect prediction, Paris, France, pp. 53-60, February 2004.
- [45] Vasilis F. Pavlidis, G. Friedman, **3-D Topologies for Networks-on-Chip**, IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Vol. 15, No. 10, pp. 1081-1090, Sept. 2007.
- [46] Marcio Kreutz, Cesar Marcon, Luigi Carro, Ney Calazans and Altamiro A. Susin, **Energy and Latency Evaluation of NoC Topologies**, IEEE International Symposium on Circuits and Systems, Vol. 6, pp. 5866-5869, May 2005.
- [47] Victor Dumitriu and Gul N. Khan, **Throughput-Oriented NoC Topology Generation and Analysis for High Performance SoCs**, IEEE Transactions On Very Large Scale Integration Systems, Vol. 17, No. 10, pp. 1433-1446, October 2009.
- [48] Yaniv Ben-Itzhak, Eitan Zahavi, Israel Cidon, Avinoam Kolodny, **HNOCS: Modular Open-Source Simulator for Heterogeneous NoCs**, International Conference on Embedded Computer Systems, Samos, pp. 51-57, July 2012.
- [49] Andrew B. Kahng, Bin Li, Li-Shiuan Peh, Kambiz Samadi, **ORION 2.0: A Power-Area Simulator for Interconnection Networks**, IEEE Transactions on Very Large Scale Integration, Vol. 20, No. 1, pp. 191 - 196, March 2011.

## List of Publications

- **Suchi Johari**, Arvind Kumar, "*Algorithmic Approach for applying Load Balancing During Task Migration in Multi-core System*", International Conference on Parallel, Distributed and Grid Computing, pp. 27-32, December 2014 (*Published*).
- **Suchi Johari**, Arvind Kumar, Vivek Kumar Sehgal, "*Heterogeneous and Hybrid Clustered Topology for Networks-on-Chip*", Seventh International Conference on Computational Intelligence, Communication Systems and Networks, Riga, Latvia, 2015 (*Accepted*).
- Arvind Kumar, **Suchi Johari**, Vivek Kumar Sehgal, "*Arbitrated IP Core Based Dynamic Task Mapping Algorithm for Networks-on-Chip*", Seventh International Conference on Computational Intelligence, Communication Systems and Networks, Riga, Latvia, 2015 (*Accepted*).