Performance Evaluation of Different Routing Algorithms in Network on Chip

A Thesis submitted in partial fulfilment of the requirements for the degree of Master of Technology

in

Electronics and Communication Engineering

(VLSI Design and Embedded System)

By

Jayant Kumar Singh

Roll No. 212ec2147

NATIONAL INSTITUTE OF TECHNOLOGY, ROURKELA

राष्ट्रीय प्रौद्योगिकी संस्थान, राउरकेला

PIN-769008

ODISHA, INDIA

Performance Evaluation of Different Routing Algorithms in Network on Chip

A Thesis submitted in partial fulfilment of the requirements for the degree of

Master of Technology in

Electronics and Communication Engineering

(VLSI Design and Embedded System)

By

Jayant Kumar Singh

Roll No. 212ec2147

Under the Supervision of

Prof. Ayas Kanta Swain

NATIONAL INSTITUTE OF TECHNOLOGY, ROURKELA

राष्ट्रीय प्रौद्योगिकी संस्थान, राउरकेला

PIN-769008

ODISHA, INDIA

DEDICATION

This thesis is dedicated to my parents, faculties and friends whose support have enabled me to write this thesis.

DEPT. OF ELECTRONICS AND COMMUNICATION ENGINEERING
NATIONAL INSTITUTE OF TECHNOLOGY, ROURKELA
ROURKELA – 769008, ODISHA, INDIA

CERTIFICATE

This is to certify that the Thesis entitled, " **Performance Evaluation of Different Routing Algorithms in Network on Chip**" submitted by " **Jayant Kumar Singh"** bearing **Roll No. 212ec2147** to the National Institute of Technology Rourkela is a bonafide research work carried out by him under my guidance and is in partial fulfilment of the requirements for the award of the degree of "**Master of Technology"** in Electronics and Communication Engineering specializing in " **VLSI Design and Embedded System** " from this institute. The embodiment of this thesis is not submitted in any other university and/or institute for the award of any degree or diploma to the best of our knowledge and belief.

Date: June 2, 2014

Place: Rourkela

Prof. Ayas Kanta Swain
Department of E. C. E
National Institute of Technology
Rourkela-769008

ACKNOWLEDGEMENT

There are many people who are associated with this project directly or indirectly whose help and timely suggestions are highly appreciable for completion of this project. First of all, I would like to thank Prof. S Meher, Head, Department of Electronics and Communication Engineering for his kind support and constant encouragements, valuable discussions which is highly commendable.

I would like to express my sincere gratitude to my supervisor Prof. Ayas Kanta Swain, for his super vision, encouragement, and support which has been instrumental for the success of this project. It was an invaluable experience for me to be one of his students. Because of him, I have gained a careful research attitude.

My due thanks to Prof. K. K. Mahapatra of the Electronics and Communication Engineering department for their course work which helped me in completing my thesis work. Thanks to those who are also the part of this project whose names could have not been mentioned here.

Lastly, I would also like to thank my parents for their love and affection and especially their courage which inspired me and made me to believe in myself.

Jayant Kumar Singh

Roll No. 212ec2147

ABSTRACT

Network on Chip (NoC) is one of the efficient on-chip communication architecture for System on Chip (SoC) where a large number of computational and storage blocks are integrated on a single chip. NoCs have tackled the disadvantages of SoCs as well as they are scalable. But an efficient routing algorithm can enhance the performance of NoC. In one chapter of the thesis three different types of routing algorithms are compared i.e. XY, OE, and DyAD. XY routing algorithm is a distributed deterministic algorithm. Odd-Even (OE) routing algorithm is distributed adaptive routing algorithm with deadlock-free ability. DyAD is a smart routing algorithm which combines the features of both deterministic and adaptive routing. In another chapter of thesis three different types of deadlock free routing algorithms are compared i.e. one deterministic routing (XY routing algorithm), three partially adaptive routing (West first, North last and Negative first) and two adaptive routing (DyXY, OE) are being compared with % of load for various traffic patterns. In another chapter of thesis, a fault tolerant algorithm is described and its performance is compared with all the deadlock free routing algorithms in a NoC having link faults and node faults. All these simulation is done in NIRGAM 2.1 simulator which is a cycle accurate systemC based simulator.

Contents

ABSTRACT ii
List of figure vi
List of table viii
Chapter 1 Introduction 1
1.1 Evolution of On chip Networks 2
1.1.1 Point to Point Communication 2
1.1.2 Shared Bus System 3
1.1.3 Network on Chip 4
1.2 Literature Review 5
1.3 Motivation 6
1.4 Thesis Objective 7
1.5 Thesis Layout 7
Chapter 2 Basics of Network on chip 8
2.1 Basic Concepts of NoC 9
2.1.1 Communication Layers 9
2.1.2 Some Network Communication related terms 10
2.2 Main Components of NoC 11
2.2.1 Resources 11
2.2.2 RNI 11
2.2.3 Router 12
2.3 Design Concepts of NoC 14
2.3.1 Topology 14
2.3.2 Switching Techniques 19
2.3.3 Routing Algorithm 23
Chapter 3 Routing Algorithm for NoC 25
3.1 Taxonomy of routing algorithms 26
3.2 Classification of Routing algorithm based on adaptivity 28
3.2.1 Deterministic Dimension-Ordered Routing 28
3.2.2 Oblivious Routing 29
3.2.3 Adaptive Routing 30
3.3 Problems in routing 34
3.3.1 Deadlock 34

3.3.2 Livelock 34
3.3.3 Starvation 34
3.4 Performance parameters for Routing algorithm 34
3.5 Simulator Used: NIRGAM 35
Chapter 4 Performance Evaluation of a Power Efficient Routing Algorithm with Improved Latency and Throughput 37
4.1 XY Routing Algorithm 38
4.2 OE Routing Algorithm 40
4.3 DyAD Routing Algorithm 42
4.4 Experimental Setup in Simulator 43
4.5 Simulation Results 43
4.6 Discussion 50
Chapter 5 Survey of Deadlock-free Routing algorithms in NoC 51
5.1 Deterministic Routing algorithm 52
5.2 Turn model Based Routing algorithms 52
5.2.1 West First Routing Algorithm 52
5.2.2 North Last Routing Algorithm 55
5.2.3 Negative First Routing Algorithm 57
5.2.4 OE Routing Algorithm 59
5.3 Dynamic XY Routing Algorithm 60
5.3.1 Routing Description 60
5.4 Experimental setup 62
5.4 Simulation Results 62
5.4.1 Simulation results for Random uniform traffic 62
5.4.2 Simulation results for Bit reversal traffic pattern 65
5.4.3 Simulation results for Bit shuffle traffic pattern 68
5.5 Discussion 71
Chapter 6 Performance Evaluation of a Fault Tolerant Routing Algorithm in NoC 74
6.1 Introduction to Routing faults in NoC 75
6.2 Fault tolerant Routing algorithm 75
6.3 DyXY based Fault tolerant Routing Algorithm 76
6.3.1 Router Architecture for this Routing algorithm 76
6.3.2 Principle of Algorithm 77
6.3.3 Implementation of routing algorithm 80

6.4 Simulation Results 81
6.5 Discussion 84
Chapter 7 Conclusion and Future work 85
7.1 Conclusion 86
7.2 Future works 86
REFERENCES 87
PUBLICATIONS 90

List of figure

Figure 1.1 Point to point communication system 3
Figure 1.2 Shared Bus System for SoC 4
Figure 1.3 NoC for SoC 5
Figure 2.1 Message composition 11
Figure 2.2 Resource Network Interface 12
Figure 2.3 Generic Router 13
Figure 2.4 Virtual Channel Router 14
Figure 2.5 Mesh Topology 15
Figure 2.6 Torus Topology 16
Figure 2.7 Fat Tree Topology 16
Figure 2.8 Butterfly Topology 17
Figure 2.9 Star Topology 18
Figure 2.10 SPIN 18
Figure 2.11 Network 19
Figure 2.12 Time space diagram for Circuit switched network 20
Figure 2.13 Example of store and forward Technique 21
Figure 2.14 VCT with no delay 22
Figure 2.15 VCT with delay 22
Figure 2.16 Example of DOR, Oblivious and Adaptive routing 24
Figure 3.1 Taxonomy of Routing algorithms 27
Figure 3.2 All possible turns 28
Figure 3.3 X-Y turns 28
Figure 3.4 Valiant's routing algorithm 30
Figure 3.5 Minimal oblivious routing 30
Figure 3.6 Adaptive routing example 31
Figure 3.7 Turn model routing 33
Figure 3.8 Negative first routing example 33
Figure 3.9 Illegal turn model routing and resulting deadlock cycle 33
Figure 3.10 NIRGAM Simulator 36
Figure 4.1 Allowed turns in XY routing 38
Figure 4.2 Path shown for XY routing for Source A to Destination B 39
Figure 4.3 Allowed turns for even column in OE routing 41
Figure 4.4 Allowed turns for odd column in OE routing 41
Figure 4.5 % of Load vs Average latency per packet 44
Figure 4.6 % of Load vs Average Throughput 45
Figure 4.7 % of Load vs Total Network Power 46
Figure 4.8 Average latency per packet for XY routing 46
Figure 4.9 Average Throughput for OE routing 47
Figure 4.10 Average Throughput for XY routing 47
Figure 4.11 Average Latency for OE routing 47
Figure 4.12 Average Latency for DyAD routing 48
Figure 4.13 Average Throughput for DyAD routing 48
Figure 4.14 Total Network Power for XY routing 48

Figure 4.15 Total Network Power for OE routing 49
Figure 4.16 Total Network Power for DyAD routing 49
Figure 5.1 Allowed turns in west first routing 54
Figure 5.2 Example of west first routing 54
Figure 5.3 Allowed turns in north last routing 56
Figure 5.4 Example of North last routing 57
Figure 5.5 Example of Negative first routing 59
Figure 5.6 Allowed turns in Negative first routing 59
Figure 5.7 Router architecture of DyXY routing 61
Figure 5.8 NoC interconnection during DyXY routing 61
Figure 5.9 % of Load vs Average Latency for random uniform traffic 63
Figure 5.10 % of Load vs Average Throughput for Random uniform traffic 64
Figure 5.11 % of Load vs Total Network Power for Random uniform traffic 65
Figure 5.12 % of Load vs Average Throughput for Bit reversal traffic pattern 66
Figure 5.13 % of Load vs Average Latency for Bit reversal traffic pattern 67
Figure 5.14 % of Load vs Total Network Power for Bit reversal traffic pattern 68
Figure 5.15 % of Load vs Average Latency for Bit shuffle traffic pattern 69
Figure 5.16 % of Load vs Average Throughput for Bit shuffle traffic pattern 70
Figure 5.17 % of Load vs Total Network Power for Bit shuffle traffic pattern 71
Figure 6.1 Router Architecture for FT_DyXY Routing Algorithm 77
Figure 6.2 Example of Fault tolerant routing 78
Figure 6.3 Flow diagram of Fault tolerant routing 80
Figure 6.4 Number of faults vs % of Packet loss For Random uniform traffic pattern 81
Figure 6.5 Number of Faults vs % of Packet loss for Bit reversal traffic 82
Figure 6.6 Number of Faults vs % of Packet loss for Bit shuffle traffic 83

List of table

Table 2.1 Summary of various switching techniques 23
Table 4.1 % of Load vs Average Latency 43
Table 4.2 % of Load vs Average Throughput 44
Table 4.3 % of Load vs Total Network Power 45
Table 5.1 % Load vs Average Latency for Random uniform traffic 62
Table 5.2 % of Load vs Average Throughput for Random uniform traffic 63
Table 5.3 % of Load vs Total network power for Random uniform traffic 64
Table 5.4 % of Load vs Average Throughput for Bit reversal traffic pattern 65
Table 5.5 % of Load vs Average Latency for Bit reversal traffic pattern 66
Table 5.6 % of Load vs Total Network Power for Bit reversal traffic pattern 67
Table 5.7 % of Load vs Average Latency for Bit shuffle traffic pattern 68
Table 5.8 % of Load vs Average Throughput for Bit shuffle traffic pattern 69
Table 5.9 % of Load vs Total Network Power for Bit shuffle traffic pattern 70
Table 6.1 Table for determining congestion level 79
Table 6.2 Number of faults vs % of Packet loss For Random uniform traffic pattern 81
Table 6.3 Number of faults vs % of Packet loss for Bit reversal traffic 82
Table 6.4 Number of faults vs % of Packet loss for Bit shuffle traffic 83

Chapter 1

Introduction

The demand of the human race is gradually increasing day by day. People always prefer a small electronic device having many more features in it. Thus VLSI industry found a new a paradigm i.e. System on Chip (SoC). According to this paradigm different electronic or computing systems are embedded on a single chip. Those computing or electronic systems are also called as intellectual property cores. Since the introduction of research into multi-core chips more than a decade ago, on-chip networks have emerged as an important and growing field of research. As core counts increase, there is a corresponding increase in bandwidth demand to facilitate high core utilization and a critical need for scalable interconnection fabrics such as on-chip networks. On-chip networks will be prevalent in computing domains ranging from high-end servers to embedded system-on chip (SoC) devices. This diversity of application platforms has led to research in on-chip networks spanning a variety of disciplines from computer architecture to computer-aided design, embedded systems, VLSI and more. In this chapter evolution of on-chip networks

1.1 Evolution of On chip Networks

In this chapter we are going to discuss the necessity of packet switched on chip networks. Here the discussion will start from various communication infrastructure for a System on Chip and how NoC became so popular in this domain. Here there is some discussion about some common terms in data networks and its relation to the NoC. This discussion will go on with main components and important design concepts of NoC. There are three common communication systems for system on chip i.e. point to point communication and shared bus system.

1.1.1 Point to Point Communication

Previously the designers prefer the direct point to point connection for the communication in system on chip. Here the resources or cores are allowed to communicate directly through wires which are connected to each cores. This system doesn't need any priority providing system or arbitration unit. For a system on chip having more number of cores, this communication system requires large routing area, large routing delay and large number of pins for each core and becomes very complex in wiring point of view. When direct point to point interconnections are used for communication, in this kind of communication system we can detect the quality of signal and delays occurred for routing. So testing of that system is a very tedious job. Due to these above problems, direct point to point interconnection system shows some disadvantages like underutilization of cores or resources, very poor reusability, high complexity and poor scalability. A System on Chip

which has less number of cores or resources can use this communication infrastructure and can give best performance as compared to other systems. The direct point to point connection for a SoC is shown in Figure 1.1.

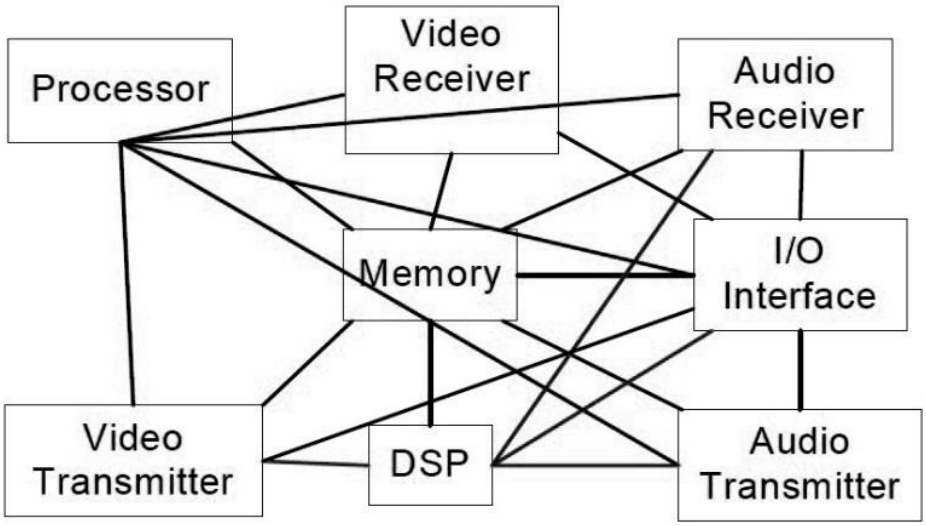


Figure 1.1 Point to point communication system

1.1.2 Shared Bus System

Most of the SoCs uses shared bus system as their inter-core communication system. Here all the cores are connected to one or more than one bus. An interface is used for the connection of the bus to the cores. In this system the communication and contention is managed by a bus arbiter system. Shared bus communication infrastructure requires less input output pins as compared to direct point to point communication system. So wiring area and cost is greatly reduced. There are different kinds of buses present in literatures such as hierarchical, segmented, pipelined buses etc. So there are many advantages of this communication system. But still it has some disadvantages like due to contention and arbitration data movement becomes slow. In scalability point of view this system is not a fair choice as it can be scaled upto certain limits otherwise its efficiency will be very bad. A shared bus is shown in Figure 1.2.

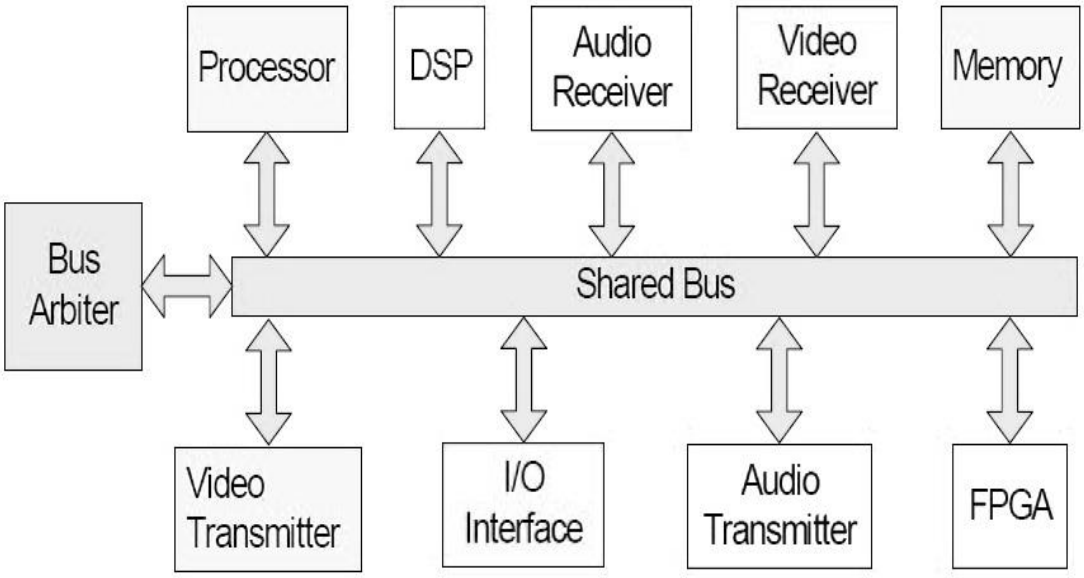


Figure 1.2 Shared Bus System for SoC

1.1.3 Network on Chip

There are many disadvantages of the above two communication systems i.e. less scalability, non-adaptive nature, underutilization of resources and less reuse factor. Many researchers proposed a communication system which can avoid above problems which is termed as Network on Chip shown in Figure 1.3. It consists of three important components i.e. Routers, Resource Network Interface (RNI) and IP cores or resources. IP cores in the NoC are connected to the network switches. RNI (Resource Network Interface) is the communication bridge between the routers and IP cores as routers and IP cores have different communication protocols. For this on chip packet switched network data is converted into some formatted packets and those packets traverse from source to destination with the help of one or more routers in the network. Scalability of this communication system is sufficiently high. It also provides high reusability factor, less complexity and reduced cost.

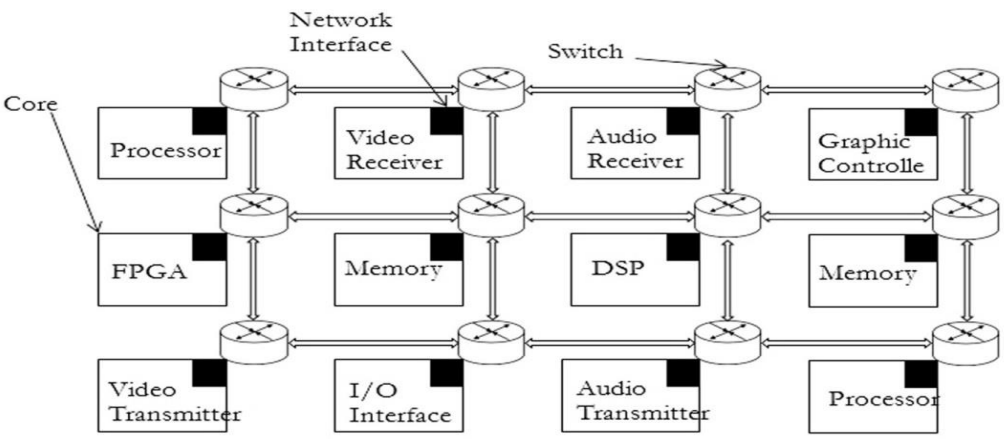


Figure 1.3 NoC for SoC

1.2 Literature Review

Routing algorithm is an important design concept of Network on Chip. The function of routing algorithm is to determine an efficient route for the data or packets to transfer from source to destination. The routing algorithms can be classified in various basis i.e. i) Adaptivity nature ii) Fault tolerance nature iii) Number of destinations. The routing algorithm is a crucial task in network layer. Gratz et al. [1] propose a regional congestion aware routing which calculates the congestion level of the regions near the router and finds an low congested path by selecting less congested links. Ascia et al. [2] introduce Neighbors-an-Path adaptive routing algorithm which uses immediate neighbor's congestion level for adaptive routing and the routers don't use any virtual channels which results in less routing area. Maurizio Palesi et al. proposes a methodology called application specific routing algorithm (APSRA) [3] to develop efficient and deadlock-free routing algorithms for Network-on-Chip (NoC) platforms that are specialized for an application or a set of concurrent applications to maximize communication adaptivity and performance. Andes Mejia et al. proposes the region-based routing (RBR) [4] mechanism which groups destinations into network regions allowing an efficient implementation with logic blocks can and also be viewed as a mechanism to reduce the number of entries in routing tables. Plalesi et al. [5] demonstrates that it is possible to design highly efficient application-specific routing algorithms which distribute traffic more uniformly by using information regarding applications communication behavior (communication topology and communication bandwidth). Rodrigo et al. proposed a new routing method called Logic Based Distributed Routing (LBDR) [6] which removes the need of routing table at all.

LBDR enables the implementation of many routing algorithms on most of the practical topologies we may find in the near future in a multi-core system. Mavevich et al. introduced a novel paradigm of NoC centralized adaptive routing [7] and a specific design for mesh topology. Here this scheme continuously monitors the global traffic load in the network and modifies the routing of packets to improve load balancing accordingly. Flich et al. [8] presented a comprehensive overview of the known topology-agnostic routing algorithms and classified these algorithms by their most important properties, and evaluate them consistently.

1.3 Motivation

Due to rapid development in VLSI industry millions of systems or features can be introduced in an electronic device by fabricating millions of transistors on a single silicon wafer or chip [9]. Now a days chip producers are trying to release the multi-core products with many more cores in the system. This multi-core wave may lead to hundreds and even thousands of cores integrated on a single chip. In addition to the integration of many general-purpose cores on a single chip, increasing transistor counts will lead to greater system integration for multiprocessor systems-on-chip (MPSoCs). A system on chip with large number of IP cores definitely needs a compatible, scalable and high bandwidth communication system. Bus system and crossbar systems came into picture but they failed to provide an efficient communication system. So on chip network came into existence and replaces those two systems with ease. This on-chip network has some switches technically called as routers and routing wires. For various computing domains multi core architecture are very necessary requirement. These architectures will increase the levels of linking capability to the data centers. The applications which are throughput oriented will definitely require a high bandwidth communication. Communication delay or latency can have a substantial role on the performance of multi-threaded systems. Synchronization between threads will require a communication having low overhead for scalability purpose. In MPSoCs, Utilization of an on-chip network can help enable design isolation: MPSoCs utilize heterogeneous IP blocks from a variety of vendors; with standard interfaces, these blocks can communicate with the help of an on-chip network in a plug-and-play fashion. The on-chip networks have some basic building blocks i.e. topology, flow control mechanism, switching techniques, routing algorithms etc. An efficient routing algorithm can enhance the performance of an on-chip network.

1.4 Thesis Objective

- Study of Network on Chip and its design issues.
- Study of different routing algorithms for NoC.
- Study and performance evaluation of a power efficient routing algorithm
- Survey of some deadlock free routing algorithms and their performance comparison for various experimental setups.
- Study of a fault tolerant routing algorithm and evaluate its performance in a NoC having faults.

1.5 Thesis Layout

The general approach of the thesis is to go through the various routing algorithms for regular topologies in Network on Chip. The thesis is organized as follows:

Chapter 2 discusses about Network on Chip, its main components and its building blocks and performance requirements of NoC.

Chapter 3 discusses about the classification of routing algorithms and problems occurred in routing.

Chapter 4 discusses about the performance evaluation of a power efficient routing algorithm with improved latency and throughput.

Chapter 5 includes the survey about some deadlock free routing algorithms and their performance comparison.

Chapter 6 includes the study of a fault tolerant routing algorithm and its performance comparison with other routing algorithms in a faulty NoC.

Chapter 7 discusses about the conclusion and future works.

Chapter 2

Basics of Network on Chip

2.1 Basic Concepts of NoC

NoC follows the simplified rules or protocols of general data communication network. In this section the discussion is about the relation of NoC to the layered communication of OSI model. Some general data network related term are also discussed in this section which have important role in NoC.

2.1.1 Communication Layers

Network on Chip uses layered communication system of OSI model like general data networks [10]. OSI models consists of seven layers i.e. Physical layer, Data link layer, Network layer, Transport layer, Session layer, Presentation layer, Application layer. Here each layer consists of some software with hardware components to perform certain functionality. Here each layer performs a task alone and independently. The layers provides various service to their upper layer and acquires service from their bottom layer. Description of the functionalities of layers (in bottom to top order) of OSI model is given below. Physical layer provides hardware support and is responsible for sending and receiving data on a carrier. In Data link layer data packet encoding and decoding into bits is done. Routing and forwarding of data is the main function of this layer. Transport layer ensures data transfer from source to destination. Session layer sets up co-ordinates and terminates conversations, dialogues between the applications. Presentation layer transforms data into the form that the application layer can accept i.e. encryption of data from application to network format and vice versa. Application layer supports applications and end user processes. When a layer performing a task, it is hidden from other layers. Each layer follows some certain set of rule which are called as protocols and the the layers communicates with each other by some kind of bridge i.e. interfaces. There are many advantages of the layered communication system and it has some unavoidable overheads. But three layers are most important for NoC i.e. i) Physical Layer ii) Data link layer iii) Network layer. Important functionalities of physical layer are providing clock signal to every connection, generates control important functionalities of data link layer is flitization, deflitization, error detection or correction etc. Main functionalities of network layer are data packetization, routing, buffering, congestion detection and control. QoS (Quality of Service) is provided by this layer by improving latency, throughput and jitter of network. In this thesis a sole importance is given to network layer and its functionalities as routing is a task of this network.

2.1.2 Some Network Communication related terms

Some Network communication related terms are described below and shown in Figure 2.1.

Message

Message is the information or data which are transmitted from source to destination resource. It is defined in application layer. A message can be of fixed length or can be of variable length according to requirement. Those messages are travelled in the networks in various forms described below.

Packet

In packetization process message is divided into certain number of packets. Packets of the same message are independent of each other. Each packet has enough info to travel throughout the network. Generally a packet has three parts i.e. i) Header ii) Payload or Body iii) Trailer. Header contains controlling and routing information such as source and destination address. Sometimes it contain the whole route for data transmission. Payload contain actual data or information. Trailer indicates the end of packet.

Flit

A packet can be divided into smaller elements. That small elements are called flits. They are flow control digits. Flits also has three parts i.e. Header flit, Body flit and Tail flit. The size of the flit is always fixed. Due to flits the storage devices required in the switches or routers will be very small. It is a good advantage.

Phit

A flit can be divided into small units known as phits. They are physical transfer digits. It is travelled across a channel among the network switches but as one unit. It can be considered as link width Phit can be considered as the measuring parameter of link width as it indicates the number of wires need for data transfer between network routers. Size of a phit can be same as the size of flit or may be not.

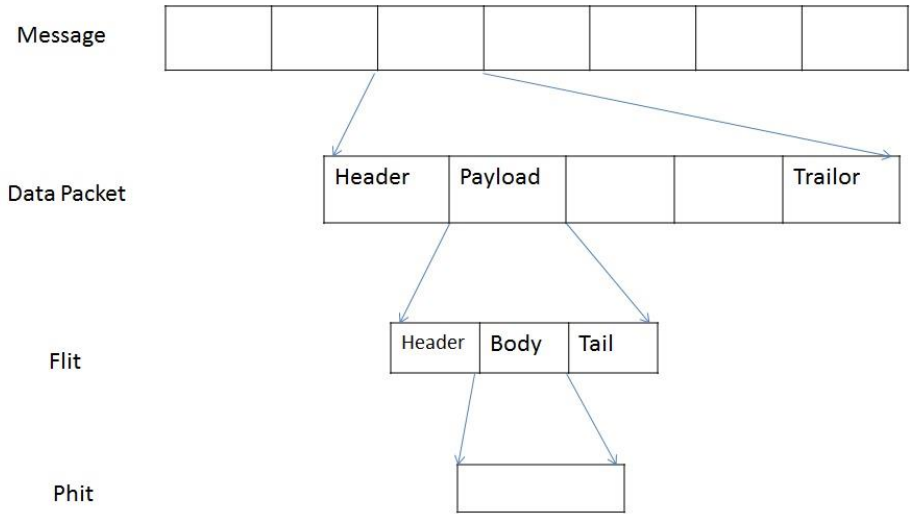


Figure 2.1 Message composition

2.2 Main Components of NoC

A NoC has three main and basic components [11] i.e. i) Network switches technically called as routers ii) Resources or IP cores iii) Resource to Network Interfaces (RNI). These basic parts of NoC are shown in Figure 1.3.

2.2.1 Resources

In a tiled, city-block style of NoC layout, the wires and routers are placed similar to street grids of a city, while the clients (e.g., IP cores or Resources) are placed on city blocks separated by wires. The IP cores or resources can be General Purpose Processors, FPGAs, Amplifiers, ADCs, DSP, memory, Graphic controller, Mixed signal Module, RF unit, application specific hardware component, I/O controller etc. Resource must have the same technology implementation as that of used in NoC. A designer can use own resources rather than buying from different vendors.

2.2.2 RNI

A Resource Network Interface is used to connect an IP core or resource to a router in NoC sown in Figure 2.2. Like that IP cores can transmit message packets to the network switch. Resource Network Interface has two parts which are i) Resource Dependent part ii) Resource Independent part as shown in Figure 2.2. Design of Resource independent part is done in such a way that Resource Network Interface acts as another network switch to the connected network switch. The method of designing resource independent part of RNI is

general kind of procedure. For reusability point of view resource dependent part should be connected to the resource having homogeneous property otherwise this resource dependent part will be different for all resources. The Resource dependent part of RNI has some functionalities like flit formation (flitization), deflitization and applying encoding system. The RNI has of two independent layers of OSI model i.e. i) Session layer ii) Transport layer. Here the functionalities of these two layers are different from the seven layers of OSI model. As per theoretical views, the session layer sets up, coordinates and terminates the conversation between the application hence acts as effective medium for connected IP cores and the transport layer ensures data transfer and operates on the network interface in the communication infrastructure. The transport layer offers a communication services to the upper layer i.e. session layer where message serves as a communication intermediate. The session layer is operated with the help of service provided by the bottom layer i.e. transport layer and connected IP cores are isolated from the communication network infrastructure.

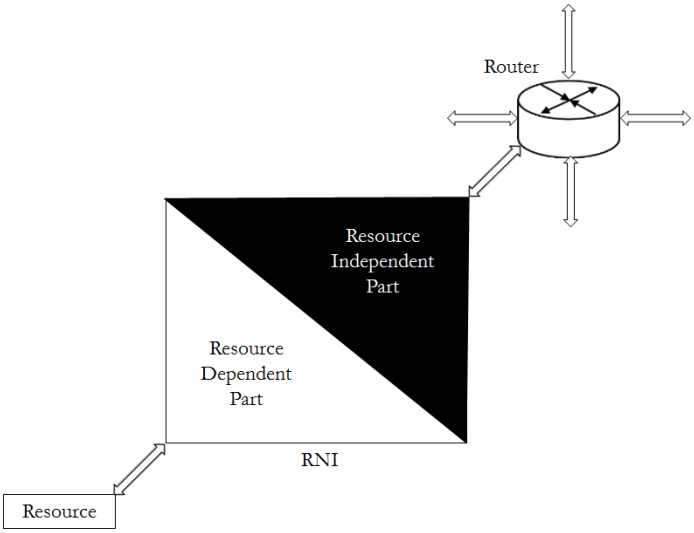


Figure 2.2 Resource Network Interface

2.2.3 Router

Router is nothing more than a switch used in the network shown in Figure 2.3. It is a very important part of the on chip network like any other network infrastructure. They are just like back bone of Network on Chip. In an on chip network, the primary task of a router is to transmit the incoming data to the destination IP core if the router is directly connected to the destination resource otherwise that router has to send it to another router. A router implementation is based on three layers of communication in OSI model i.e. Physical, Datalink and Network layer. A designer should consider the simplicity of a router

and design it like wise so that he can avoid some overheads like cost, area and power. Routing function implementation is the sole purpose of router for distributed routing. For routing purpose a router may contain a routing table which is called as table based router and that table stores the entire route. In another way router implements routing algorithm to calculate the routing path dynamically. The router used for distributed routing is very complex because it needs memory and extra logic to implement entire routing function. A generic router shown in Figure 2.3 consists of five ports i.e. east, west, north, south and local port and a central crosspoint matrix. The first four ports are used to connect to other routers and the local port is used to connect the IP core. In the router every port has an input channel and an output channel. The data packets move into the input channel of a port of router by which it is moved to the output channel of other port. The input and output channels have their own decoding logic which enhances the performance of the router. Buffers work as temporary storage of data. Here the buffering method used is store and forward. Control logic is required to make arbitration decisions. Thus, a communication is set up between the input and output ports. This connection or configuration between these ports is formed by the central crosspoint matrix. In this research virtual channel router is used for the NoC design. This router architecture consists of virtual channel allocator, switch allocator, crossbar switches, route computation unit, input channels and output channels where input and output channels are multiplexed with some virtual channels shown in Figure 2.4.

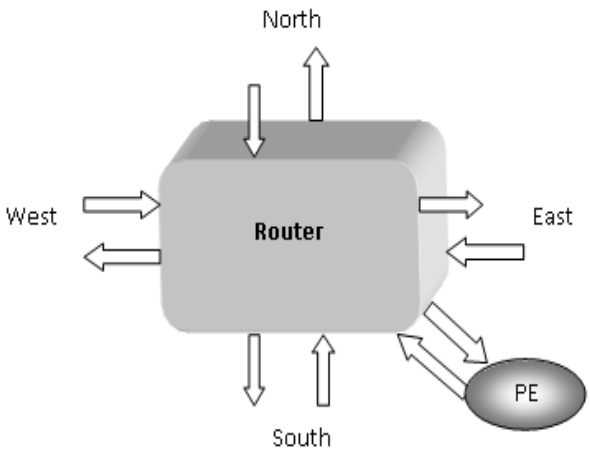


Figure 2.3 Generic Router

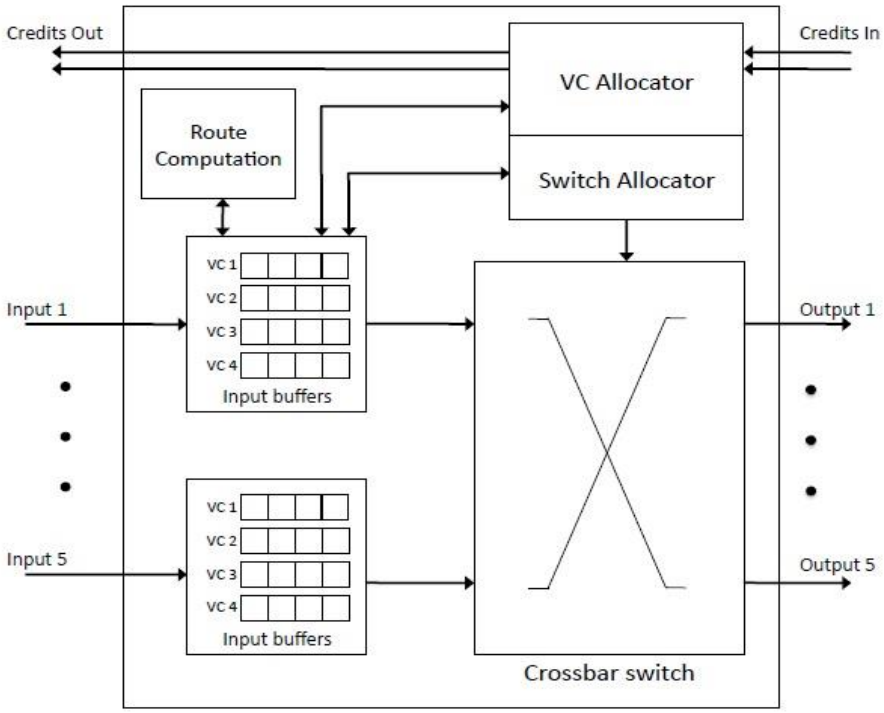


Figure 2.4 Virtual Channel Router

2.3 Design Concepts of NoC

In this section the important design concepts of NoC is described which are backbone of NoC. They are i) Topology ii) Switching techniques iii) Routing algorithms. The performance of NoC depends upon these above concepts. So the researchers keep on researching in these areas from past few years.

2.3.1 Topology

Most of the pattern of interconnection used for Network on Chip used for parallel computing field [12]. According to the SoC paradigm these architectures are different for on chip and off chip networks. An ideal architecture or topology should provide less latency, more throughput and less power consumption, less routing area and less complexity. It is definitely impossible to put all these features in a system because there is tradeoff between these features. So some researches has to sacrifice some advantages for these architectures to gain another one. So there is no architecture which can provide the required and desired performance of researchers. Topology are nothing but arrangement of nodes and channels inside the network. An efficient topology should be selected for a network so that the performance will be improve and it can fulfill the bandwidth and latency

requirement with low cost. Topology can be regular or irregular. A topology can be called non-blocking if it can manage and serve all the requests coming to it. So the researchers proposed various topologies for NoC. Some basic network topologies are described below.

2.3.1.1 Mesh

This is a regular network topology. This type of network consists of P number of rows and Q number of columns. Here all IP cores are connected to their respective router and the routers are connected in each interconnection of wires. Here the address of the router and IP cores are identified by (x,y) co-ordinates of the network. Advantages of Mesh Topology is the easy detection and isolation of faults in the network. They are easy to implement. This the simplest topology among other topologies. It is also called as Manhatten Street Network as it look like a city with streets diagram. This topology is also more protected as the messages go through a dedicated line and the messages will only reach its intended addresses. A 4x4 Mesh network is shown in Figure 2.5.

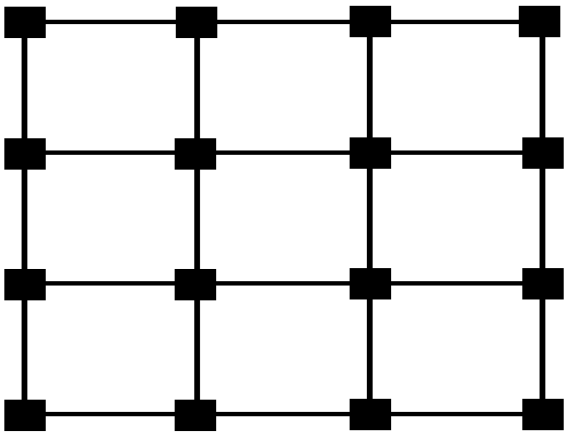


Figure 2.5 Mesh Topology

2.3.1.2 Torus

This topology is similar to the mesh topology. This is also a regular network topology. But the only difference is that end nodes of a column are connected and the end nodes of a row is connected. Due to the long wrap-around channels the packet transmission delay may become significantly long and require usage of repeaters. This can be avoided by folding the torus. Folding is done by shifting all nodes in even rows to the right and all nodes in even positions of each row down, next connecting all the neighboring nodes in

newly gained rows and columns then pair-wise connecting edge nodes in rows and columns. Torus topology has some advantages above mesh topology i.e. torus has better path diversity than mesh and it has more minimal paths than mesh topology. Design is also simple. This is shown in Figure 2.6.

2.3.1.3 Tree

This is one kind of irregular network topology. In this topology there is a root node. This root node consists of some branches and those branch nodes are called child nodes. This nodes are also called leaves. The root node is called ancestor leaf and the branch nodes are also called child leaves. In another kind of tree topology i.e. fat tree topology shown in each leaf has a replicated ancestor which means that there is alternative route present between the leaves. This topology is shown in Figure 2.7.

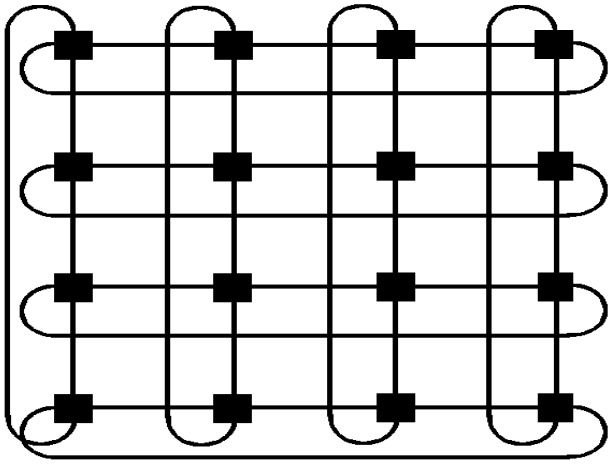


Figure 2.6 Torus Topology

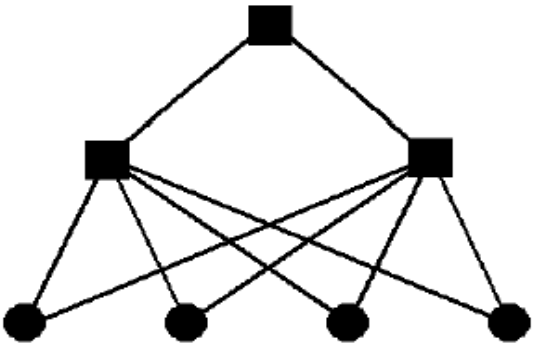


Figure 2.7 Fat Tree Topology

2.3.1.4 Butterfly

This network is of butterfly shape. It may be unidirectional or bi-directional. A unidirectional butterfly network consists of eight input ports, eight output ports and three network switch or router level. Data packets are received by the left side input ports and traversed to the correct right side output port. In bidirectional butterfly network all input and output ports are in the same side. Data packets are received by the input ports and traversed to the other side of network and then turn back and transmitted to the correct output port of the network. A butterfly topology with 4 inputs, 4 outputs and 2 router stages (Each router stage contains 2 routers) are shown in Figure 2.8.

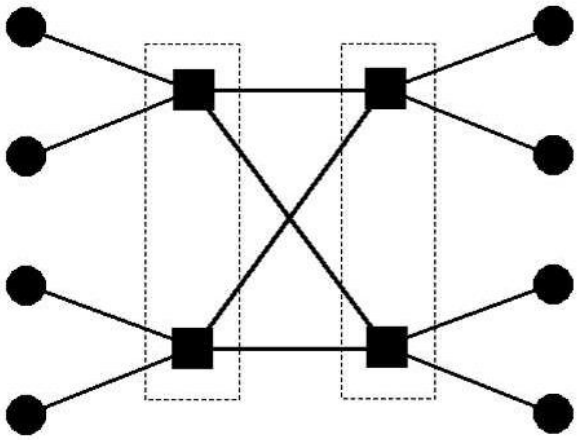


Figure 2.8 Butterfly Topology

2.3.1.5 Star Topology

As per the nomenclature this topology is looking like a simple star sown in Figure 2.9. There is a router in the Centre of the topology. All other IP cores or routers are connected to the spikes of the central router. The advantage of this topology resource carrying capability of the network is very large. All the traffics of other routers is passed through the central router. So that router becomes the busiest one. So congestion level in the middle of network is very high. That can be a biggest disadvantage. So packet drop probability is more in this case. Latency value is almost high.

2.3.1.6 SPIN

It is advanced kind of topology which uses the concept fat tree topology. SPIN represents Scalable Programmable Integrated Network. Here routers are present on the

nodes and resources are present on the leaves. Each node has 4 leaves and parent is replicated 4 times at any level of the topology. For every router the number of parent ports are equal to the number of child ports. The Network size is grown at the rate (N log N)/8 where N is the number of resources. In this network multiple paths exists between the input and output ports. So it acquired the non-blocking capability. Because of the multiple path exists in this network, the network can suffer hardware complexity, more power consumption and more on chip space. The SPIN topology is shown in has 3 levels and black blocks are routers and white blocks are resources.

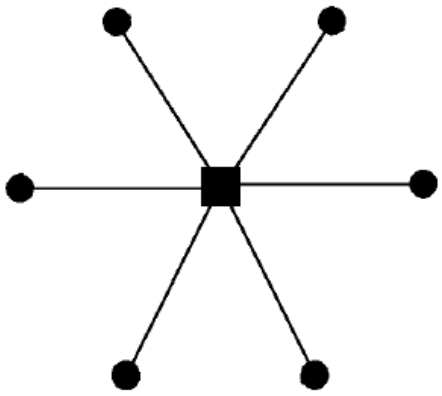


Figure 2.9 Star Topology

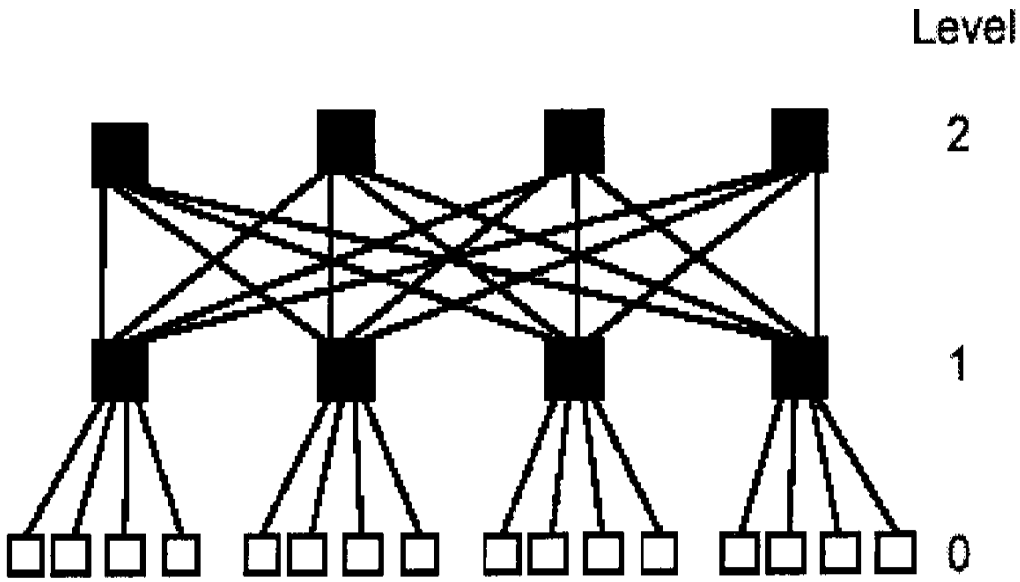


Figure 2.10 SPIN

2.3.2 Switching Techniques

In this sub section the discussion will be on various flow control mechanism. It is also called as switching techniques [13]. This technique operates in message level. The main task of switching technique is to establish connection between input and output channel inside the router. There are two types of popular switching techniques i.e. a) Circuit Switching Technique b) Packet Switching Technique. Packet switching has some classifications i.e. i) S&F (Store and forward) switching ii) VCT (Virtual Cut Through) switching iii) Wormhole switching iv) Virtual channel based switching.

2.3.2.1 Circuit Switching Technique

In this technique an electrical interconnection is established between the source and destination and then the message is transmitted. This technique pre-allocates resources or links across more than one hops to the entire message. Links must be reserved to transmit the entire message. For that a small setup message also called as a probe is sent into the network and reservation for the links needed to transmit the entire message or multiple messages is done from the source to the destination. Once the probe reaches the destination (having successfully allocated the necessary links), the destination sends back an acknowledgement message to the source. As soon as the source receives the acknowledgement message, the message will be released by the source then the message can travel quickly through the network. When the message will complete its traversal, the resources are deallocated. After the setup phase, per-hop latency to acquire resources is avoided. With sufficiently large messages, this latency reduction can amortize the cost of the original setup phase. In addition to possible latency benefits, circuit switching is also buffer less. As links are pre-reserved, buffers are not needed at each hop to hold packets that are waiting for allocation, thus saving on power. While latency can be reduced, circuit switching suffers from poor bandwidth utilization. The links are idle between setup and the actual message transfer and other messages seeking to use those resources are blocked.

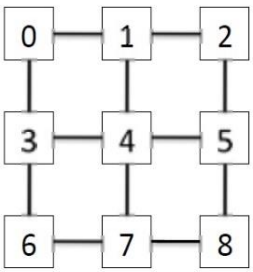


Figure 2.11 Network

Figure 2.11 and Figure 2.12 illustrates an example of how circuit-switching flow control works. Dimension order X-Y routing is assumed with the network shown in Figure 2.11. As time proceeds from left to right (Figure 2.12), the setup flit, S constructs a circuit from Core 0 to Core 8 by traversing the selected route through the network. At time 4, the setup flit has reached the destination and begins sending an acknowledgement flit, a back to Core 0. At time 5, Core 2 wants to initiate a transfer to Core 8; however, the resources (links) necessary to reach Core 8 are already allocated to Core 0. Therefore, Core 2's request is stalled. At time 9, the acknowledgement request is received by Core 0 and the data transfers, D can begin. Once the required data are sent, a tail flit, T is sent by Core 0 to deallocate these resources. At time 19, Core 2 can now begin acquiring the resources recently deallocated by the tail flit. From this example, we see that there is significant wasted link bandwidth, during the setup time and when links have been reserved but there is no data that needs to be transmitted (wasted link bandwidth is shaded in grey). Core 2 also suffers significant latency waiting for resources that are mostly idle.

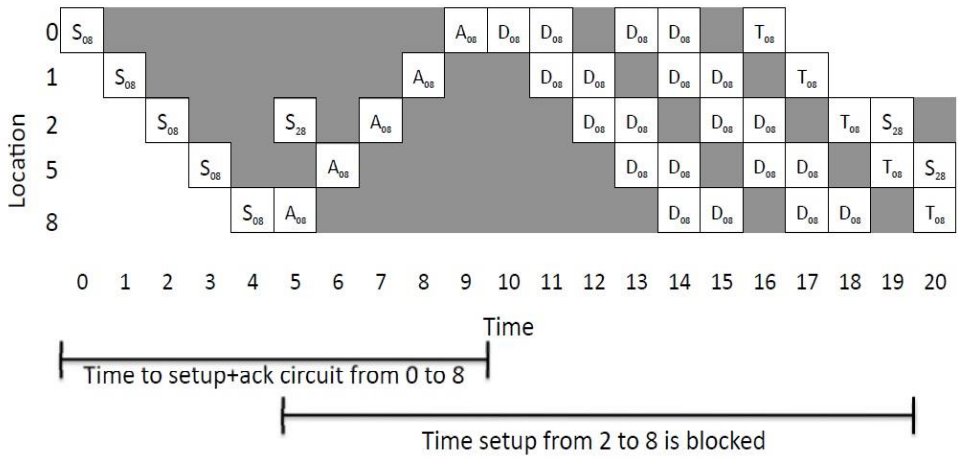


Figure 2.12 Time space diagram for Circuit switched network

2.3.2.2 Packet Switching Technique

Circuit-switching allocates resources to messages and does so across multiple network hops. There are several inefficiencies to this scheme; next, we look at schemes that allocate resources to packets. Packet-based flow control techniques first break down messages into packets, then interleave these packets on the links, thus improving link utilization. Unlike circuit switching, the remaining techniques will require per-node buffering to store in-flight packets.

2.3.2.2.1 Store and Forward Switching Technique

With packet-based techniques, messages are broken down into multiple packets and each packet is handled independently by the network. In store-and-forward (S & F) switching technique each resource or node has to wait until whole data packet has been received by the destination node. Then other packets will be forwarded to the next node. For this reason this switching technique will suffer from long delay problem at each hop. So this kind of technique is not suitable for NoC. Moreover, store and forward flow control requires that there be sufficient buffering at each router to buffer the entire packet. These high buffering requirements reduce store and forward switching's amenability to on-chip networks. In Figure 2.13, we depict a packet traveling from Core 0 to Core 8 using store and forward switching. Once the tail flit has been buffered at each node, the head can then allocate the next link and depart for the next router. Serialization delay is paid for at each hop for the body and tail flits to catch up with the head flit. For a 5-flit packet, the latency is 5 cycles to transmit the packet at each hop.

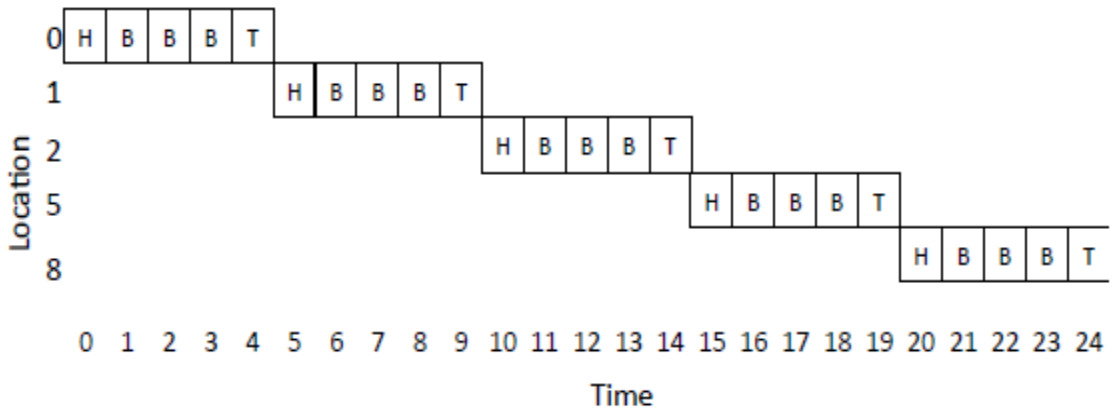


Figure 2.13 Example of store and forward Technique

2.3.2.2.2 Virtual Cut Through Switching Technique

Store and forward technique experience more delay at each hop. So to reduce the delay Virtual cut through (VCT) switching technique came into existence. Here the transmission of a packet is allowed to proceed to the next node before the entire data packet is received by the destination router. So this technique got the advantage of reduced experienced packet latency. It can replace the store and forward switching technique as shown in Figure 2.13. In Figure 2.14, 25 cycles are required to transmit the entire packet; with virtual cut-through, this delay is reduced to 9 cycles. Here storage and bandwidth are allocated in the unit size equivalent to size of the packet. If the neighboring downstream

router has enough storage space to store entire packet then only the packet from the current router will move forward. When packet size is large VCT switching technique needs a large buffer. So a Network on Chip having constraints like area and power may find it very tough to adapt this technique. In Figure 2.15 VCT with delaythe entire packet is delayed when traveling from node 2 to node 5 even though node 5 has buffers available for 2 out of 5 flits. No flits can proceed until all 5 flit buffers are available.

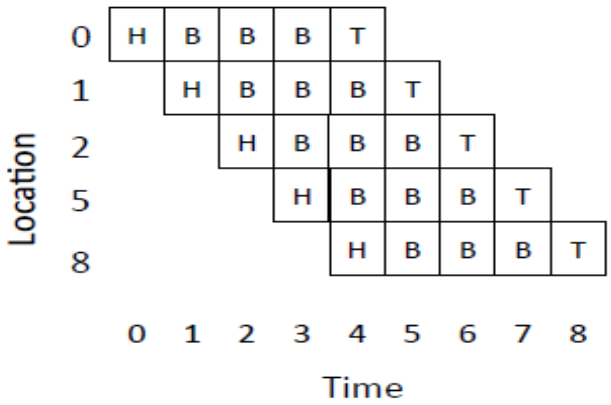


Figure 2.14 VCT with no delay

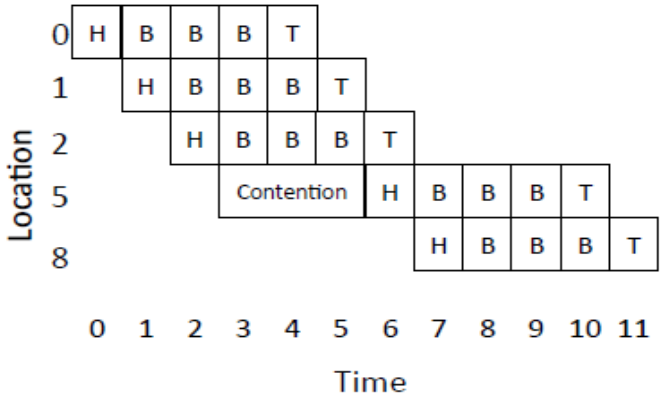


Figure 2.15 VCT with delay

2.3.2.2.3 Wormhole Switching Technique

Wormhole switching technique [13] divides the packets into flits, allowing flits to move on to the neighboring router before the entire packet is received at the destination like virtual cut-through switching technique. In wormhole switching the flit can move forward to the current node if there is sufficient space in the buffer for the flit. Here storage and bandwidth is allocated to the flits not to the entire packet like S&F switching and VCT switching. So it enables the routers to use small sized buffer and some on-chip area is being saved. In wormhole flow control there is ineffective use of link bandwidth as it uses only the buffers effectively. This technique allocates storage space of a flit size, a link has to be

reserved for the duration of lifetime of packet in the router. All the physical links will be left idle if a reserved packet is blocked. Wormhole switching technique will acquire more routers if the packet size is large because that packet has large number of flits. As a result the physical links will remain idle. Throughput will degrade due to the blocked messages. In a example suppose each router has 2 flit buffers. When the head flit experiences contention traveling from 1 to 2, the remaining two body and tail flits are stuck at Core 0 as no buffer space is available at Core 1 until the head moves to Core 2.However, the channel is still held by the packet even though it is idle as shown in grey. Due to wormhole switching packet latency is reduced by transmitting a flit from the router as soon as availability of downstream buffer occurs (in the absence of contention, the latency is the same as virtual cut through). Additionally, wormhole flow control requires much smaller buffers than packet-based techniques. Due to constraints like area and power of Network on Chip, wormhole switching technique is the suitable and preferable technique adopted thus far.

Table 2.1 Summary of various switching techniques

Switching Techniques	Links	Buffers	Comments
Circuit Switching	Messages	Buffer less	Requires setup & acknowledgment
Store and Forward	Packet	Packet	Head flit must wait for entire packet before proceeding on next link
Virtual Cut-Through	Packet	Packet	Head can begin next link traversal before tail arrives at current node
Wormhole	Packet	Flit	Head of line blocking reduces efficiency of link bandwidth

2.3.3 Routing Algorithm

First we need to determine the topology for the on-chip network and then a specific routing algorithm should be chosen. A routing algorithm determines the entire path for the message or data packets to reach the destination. Main task of a routing algorithm is to distribute the traffic from different nodes evenly throughout the network. Like that it will avoid hotspots and improve the network latency and throughput value by minimizing

contention. A particular routing algorithm affects the router design complexity, area and hence affects the power consumption in whole network to achieve all performance requirements. There are various classification of routing algorithm. The detailed classification is described in chapter 3. Generally routing algorithms are classified into three types. Those are i) Deterministic routing ii) Oblivious Routing iii) Adaptive routing. Researchers proposed various routing algorithms. But there is a Dimension order routing (DOR) which is very simple to implement. So it is preferable for the networks. DOR is an example of deterministic routing. XY routing comes under this DOR which is described in chapter 4. Deterministic routing is a subset of oblivious routing. In deterministic routing one path is calculated between source and destination and routing in that path is done throughout the process. In oblivious routing is done in different paths but it doesn't bother about the congestion in network. In adaptive routing more than one path is calculated between source and destination but only path is selected according to the congestion in the network. There is another classification of routing algorithms i.e. Minimal and Non-minimal routing. In minimal routing a path having smallest number of hops is selected. Non-minimal routing is just the opposite. Without network congestion non-minimal routing will increase the latency value. But with congestion it's performance is acceptable. A minimal routing which can avoid congested links will give satisfactory result by reducing the network latency. In Figure 2.16 Example of DOR, Oblivious and Adaptive routingan example of DOR, oblivious and adaptive routing is shown. All the nodes are identified by (x,y) coordinates. Here (0,0) is chosen as the source and (2,3) is chosen as the destination. In DOR routing is done in that one path throughout the network. In obvious routing two paths are calculated for routing. In adaptive routing algorithm the routing is done with that path avoiding congested links.

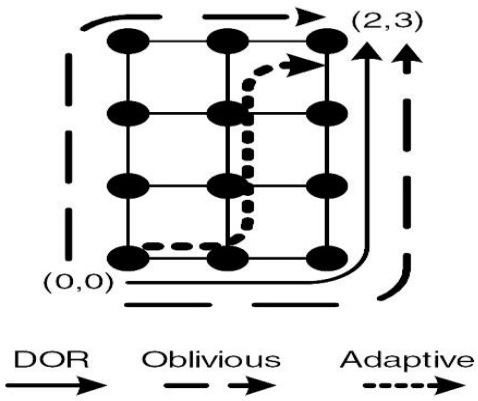


Figure 2.16 Example of DOR, Oblivious and Adaptive routing

Chapter 3
Routing Algorithm for NoC

3.1 Taxonomy of routing algorithms

Routing in NOC determines the path that each packet follows between source and destination pair. There are some properties of routing algorithms which are essentially required for interconnection networks i.e. connectivity, adaptivity, deadlock and livelock freedom, fault tolerance [14]. Connectivity is the ability to route packets from any source node to any destination node. Adaptivity is the ability to route packets through alternative paths in the presence of contention or faulty components. Deadlock freedom is the ability to guarantee that packets will not block or wander across the network forever. Fault tolerance is the ability to route packets in the presence of faulty components. There are many ways to classify routing in on-chip-networks. According to number of destinations routing algorithms are classified into two types i.e. unicast and multicast routing [15]. In case of unicast routing the travelling packets have only one destination, but in case of multicast routing, the packets have more than one destination. Between the two routing available unicast routing policies are considered as a good methodology for Network on Chip as it has point-to-point communication links among various nodes on a system on chip. According to routing assessment capability unicast routing has four classifications. They are source routing, distributed routing, centralized routing and multiphase routing. According to source routing, routing path is determined by some routing decisions when data is generated by the node and stored in the packet header. But according to distributed routing, routing path is decided as the packets or flits flow in the network. Multiphase routing is formed by combining the source and destination routing scheme. In centralized routing, data flow in a system is controlled by a centralized controller. In implementation point of view routing algorithms are of two types i.e. lookup table based and FSM based. According to Lookup table routing algorithms a lookup table is present in each router and they are implemented in software level. Hence they are popular. The entries of the lookup table is kept on updating by the routing algorithm If these entries will change it can change the whole routing algorithm. Finite State Machine based routing algorithms can be implemented in both ways i.e. software and hardware.

In adaptivity point of view routing algorithms are also classified into two categories. In deterministic routing one path is calculated between source and destination and through that path only routing is done inside a network. In adaptive routing algorithms multiple paths are calculated between source and destination but routing is done in one selected path which is less congested. Disadvantages of this type algorithm is its implementation

complexity, cost and more power consumption. So we have to consider the right QoS metrics before implementation of this.

There are another kind of routing algorithm i.e. backtracking routing algorithm which is fault-tolerance in nature. Another algorithm exists i.e. progressive routing where a channel is held in reserve for flits to be moved forward. Some routing algorithms send packets or flits only in that direction which is nearer to the destination and these routing algorithms are known as profitable algorithms. A misrouting algorithm may forward a packet or flit away from the destination as well and can cause packet loss. According to the number of available routing paths, routing algorithms can also be classified i.e. complete and partial routing algorithms.

Various application specific routing algorithms have been proposed for the NOC [3]. Many researchers suggested static routing algorithms and performed communication analysis based on the static behavior of NOC processes, thus, determining the static routing for NOC. Siebenborn et al. and Hu et al. [16] used a Communication Dependency Graph to evaluate inter-process communications.

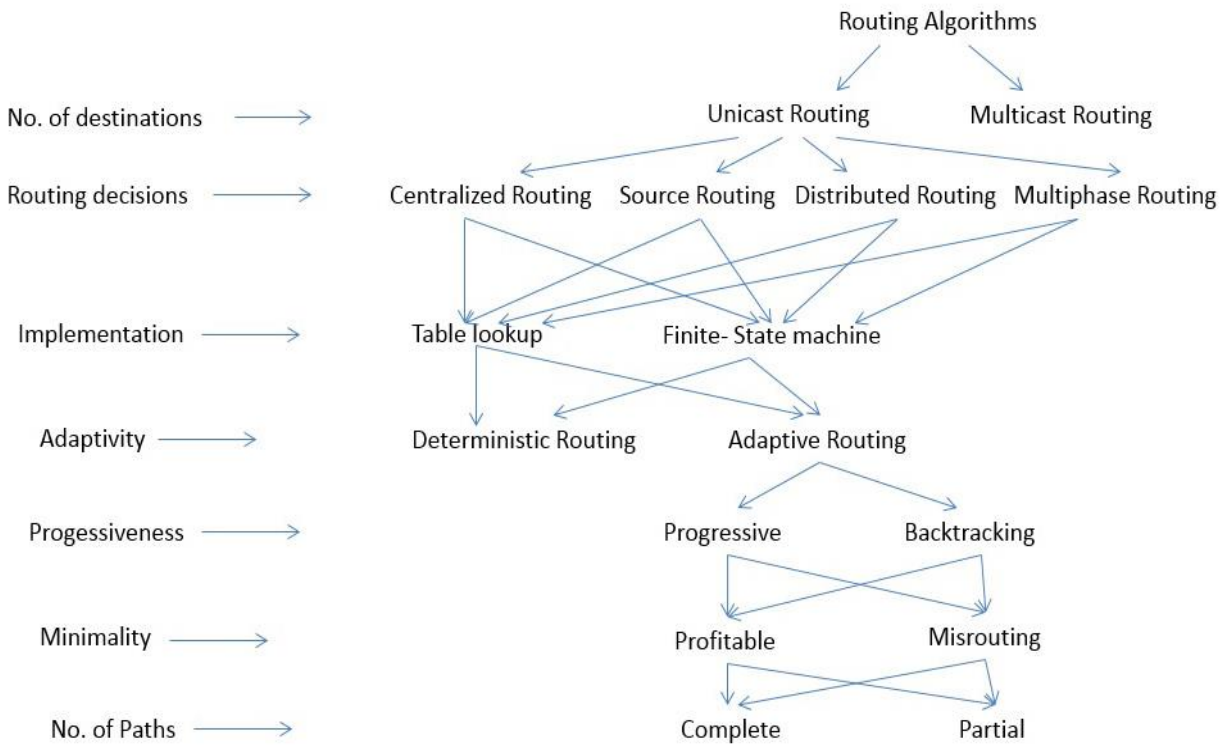


Figure 3.1 Taxonomy of Routing algorithms

3.2 Classification of Routing algorithm based on adaptivity

3.2.1 Deterministic Dimension-Ordered Routing

A routing algorithm can be described by which turns are permitted. Figure 3.2 illustrates all possible turns in a 2D mesh network while Figure 3.3 illustrates the more limited set of permissible turns allowed by DOR X-Y routing. Allowing all turns result in cyclic resource dependencies, which can lead to network deadlock. To prevent these cyclic dependencies, turns may be disallowed. As you can see, no cycle is present in Figure 3.3. Specifically, a message traveling east or west is allowed to turn north or south; however, messages traveling north and south are permitted no turns. Two of the four turns will not be permitted, so a cycle is not possible. Alternatively, Y-X routing can be used where messages traveling north or south are allowed to turn east or west but once a message is traveling East or West, no further turns are permitted. Depending on the network dimensions, i.e. whether there are more nodes along X or Y, one of these routing algorithms will balance load better with uniform random traffic since channel load is higher along the dimension with fewer nodes. Dimension order routing is both simple and deadlock-free; however, it eliminates path diversity in a mesh network and thus lowers throughput. With dimension order routing, exactly one path exists between every source and destination pair. Path diversity, the routing algorithm is unable to route around faults in the network or avoid areas of congestion. As a result of routing restrictions, dimension order routing does a poor job of load balancing the network [17].

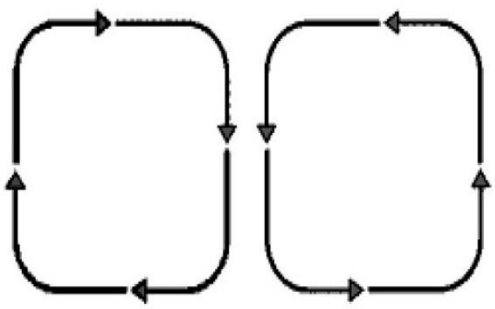


Figure 3.2 All possible turns

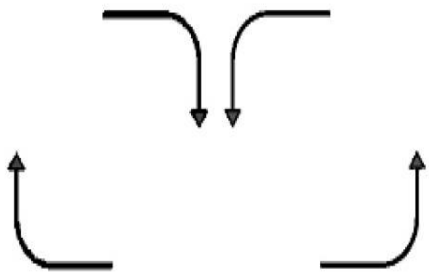


Figure 3.3 X-Y turns

3.2.2 Oblivious Routing

Using an oblivious routing algorithm [18], routing paths are chosen without regard to the state of the network. By not using information about the state of the network, these routing algorithms can be kept simple. Valiant's randomized routing algorithm is one example of an oblivious routing algorithm. To route a packet from source s to destination d using Valiant's algorithm, an intermediate destination d' is randomly selected. The packet is first routed from s to d' and then from d' to d. By routing first to a randomly selected intermediate destination before routing to the final destination, Valiant's algorithm is able to load balance traffic across the network; the randomization causes any traffic pattern to appear to be uniform random. Load balancing with Valiant's algorithm comes at the expense of locality; for example, by routing to an intermediate destination, the locality of near neighbor traffic on a mesh is destroyed. Hop count is increased, which in turn increases the average packet latency and the average energy consumed by the packet in the network. Valiant's routing algorithm can be restricted to support only minimal routes [158], by restricting routing choices to only the shortest paths in order to preserve locality. In a k-ary n-cube topology, the intermediate node 'd' must lie within the minimal quadrant; the smallest n-dimensional sub-network with s and d as corner nodes bounding this quadrant. With both Valiant's randomized routing and minimal adaptive routing, dimension order routing can be used to route from s to d' and from d' to d. If DOR is used, not all paths will be exploited but better load balancing is achieved than deterministic routing from s directly to d. Figure 3.4 and Figure 3.5 illustrates a routing path selected using Valiant's algorithm and minimal oblivious routing. In Figure 3.4, Valiant's algorithm randomly selects an intermediate destination d'. The random selection can destroy locality and significantly increase hop count; here, the hop count is increased from three hops to nine hops. To preserve locality, minimal oblivious routing can be employed as in Figure 3.5. Now, d' can only be selected to lie within the minimal quadrant formed by s and d, preserving the minimum hop count of three. One possible selection is highlighted (two other paths are possible for this source-destination pair as shown with dashed lines). Valiant's routing algorithm and minimal oblivious routing are deadlock free when used in conjunction with X-Y routing. An example of an oblivious routing algorithm that is not deadlock free is one that randomly chooses between X-Y or Y-X routes. The oblivious algorithm that randomly chooses between X-Y or Y-X routes is not deadlock-free because all four turns from Figure 4.2 are possible leading to potential cycles in the link acquisition graph.

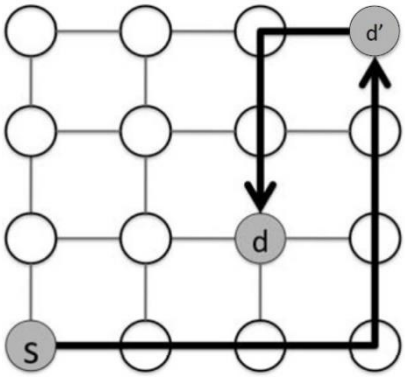


Figure 3.4 Valiant's routing algorithm

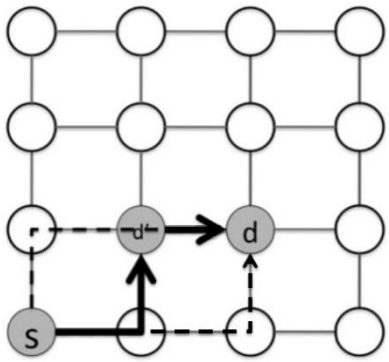


Figure 3.5 Minimal oblivious routing

3.2.3 Adaptive Routing

A more sophisticated routing algorithm can be adaptive, i.e. the path a message takes from A to B depends on the network traffic situation. For instance, a message can be going along the X-Y route, see congestion at (1,0)'s east outgoing link and instead choose to take the north outgoing link towards the destination. Local or global information can be leveraged to make adaptive routing decisions [19]. Adaptive routing algorithms often rely on local router information such as queue occupancy and queuing delay to gauge congestion and select links. The backpressure mechanisms used by flow control (discussed in the next chapter) allow congestion information to propagate from the congestion site back through the network. Figure 3.6 shows all possible (minimal) routes that a message can take from Node (0,0) to Node (2,3).There are nine possible paths. An adaptive routing algorithm that leverages only minimal paths could exploit a large degree of path diversity to provide load balancing and fault tolerance. Adaptive routing can be restricted to taking minimal routes between the source and the destination. An alternative option is to employ misrouting, which allows a packet to be routed in a misrouting non-productive direction resulting in non-minimal paths. When misrouting is permitted, livelock becomes a concern. Without mechanisms to guarantee forward progress, livelock can occur as a livelock packet is continuously misrouted so as to never reach its destination. We can combat this problem by allowing a maximum number of misroutes per packet and giving higher priority to packets than have been misrouted a large number of times. Misrouting increases the hop count but may reduce end-to-end packet latency by avoiding congestion (queuing delay). With a fully-adaptive routing algorithm, deadlock can become a problem. For example, the adaptive route shown in Figure 3.6 is a superset of oblivious routing and is subject to potential deadlock. Planar-adaptive routing limits the resources needed to handle deadlock

by restricting adaptivity to only two dimensions at a time. Duato has proposed flow control techniques that allow full routing adaptivity while ensuring freedom from deadlock. Deadlock-free flow control will be discussed in Chapter 5. Another challenge with adaptive routing is preserving inter-message ordering as may be needed by the coherence protocol. If messages must arrive at the destination in the same order that the source issued them, adaptive routing can be problematic. Mechanisms to re-order messages at the destination can be employed or messages of a given class can be restricted in their routing to prevent re-ordering.

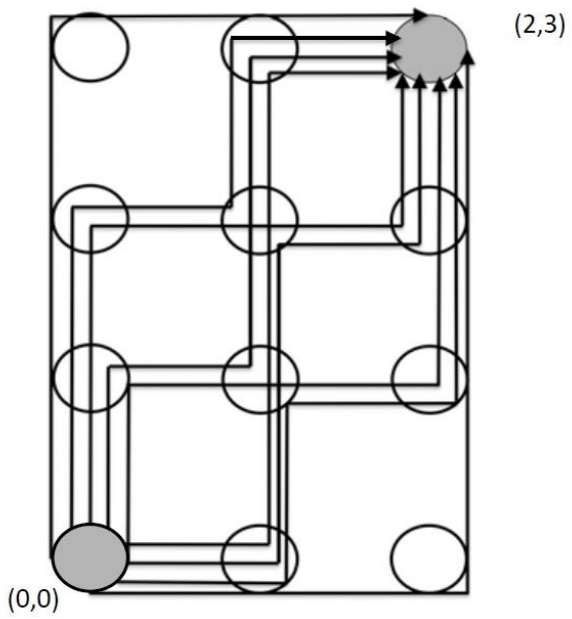


Figure 3.6 Adaptive routing example

3.2.3.1 Adaptive Turn Model Routing

While we introduced turn model routing earlier chapter, discussing how dimension order X-Y routing eliminates two out of four turns (Figure 3.3), here, we explain how turn model can be more turn model routing broadly applied to derive deadlock-free adaptive routing algorithms. Adaptive turn model routing eliminates the minimum set of turns needed to achieve deadlock freedom while retaining some path diversity and potential for adaptivity. With dimension order routing only four possible turns are permitted of the eight turns available in a two dimensional mesh. Turn model routing increases the flexibility of the algorithm by allowing six out of eight turns. Only one turn from each cycle is

eliminated. In Figure 3.7, three possible routing algorithms are illustrated. Starting with all possible turns (shown in Figure 3.2), the north to west turn is eliminated; after this elimination is made, the three routing algorithms shown in Figure 3.7 can be derived. In Figure 3.7a, the west-first algorithm is shown; in addition to eliminating the North to West turn, the South to West turn is eliminated. In other words, a message must first travel in the West direction before traveling in any other direction. The North-Last algorithm (Figure 3.7b) eliminates both the North to West and the North to East turns. Once a message has turned North, no further turns are permitted; hence, the North turn must be made last. Finally, Figure 3.7c removes turns from North to West and East to South to create the Negative-First algorithm. A message travels in the negative directions (west and south) first before it is permitted to travel in positive directions (east and north). All three of these turn model routing algorithms are deadlock-free. Figure 3.9 illustrates a possible turn elimination that is invalid; the elimination of North to West combined with the elimination of West to North can lead to deadlock. A deadlock cycle is depicted in Figure 3.9 that can result from a set of messages using the turns specified in Figure 3.9 Illegal turn model routing and resulting deadlock cycle. Odd-even turn model routing proposes eliminating a set of two turns depending on whether the current node is in an odd or even column. For example, when a packet is traversing a node in an even column1, turns from East to North and from North to West are prohibited. For packets traversing an odd column node, turns from East to South and from South to West are prohibited. With this set of restrictions, the odd-even turn model is deadlock free provided 180◦ turns are disallowed. The odd-even turn model provides better adaptivity than other turn model algorithms such as West-First. With West-First, destinations to the West of the source, have no flexibility; with odd-even routing, there is flexibility depending on the allowable turns for a given column. In Figure 3.8 Negative first routing example, we apply the Negative-First turn model routing to two different source destination pairs. In Figure 3.8a, three possible routes are shown between (0,0) and (2,3) (more are possible); turns from North to East and from East to North are permitted allowing for significant flexibility. However, in Figure 3.8b, there is only one path allowed by the algorithm to route from (0,3) to (2,0).The routing algorithm does not allow the message to turn from East to South. Negative routes must be completed first, resulting in no path diversity for this source-destination pair. As illustrated by this example, turn model routing provide more flexibility and adaptivity than dimension-order routing but it is still somewhat restrictive.

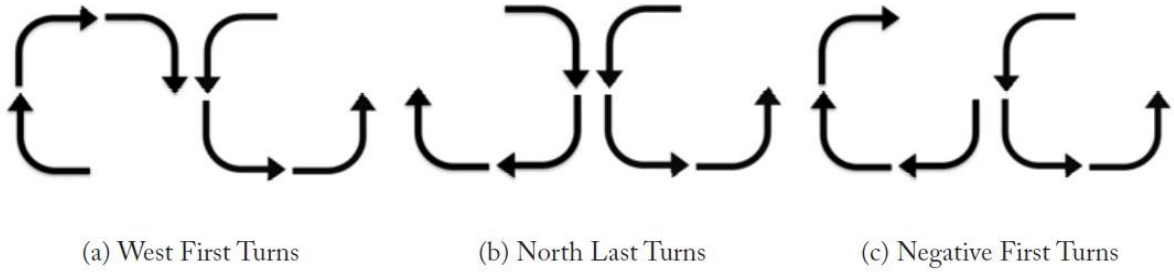


Figure 3.7 Turn model routing

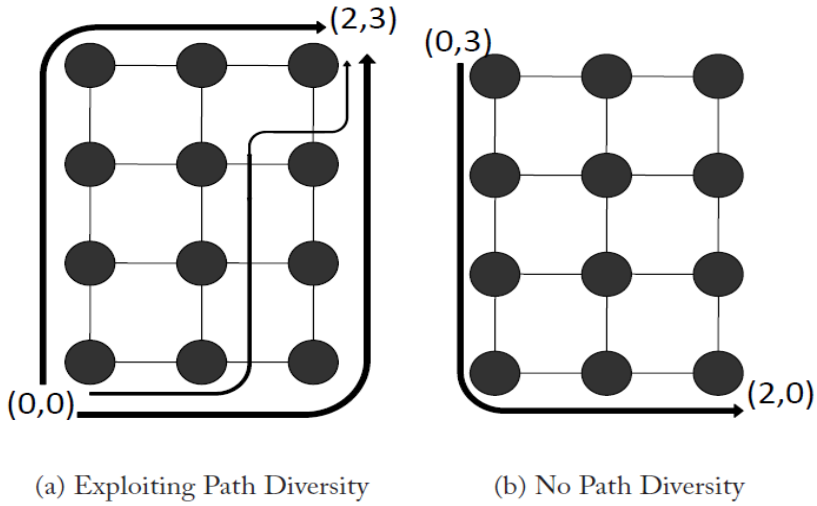


Figure 3.8 Negative first routing example

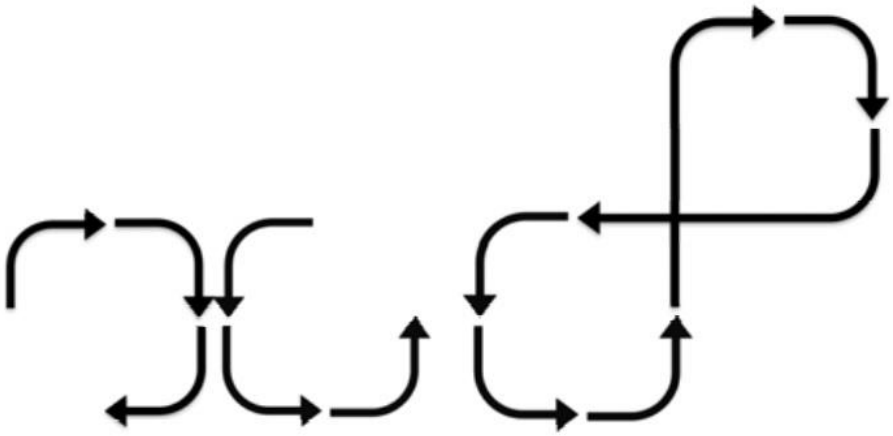


Figure 3.9 Illegal turn model routing and resulting deadlock cycle

3.3 Problems in routing

There are different kinds of problems arise during the routing process. Especially in oblivious routing this type of problem arises which results in blockage of traffic. So routing devices has to wait for the reduction of traffic and then try sending repeatedly. Deadlock, livelock and starvation are potential problems on both oblivious and adaptive routing. These problems are discussed as follows in the next subsection.

3.3.1 Deadlock

When the data packets are moving around the network, they usually reserve some resource in between the path. When all the packets are waiting for each other to release the resources in a cyclic manner then this kind of situation is called as deadlock. As a result the all packets will be blocked inside the deadlock condition and they can't be routed to their destination and that is a huge loss in the on chip networks.

3.3.2 Livelock

This kind of problem happens when the packets are moving around a destination without reaching there. So data can't be routed to the destination. This kind problem will happen in non-minimal routing algorithm where the routing algorithm choose the longest path whether the shortest path exist or not by observing the network congestion. For throughput improvement this kind of problem should be avoided.

3.3.3 Starvation

This kind of problem is very similar to the real life situations. Different priority assignment to the data packets in NoC can cause this kind of problem. So high priority packets can reach their destination easily but the low priority elements will never reach their destination. This kind of situation arises because the high priority packets reserve the resources and the low priority packets get starved for the resources. This kind of problem can be avoided by using an appropriate routing algorithm which has some bandwidth reservation for the low priority packets.

3.4 Performance parameters for Routing algorithm

Performance requirements that every NoC must satisfy

- Small latency
- Guaranteed throughput
- Path diversity
- Sufficient transfer capacity

- Low power consumption
- Fault and distraction tolerance
- Architectural requirements of scalability and programmability

But there are three important parameters for NoC which we are considering in this thesis i.e. Network Latency, Network Throughput and Total Network Power, Packet drop.

Network Latency is measured from the time its head flit is generated by the source to the time its tail flit is consumed by the destination. Let L_{ij} be the packet j and N_i be the number of packet received by processor i (After warm-up time). N is the number of processors in the platform.

Average Network Latency = $L_{avg} = \frac{1}{N}\sum_{i=1....N}\left(\frac{1}{N_i}\sum_{\forall j} L_{ij}\right)$ for i = 1, 2… N and for all j

Network Throughput is defined as the rate at which the network can successfully accept and deliver the injected packets. Let Tsim and Twarm be the simulation time and warm-up time respectively.

Average Network Throughput (in packets per unit time per node) is given by

$$T_{avg} = \frac{1}{N(T_{sim} - T_{warm})}\sum_{i=1.....N} N_i \qquad where\ i = 1,2 \dots \dots . N$$

Total Network power estimation is based on activities of components while running a certain traffic pattern. It should be less for all NoC.

3.5 Simulator Used: NIRGAM

NIRGAM is an extensible and modular SystemC based simulator (NIRGAM), which let the user plug-in and experiment with different applications and routing algorithms [18]. It allows the user to analyze the performance (Average latency, throughput and total network power) of a NoC design for a user specified application and a user specified routing algorithm. At present, NIRGAM (NoC Interconnect RoutinG and Applications' Modeling) simulator supports mesh, torus, mesh with link failures and irregular topologies with wormhole switching mechanism. Quality of Service (QoS) is supported where the user can reserve a specified amount of band-width for Guaranteed Throughput (GT) traffic. ORION power model has been integrated to calculate router power consumption for any specified number of clock cycles. It allows the users to develop their own applications or they can

use the available applications which is attached to the IP cores in the defined topologies. The available applications are: source (sender) application, sink (receiver) application and synthetic traffic generators (constant bit rate (CBR), bursty, input trace based traffic). Supported traffic patterns include: butterfly, bit-reversal, bit-shuffle and bit-transpose. This simulator facilitate the users to include their own applications and attach them to the defined IP cores. NIRGAM is aimed primarily at the NoC research community wherein it provides researchers with convenient and efficient mechanism to experiment with NoC design in terms of routing algorithms and applications on various topologies. Users can easily plug-in their own routers and applications. The simulator is capable of dynamically loading a router and attaching any user-specified application library to any core. A number of demonstrations showing the various capabilities of NIRGAM will be available during the poster session.

1. Sending CBR traffic from a core to a specified or randomly chosen destination core.
2. Sending Bursty traffic from a core to a specified or randomly chosed destination core.
3. Sending Bursty and CBR traffic after reserving a chosen band-width for Bursty traffic.
4. Sending Bursty and CBR traffic in a mesh topology with link failures.
5. Router power estimation for any specified number of clock cycles.

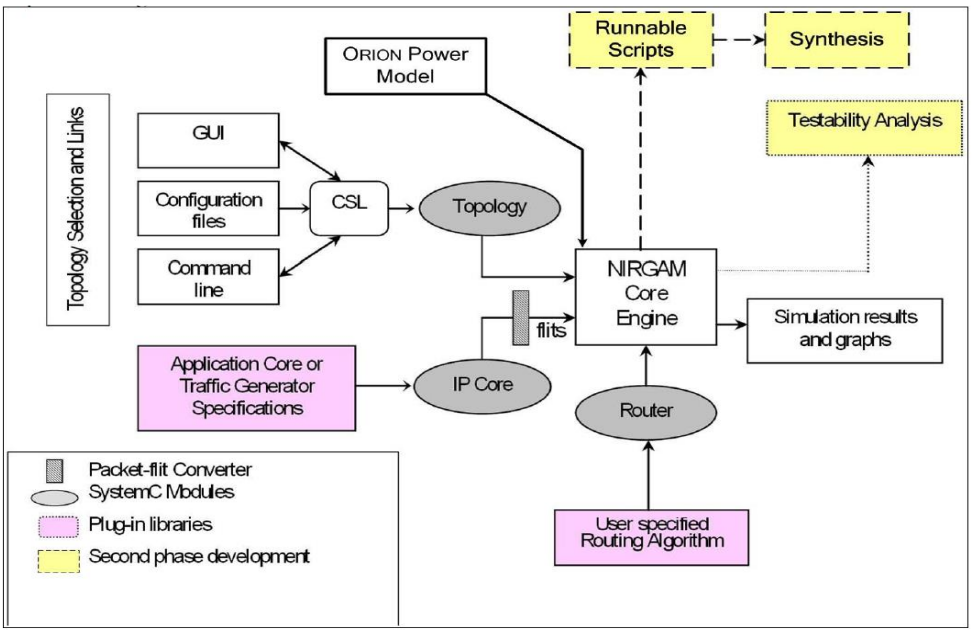


Figure 3.10 NIRGAM Simulator

Chapter 4

Performance Evaluation of a Power Efficient Routing Algorithm with Improved Latency and Throughput

4.1 XY Routing Algorithm

As discussed in previous chapter a deterministic routing algorithm uses a fixed routing path throughout the process. XY routing algorithm comes under deterministic routing algorithm. This algorithm can be implemented for both for regular and irregular network topology. It is called dimension order routing (DOR). It follows the concept of minimal turning routing. In this routing each node or router of NoC is identified by the (x, y) co-ordinates of that node for a 2D mesh. According to this algorithm the data packets will traverse in X-direction towards the destination column. After finding the destination column the data packets will traverse to the destination node. Data movement in this algorithm is described in an example in Figure 4.2. This algorithm simply states that "First the data will move in X-direction and then in Y-direction". That is why the name of the algorithm is XY Routing algorithm. According to this algorithm the packets can't move first in Y-direction then in X-direction. So it has some routing or turning restrictions. Due to which it becomes deadlock free.

According to this algorithm, (x ,y) coordinate of Current router is compared to the (x,y) coordinate of Destination router [19]. If they are equal, this will indicate data packets have reached its destination and they will be routed to the resource through the local port. If they are not equal, the x-coordinate of destination is initially compared to the x-coordinate of current router. When x-coordinate of current router is less than x-coordinate of destination router, the data packets will be transmitted to the East direction, when x-coordinate of current router is greater than x-coordinate of destination router, then packets will be routed to West direction. Then check if the data packet is already on the destination column or not. If this last condition is true, the vertical address of destination is compared to the vertical address of current router. Data packets will be traversed to South direction when vertical address of current router is less than vertical address of destination router, to North when vertical address of current router is greater than vertical address of destination router.

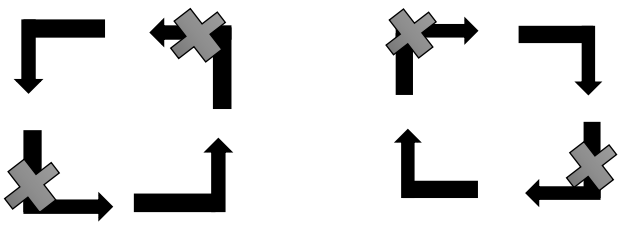


Figure 4.1 Allowed turns in XY routing

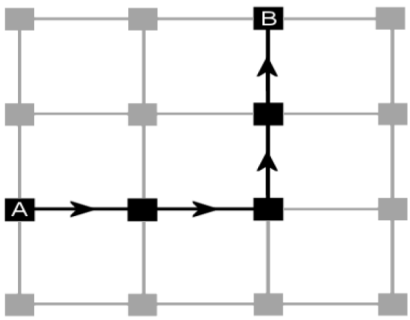


Figure 4.2 Path shown for XY routing
for Source A to Destination B

Some common term are

Source router: (S_x, S_y)

Destination router: (D_x, D_y)

Current router: (C_x , C_y)

$X_{offset} = D_x - C_x$

$Yoffset = D_y - C_y$

Algorithm:

begin

if ($X_{offset} > 0$) then

Select EAST port;

else

if ($X_{offset} <0$) then

Select WEST port ;

else

if ($X_{offset} = 0$) then

{

if ($Y_{offset} < 0$)

Select SOUTH port ;

else

if ($Y_{offset} > 0$)

Select NORTH port;

else

if ($Y_{offset} = 0$)

Select LOCAL port;

}

End

4.2 OE Routing Algorithm

As discussed before in adaptive routing algorithm more than one routing path is calculated between source and destination node and one path will be selected for routing according network congestion conditions. Those paths are kept on changing according to the dynamic change of the network congestion condition. OE routing algorithm is also called as Odd-Even Routing algorithm. It comes under class of distributed adaptive routing algorithm. It is based on odd-even turn model. It introduces various routing restrictions or turning restriction. As a result it is deadlock free. Odd-even turn model doesn't require any virtual channel in 2D mesh topology [20]. In a 2-D mesh each node is recognized by its (x, y) coordinate. According to the turn model, if its x dimension element is even then that column is considered as even column and if column's x co-ordinate is an odd number then that column is considered as odd column. A 90- degree change of traveling direction is called as a turn. As per travelling direction there are eight kind of turns available. ES turn can be described as turn if it has 90 degree variation in direction from East to South. WS turn can be described as turn if it has 90 degree variation in direction from West to South. Like that we can describe other six kinds of turns named as EN, WN, SE, SW, NE, and NW turns, where N, S, W and E specify North, south, West and East respectively.

According to OE routing algorithm the following theorems are necessary to avoid deadlocks.

- Theorem1: If a node is present on an even column the packets can't take EN turns shown in Figure 4.3 and if a node is present on an odd column the packets can't take NW turns shown in Figure 4.4 .
- Theorem 2: If a node is present on an even column the packets can't take ES turns shown in Figure 4.3 and if a node is present on an odd column the packets can't take SW turns shown in Figure 4.4.

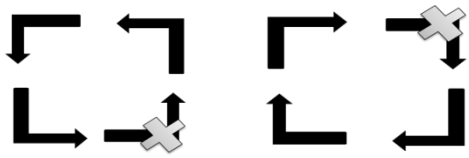


Figure 4.3 Allowed turns for even column in OE routing

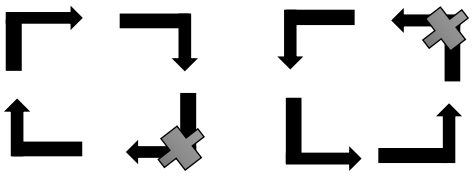


Figure 4.4 Allowed turns for odd column in OE routing

Algorithm:

/*Source router: (Sx,Sy);destination router: (Dx,Dy); current router: (Cx,Cy); Ex= Dx-Cx ; Ey=Dy-Cy ; avail_dim_set is an array consisting of all possible routing directions */

Begin

1. avail_dim_set is empty.
2. If (E_x= 0 and E_y= 0), Select LOCAL port for routing.
3. If (E_x= 0), it indicates packets are in destination column
4. If above condition is satisfied then check for E_y and compute either SOUTH or NORTH to be stored in avail_dim_set.
5. If (E_x >0) Check for E_y. If E_y = 0 then add EAST in avail_dim_set. Else check for C_x is odd or C_x = D_x

6. If the above condition is satisfied then check E_y accordingly add SOUTH or NORTH in avail_dim_set.

7. If D_x is odd or E_x != 1, then add EAST to avail_dim_set.

8. If (E_x <=0) then add WEST to avail_dim_set and check for C_x. If C_x is even, then check for E_y

9. If E_y <0, add SOUTH to avail_dim_set or add NORTH to avail_dim_set.

End

4.3 DyAD Routing Algorithm

This is one kind of smart or intelligent routing algorithm [21]. This routing algorithm is dynamic in nature. It consists of the combined advantages of both deterministic routing algorithm and adaptive routing algorithm. As discussed in earlier chapters deterministic routing algorithm has the advantage of having less latency in less congested network and simple router design. The adaptive routing algorithms has the advantages of avoiding congested links in the network and having high throughput. At low levels of congestion in a network adaptive routing algorithm's performance is not satisfactory. It will suffer from high latency overhead due to this extra logic activities. DyAD routing algorithm switches from Adaptive to Deterministic based on current network congestion. In DyAD routing each and every router in NoC always monitors the network condition. Congestion values are calculated for all neighbouring routers and decisions are made according to this real time information. DyAD router functions as a deterministic router when network congestion is having less value. So it will get the advantage of less latency. The DyAD router operates in adaptive routing mode when the network congestion level is more. So it is getting the advantage of avoiding the congested links by exploring other routing paths, this leads to higher network throughput which is highly desirable for applications implemented using the NoC approach. The freedom from deadlock and livelock can be guaranteed when mixing deterministic and adaptive routing modes into the same NoC.

Algorithm:

Begin

If (CONGESTION = 0) then

Deterministic routing algorithm (XY Routing)

Else

Adaptive routing algorithm (OE Routing)

End

4.4 Experimental Setup in Simulator

The experimental setups for the evaluation of the routing algorithms are 3x3 mesh topology where each node is connected with CBR (Constant Bit Rate) traffic generator of value 12 Gbps. Buffer depth (number of buffers) of input channel FIFO is 32. Number of virtual channels per physical channel is 4. Link length is 3 um. Packet size is taken as 20 bytes with flit interval 2 clock cycles and clock frequency 1 GHz.

4.5 Simulation Results

The simulation results are evaluated in the given experimental configuration described in previous section for average latency, average throughput and total network power with % of variation of load in uniform random traffic pattern.

Table 4.1 % of Load vs Average Latency

% of Load	Overall Average Latency per channel (clk cycle/packet)		
	XY	*OE*	*DYAD*
10	11.251	10.5371	9.5763
30	11.4877	10.7535	9.8734
50	13.5859	12.3141	11.9326
70	13.1227	13.3141	12.9857
90	14.7451	12.4611	11.9523
100	29.4295	26.1582	24.8962

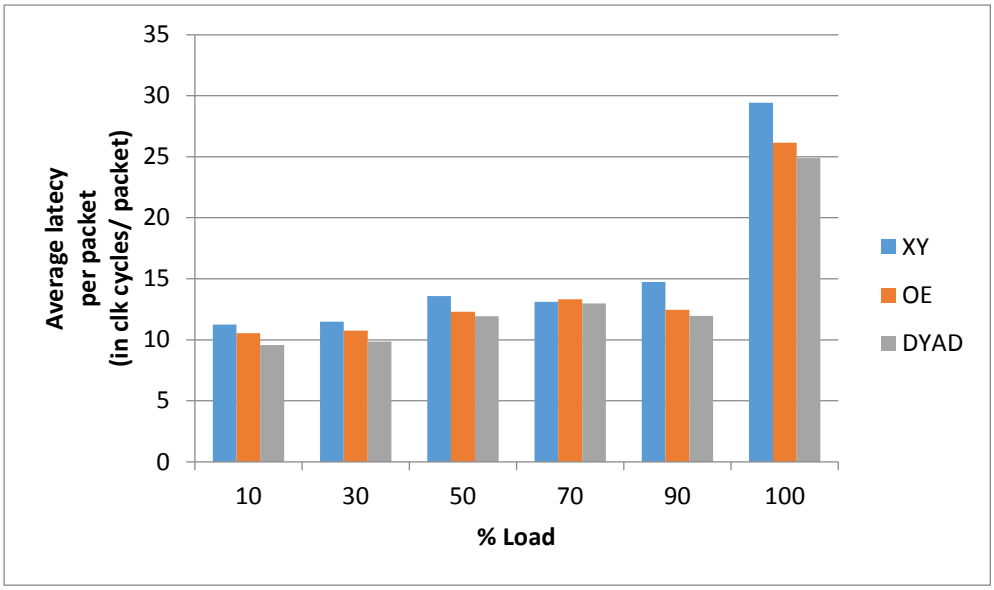


Figure 4.5 % of Load vs Average latency per packet

Table 4.2 % of Load vs Average Throughput

% of Load	Average Throughput (Gbps)		
	XY	*OE*	*DyAD*
10	1.95374	2.28772	3.3256
30	3.58726	5.12377	7.5328
50	8.198	9.2098	10.5469
70	9.38296	10.4799	11.7646
90	9.96787	10.9844	12.772
100	10.2839	11.9231	13.3765

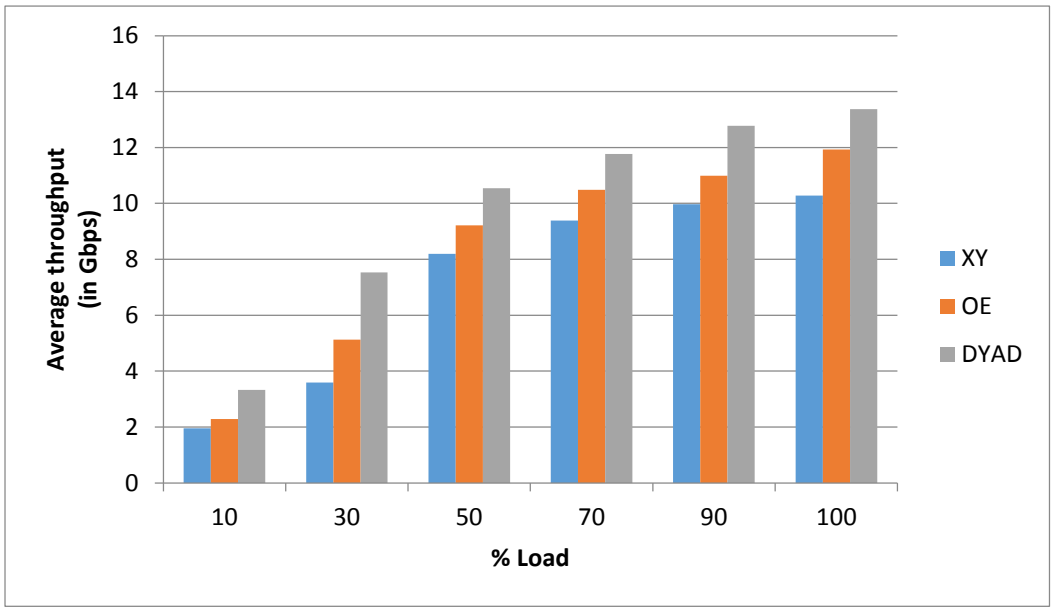


Figure 4.6 % of Load vs Average Throughput

Table 4.3 % of Load vs Total Network Power

% of Load	Total Network Power (in mW)		
	XY	*OE*	*DyAD*
10	1.95374	2.28772	1.1156
30	4.58726	5.13877	2.5428
50	9.45833	10.4572	5.369
70	9.38296	10.4799	5.6246
90	9.56787	10.4404	5.7352
100	12.2839	13.5231	9.3885

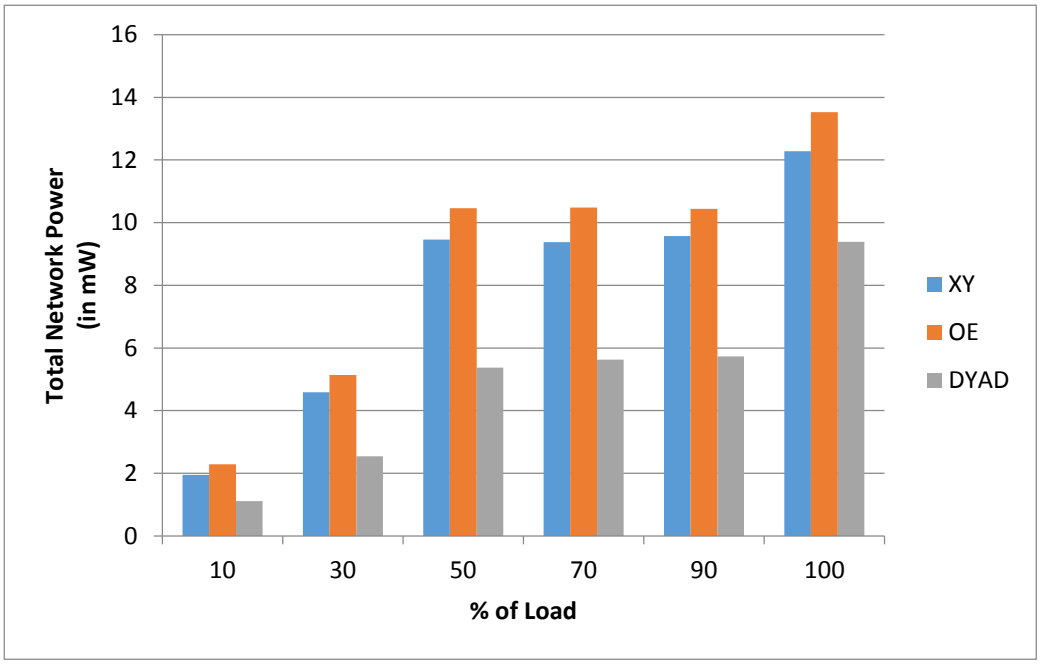
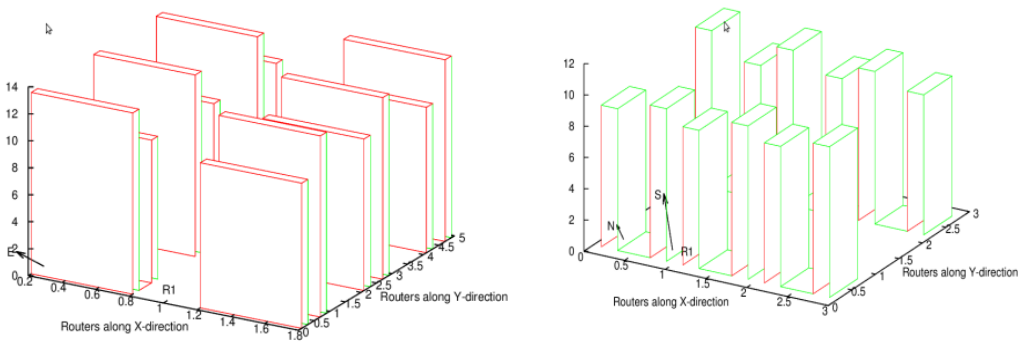


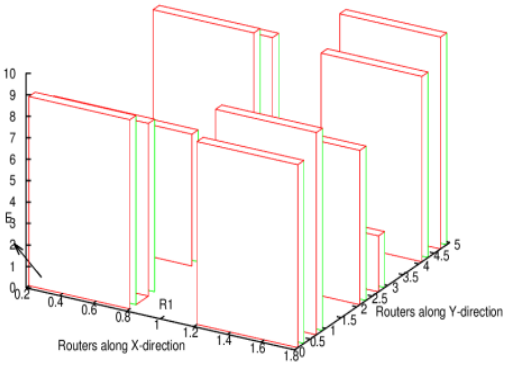
Figure 4.7 % of Load vs Total Network Power

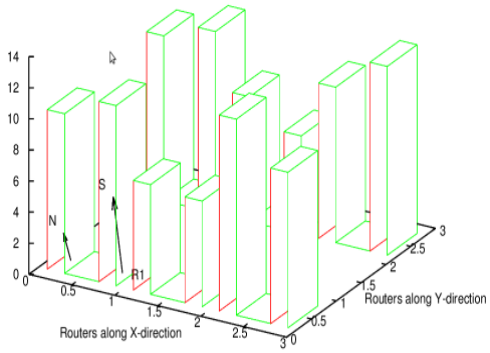


X-Latency

Y-Latency

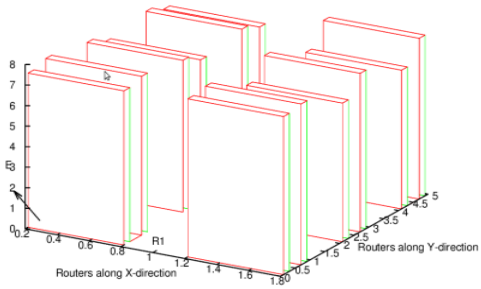
Figure 4.8 Average latency per packet for XY routing

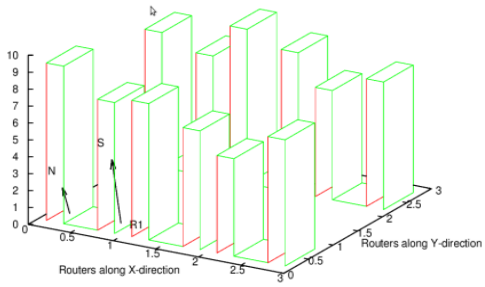




X-Throughput Y-Throughput

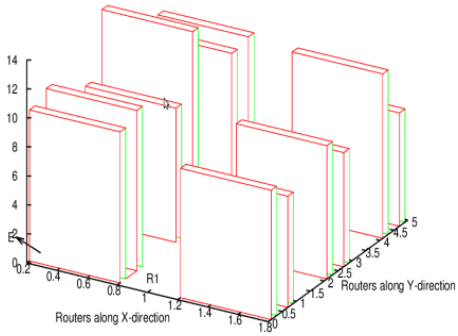
Figure 4.9 Average Throughput for OE routing

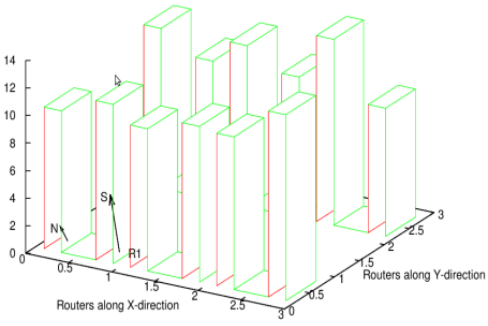




X-Throughput Y-Throughput

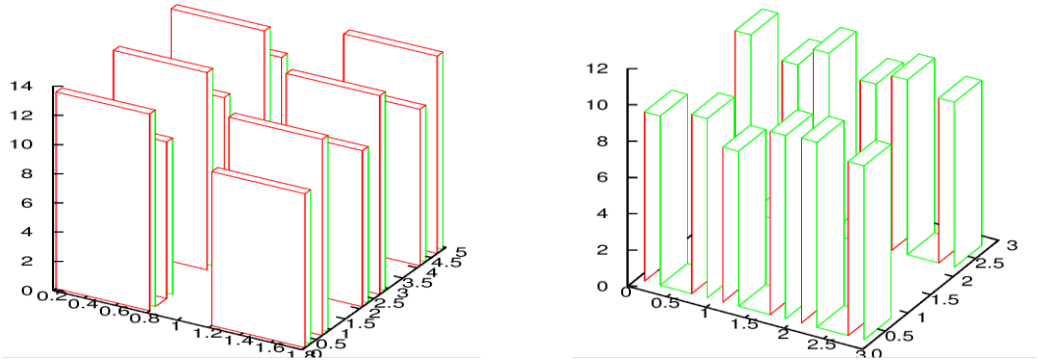
Figure 4.10 Average Throughput for XY routing





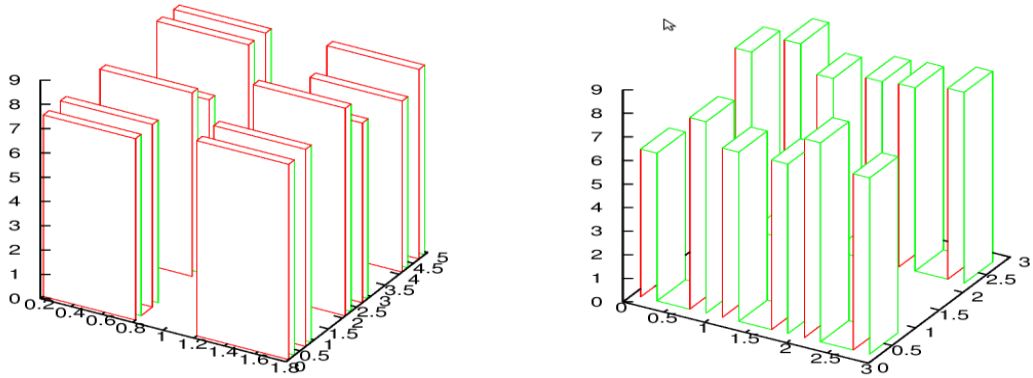
X-Latency Y-Latency

Figure 4.11 Average Latency for OE routing



X-Latency Y-Latency

Figure 4.12 Average Latency for DyAD routing



X-Throughput Y-Throughput

Figure 4.13 Average Throughput for DyAD routing

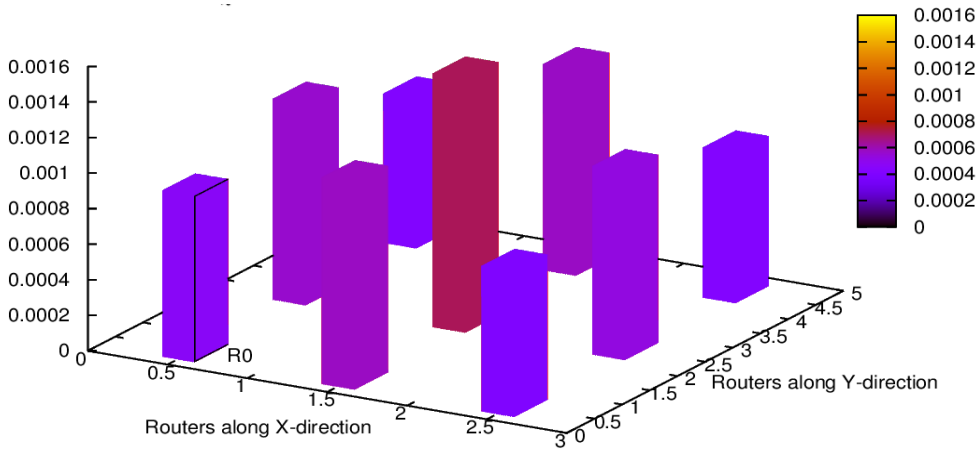


Figure 4.14 Total Network Power for XY routing

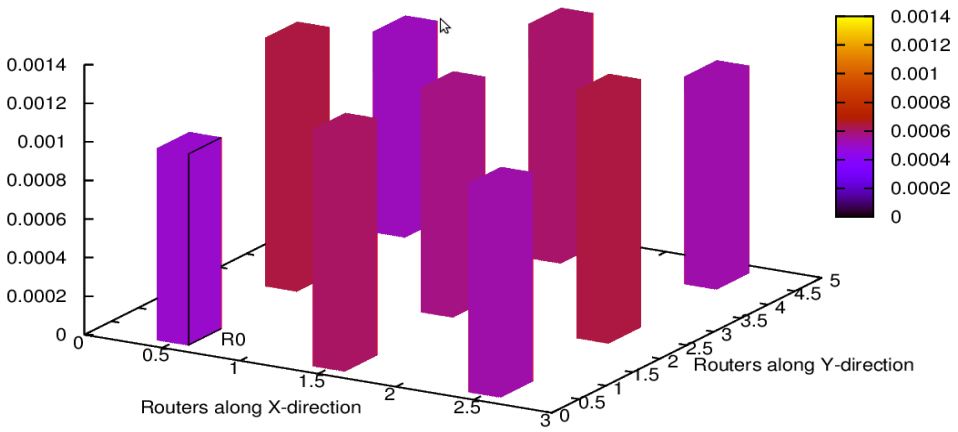


Figure 4.15 Total Network Power for OE routing

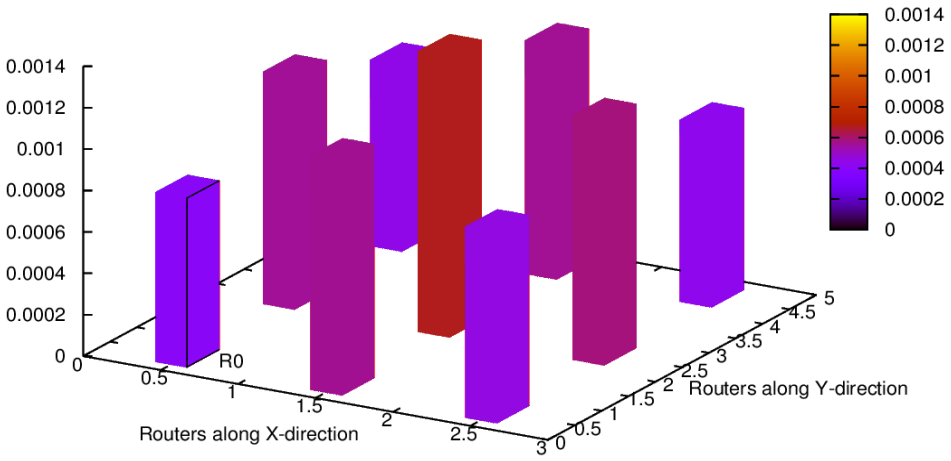


Figure 4.16 Total Network Power for DyAD routing

In NIRGAM simulator % of Load (% of maximum bandwidth utilized) is varied and according to that the effect on performance parameters (Average Latency, Average Throughput and Total Network Power) is observed for warm-up time 800 clock cycles and total simulation time 50000 clock cycles. In this experiment % of Load is varied from 10% to 100% and in Table 4.2, 4.3 and 4.4 shows the simulation results of average latency (in clock cycle per packet), average throughput (in Gbps) and total network power (in mW) respectively for XY, OE, DyAD routing algorithm. Figure 4.5,Figure 4.6 and Figure 4.7. shows the histograms of Table 4.2, 4.3 and 4.4 respectively. In Figure 4.8, Figure 4.11 and Figure 4.12 Average latency per packet graph is plotted for 9 nodes for both XY, OE and DyAD routing algorithm where X-latency indicates the latency value in X-direction(East and West) and Y-latency indicates the latency value in Y-direction (North and South) in NoC. In Figure 4.9,Figure 4.10 and Figure 4.13 Average throughput graph is plotted for 9

nodes for both OE,XY and DyAD routing algorithm where X-throughput indicates the throughput value in X-direction(East and West) and Y-throughput indicates the throughput value in Y-direction (North and South) in NoC. Figure 4.14, Figure 4.15 and Figure 4.16 shows the total power results for XY, OE and DyAD routing respectively.

Performance metrics is the ratio between average throughput and average latency. More the "P" betters the Routing Algorithm.

P = Performance Metrics (Per channel basis) = Average Throughput/Average Latency

For XY Routing (50% Load) P = (8.198/13.5859) = 0.6034

For OE Routing (50% Load) P = (9.2098/12.3141) = 0.7479

For DyAD Routing (50% Load) P = (10.5469/11.9326) = 0.8836

4.6 Discussion

OE routing algorithm has less overall average latency than XY routing algorithm. In average throughput case we can see the complete dominance of OE over XY routing algorithm. But DyAD routing algorithm consistently outperforms XY and OE routing in latency and throughput case. But in total network power case OE has more power consumption than XY for all load conditions. So XY routing algorithm is more power efficient than OE routing algorithm. From Table III it is proved that DyAD routing algorithm is more power efficient than both XY and OE routing. The Performance metrics shows that OE routing algorithm is better routing algorithm than XY routing algorithm and DyAD routing algorithm is better than both XY and OE routing algorithm in performance aspects i.e. latency, throughput and total network power.

Chapter 5

Survey of Deadlock-free Routing algorithms in NoC

In this chapter detail of some algorithms is discussed. They have the common characteristics that they are deadlock-free in nature. The deadlock-free algorithms are XY (a deterministic DOR routing), Odd Even Routing (Turn model based adaptive routing), West first, North Last, Negative First (Turn model based partially adaptive routing) and DyXY (minimal adaptive routing). Then we are going to compare their performances in different traffic pattern by applying network load.

5.1 Deterministic Routing algorithm

XY routing belongs to this routing algorithm. The description of this algorithm is in chapter 4.

5.2 Turn model Based Routing algorithms

In this section we are going to discuss some algorithms which are partially adaptive in nature. Adaptive turn model routing [22] eliminates the minimum set of turns needed to achieve deadlock freedom while retaining some path diversity and potential for adaptivity. With dimension order routing only four possible turns are permitted of the eight turns available in a two dimensional mesh. Turn model routing increases the flexibility of the algorithm by allowing six out of eight turns. Only one turn from each cycle is eliminated. In partially adaptive routing algorithms, more than one path will be calculated between source and destination. According to the network congestion condition one path will be selected for routing and in that path routing is done throughout the process. The common characteristic of these routing algorithms are their deadlock free nature. Because it is shown in previous chapter that right turn restriction can make a routing algorithm deadlock free.

5.2.1 West First Routing Algorithm

It is a partially adaptive routing algorithm. In mesh network it applies two routing restriction at any node i.e. South to West Turn and North to West Turn. The packets can't take a turn from SOUTH port to WEST port or NORTH port to WEST port of a node shown in Figure 5.1. According to this algorithm a message must first travel in the West direction (if necessary) before traveling in any other direction. After travelling to WEST direction it can adaptively route through SOUTH, EAST and NORTH direction. Message can't be routed in WEST direction later because of the routing restrictions applied. West first routing restricts at least half of the source and communication to one minimal path while rest of the pairs can communicate full adaptively [24].

Algorithm Description:

INPUTS: Current router address- $X_{current}$ is (X-coordinate of Current router) and $Y_{current}$ is (Y-coordinate of Current router).

Destination router address- $X_{destination}$ is (X-coordinate of Destination router) and $Y_{destination}$ is (Y-coordinate of Destination router).

Procedure:

Begin

$Ex = X_{destination} - X_{current}$

$Ey = Y_{destination} - Y_{current}$

```
If (Ex < 0) then        // West directed packet
        Select Output Channel = WEST;
End if;
If (Ex > 0 and Ey < 0) then
        Select Output Channel between (EAST, SOUTH);
End if;
If (Ex > 0 and Ey > 0) then
        Select Output Channel between (EAST, NORTH);
End if;
If (Ex > 0 and Ey = 0) then
        Select Output Channel = EAST;
End if;
If (Ex = 0 and Ey < 0) then
        Select Output Channel = SOUTH;
```

End if;

If (Ex = 0 and Ey > 0) then

Select output Channel = NORTH;

End if;

If (Ex = 0 and Ey = 0) then

Select Output Channel = LOCAL port;

End if;

End.

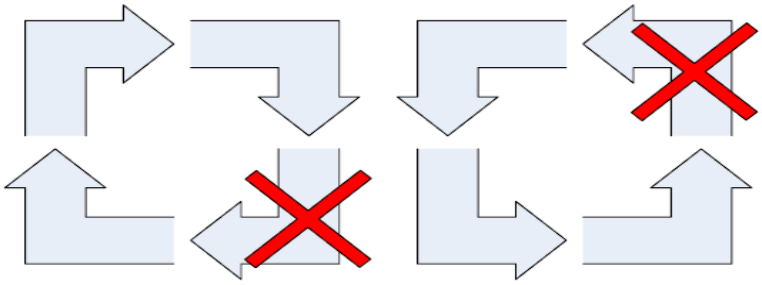


Figure 5.1 Allowed turns in west first routing

Suppose in a network SOURCE node is (1,1) and destination node is (2,4). Figure 5.2 shows the possible routing paths in West first routing algorithm.

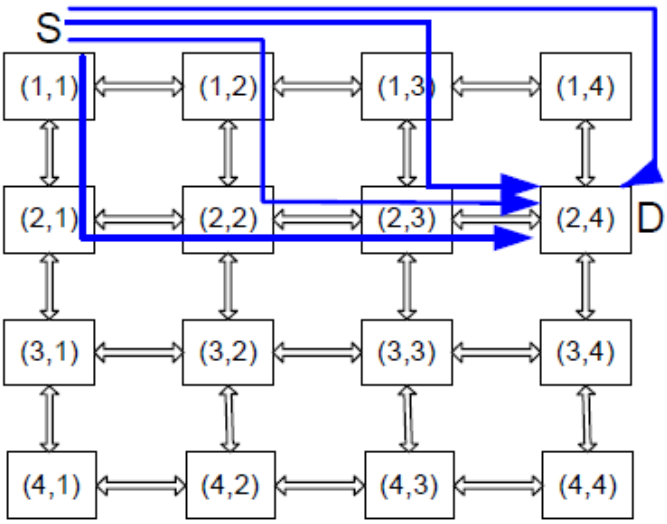


Figure 5.2 Example of west first routing

5.2.2 North Last Routing Algorithm

It is a partially adaptive routing algorithm. In mesh network it applies two routing restriction at any node i.e. North to West Turn and North to East Turn. The packets can't take a turn from NORTH port to WEST port or NORTH port to EAST port of a node shown in Figure 5.3 . According to this algorithm the message will be routed in NORTH direction only if it is the last direction to be travelled. Once a message has turned North, no further turns are permitted; hence, the North turn must be made last. In this algorithm message is routed first adaptively in WEST, SOUTH and EAST direction and at last in NORTH direction. If in a communication it is needed to move a packet in NORTH direction along with other directions then that packet should traverse in that other direction that in the end towards NORTH direction.

Algorithm Description:

INPUTS: Current router address- $X_{current}$ is (X-coordinate of Current router) and $Y_{current}$ is (Y-coordinate of Current router).

Destination router address- $X_{destination}$ is (X-coordinate of Destination router) and $Y_{destination}$ is (Y-coordinate of Destination router).

Procedure:

Begin

$Ex = X_{destination} - X_{current}$

$Ey = Y_{destination} - Y_{current}$

If (Ey > 0 and Ex < 0) then

    Select Output Channel between (NORTH, EAST);

End if;

If (Ex >= 0 and Ey > 0) then

    Select Output Channel = EAST;

End if;

```
If (Ex < 0 and Ey < 0) then
        Select Output Channel between (NORTH, WEST);
End if;
If (Ey < 0 and Ex >= 0) then
        Select Output Channel = WEST;
End if;
If (Ey = 0 and Ex > 0) then
        Select output Channel = SOUTH;
End if;
If (Ex < 0 and Ey = 0) then
        Select Output Channel = NORTH;
End if;
If (Ex = 0 and Ey = 0) then
        Select Output Channel = LOCAL port;
End if;
End.
```

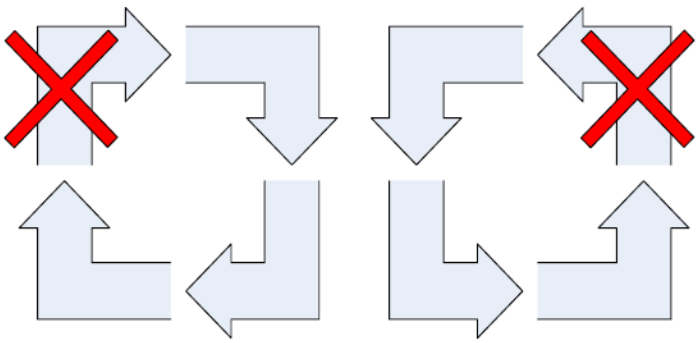


Figure 5.3 Allowed turns in north last routing

Suppose in a network SOURCE node is (1,1) and destination node is (2,4). Figure 5.4 shows the possible routing paths in North Last routing algorithm.

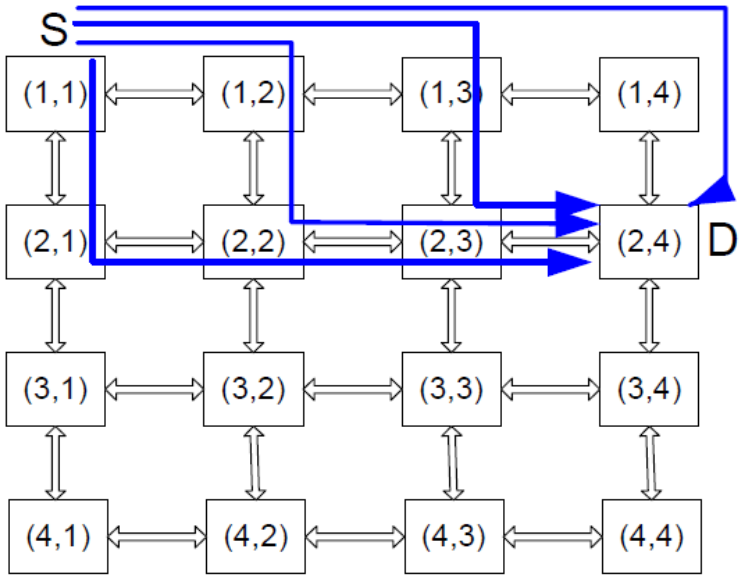


Figure 5.4 Example of North last routing

5.2.3 Negative First Routing Algorithm

It is a partially adaptive routing algorithm. In mesh network it applies two routing restriction at any node i.e. North to West Turn and East to South Turn. The packets can't take a turn from NORTH port to WEST port or EAST port to SOUTH port of a node shown in Figure 5.6. According to this algorithm a message can take all other turns except the turns from positive direction to negative direction where the positive directions are NORTH and EAST and negative directions are SOUTH and WEST. So a message travels in the negative directions (west and south) first before it is permitted to travel in positive directions (east and north). In this algorithm packet is routed adaptively in WEST and SOUTH direction and then adaptively in EAST and NORTH direction. This algorithm can be minimal or non-minimal. But the non-minimal version is more adaptive and fault tolerant.

Algorithm Description:

INPUTS: Current router address- $X_{current}$ is (X-coordinate of Current router) and $Y_{current}$ is (Y-coordinate of Current router).

Destination router address- $X_{destination}$ is (X-coordinate of Destination router) and $Y_{destination}$ is (Y-coordinate of Destination router).

Procedure:

Begin

$Ex = X_{destination} - X_{current}$

$Ey = Y_{destination} - Y_{current}$

If (Ey > 0 and Ex > 0) then

    Select Output Channel between (SOUTH, EAST);

End if;

If (Ex <= 0 and Ey > 0) then

    Select Output Channel = EAST;

End if;

If (Ex < 0 and Ey < 0) then

    Select Output Channel between (NORTH, WEST);

End if;

If (Ey < 0 and Ex >= 0) then

    Select Output Channel = WEST;

End if;

If (Ey = 0 and Ex > 0) then

    Select output Channel = SOUTH;

End if;

If (Ex < 0 and Ey = 0) then

    Select Output Channel = NORTH;

End if;

If (Ex = 0 and Ey = 0) then

    Select Output Channel = LOCAL port;

End if;

End.

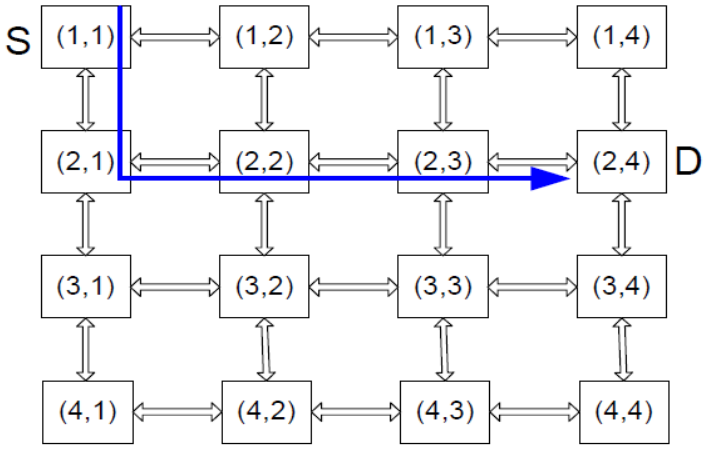


Figure 5.5 Example of Negative first routing

Suppose in a network SOURCE node is (1,1) and destination node is (2,4). Figure 5.5 shows the possible routing paths in North Last routing algorithm.

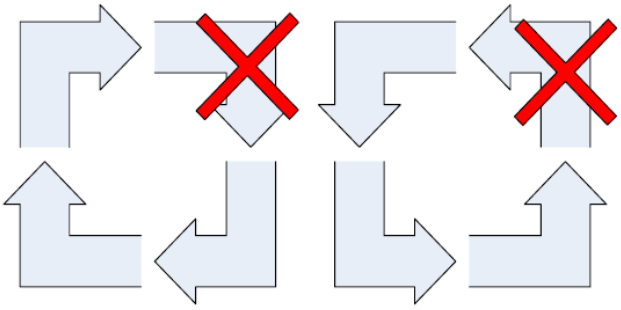


Figure 5.6 Allowed turns in Negative first routing

5.2.4 OE Routing Algorithm

OE routing is abbreviated as Odd Even routing algorithm. It is a turn model based adaptive routing algorithms. The details of this algorithm is described in Chapter 4.

5.3 Dynamic XY Routing Algorithm

As discussed earlier deterministic routing gas some advantages of simplicity in router design and there is degradation in throughput value when packet injection ratio increases. So XY routing algorithm is modified to an algorithm which is congestion aware in nature and having deadlock free nature [23]. It is proved that adaptiveness decreases the hotspot node formation and avoidance of faulty components. The adaptiveness can acquire by any network by monitoring the congestion level continuously. The deadlock and livelock free nature can be acquired by adding some routing restriction or by traversing through the shortest route between source to destination.

5.3.1 Routing Description

According to Dynamic XY routing algorithm data has to traverse through a route between source to destination which has least length [26]. If there exists more than one route, the routing algorithm will choose an output channel for the packet which has less congestion value.

Summary of routing algorithm:

1. Destination of incoming data packet or flits is to be identified from the header.
2. The (x,y) coordinate of current router is compared with the (x,y) of destination router.
    a. If destination router address is equal to current router address then it had reached destination. Local port is selected as output channel.
    b. If they are not equal, the routing is done like Static XY routing algorithm as discussed in chapter 4. The packets will traverse first in X direction then in Y direction.
    c. Else STRESS (Congestion) values of current router's neighbors is compared and data packets will be traversed to the neighbor which has the smallest stress value.

The parameter which is called as STRESS value indicates whether the link is congested or not with its value in the router. This value is very important for this algorithm. The "Instant Queue Length" of each router can be considered as stress value. Simply we can say the number of cell occupied in all input buffers can be called as stress value. Each router has to store instant stress value for all neighboring routers and this value is kept on

updating by some sort of event driven mechanism. The router architecture for DyXY is shown in Figure 5.7.

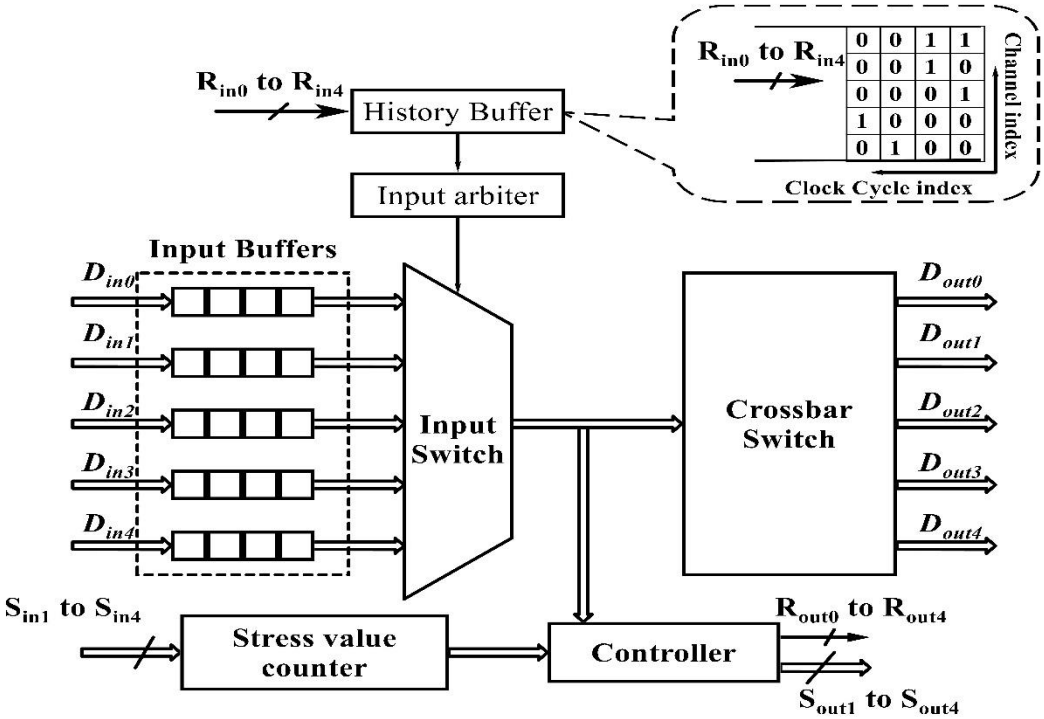


Figure 5.7 Router architecture of DyXY routing

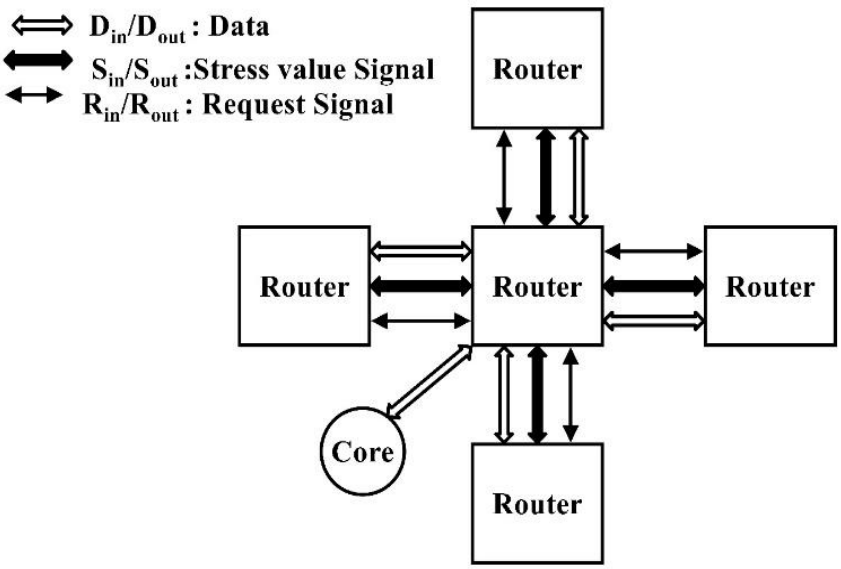


Figure 5.8 NoC interconnection during DyXY routing

5.4 Experimental setup

The experimental setups for the evaluation of the routing algorithms are 4x4 mesh topology where each node is connected with CBR (Constant Bit Rate) traffic generator of value 32 Gbps. Buffer depth (number of buffers) of input channel FIFO is 32. Number of virtual channels per physical channel is 4. Link length is 3 um. Packet size is taken as 20 bytes with flit interval 2 clock cycles and clock frequency 1 GHz

5.4 Simulation Results

5.4.1 Simulation results for Random uniform traffic

Table 5.1 % Load vs Average Latency for Random uniform traffic

% of Load	*Average Latency (Clock cycle per packet)*					
	XY	*WF*	*NL*	*NF*	*OE*	*DyXY*
10	61.8059	67.3717	64.7401	64.7401	69.1711	66.4474
30	62.7364	68.4088	66.1588	68.9579	71.3614	66.6011
50	66.4783	68.5387	67.7663	69.0121	73.661	67.2065
70	76.4973	72.819	70.047	72.137	73.923	69.532
90	81.3295	77.750	73.218	76.243	76.417	74.199
100	85.1785	81.245	78.003	83.292	78.915	76.0116

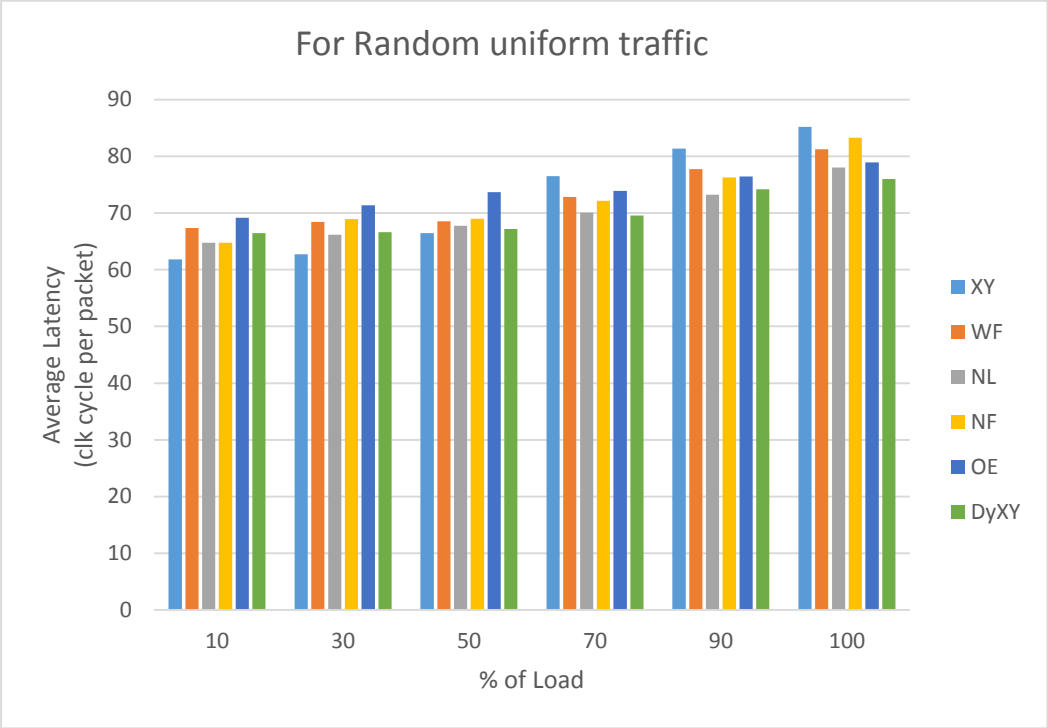


Figure 5.9 % of Load vs Average Latency for random uniform traffic

Table 5.2 % of Load vs Average Throughput for Random uniform traffic

% of Load	*Average Throughput (Gbps)*					
	XY	*WF*	*NL*	*NF*	*OE*	*DyXY*
10	2.1929	3.496	4.2254	3.219	5.367	5.119
30	4.236715	6.019	7.18764	6.168	8.021	8.365
50	5.90696	7.311	7.9612	7.521	9.336	10.186
70	7.2158	10.896	11.0123	10.801	13.861	14.013
90	8.721	12.991	13.914	12.812	16.113	16.961
100	10.157	15.124	16.009	14.112	19.379	20.137

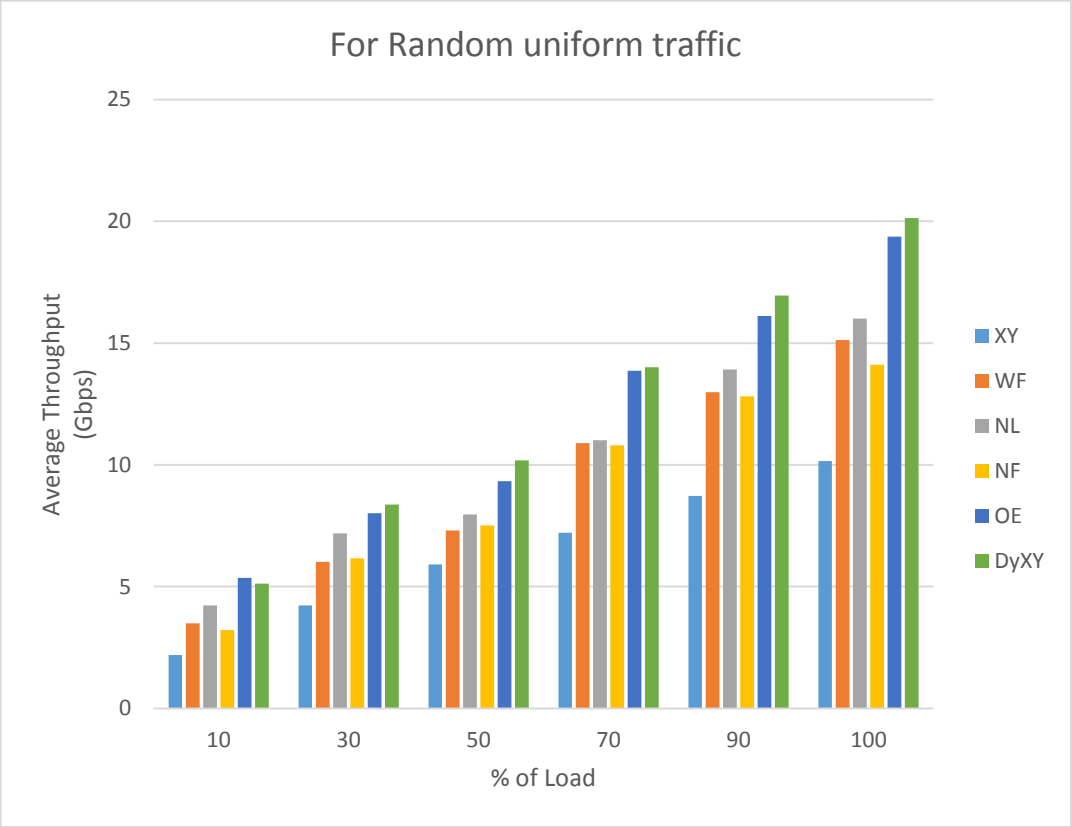


Figure 5.10 % of Load vs Average Throughput for Random uniform traffic

Table 5.3 % of Load vs Total network power for Random uniform traffic

% of Load	*Total Network Power (mW)*					
	XY	*WF*	*NL*	*NF*	*OE*	*DyXY*
10	4.285	4.305	4.309	4.306	4.68502	4.3134
30	6.391	7.6	7.057	7.758	8.936	8.958
50	9.592	10.9	10.737	10.998	12.312	11.166
70	13.191	14.714	14.211	14.8	16.236	15.01
090	17.732	18.8	18.6	18.989	20.921	19.136
100	18.148	20.395	20.1	20.411	21.2514	20.6797

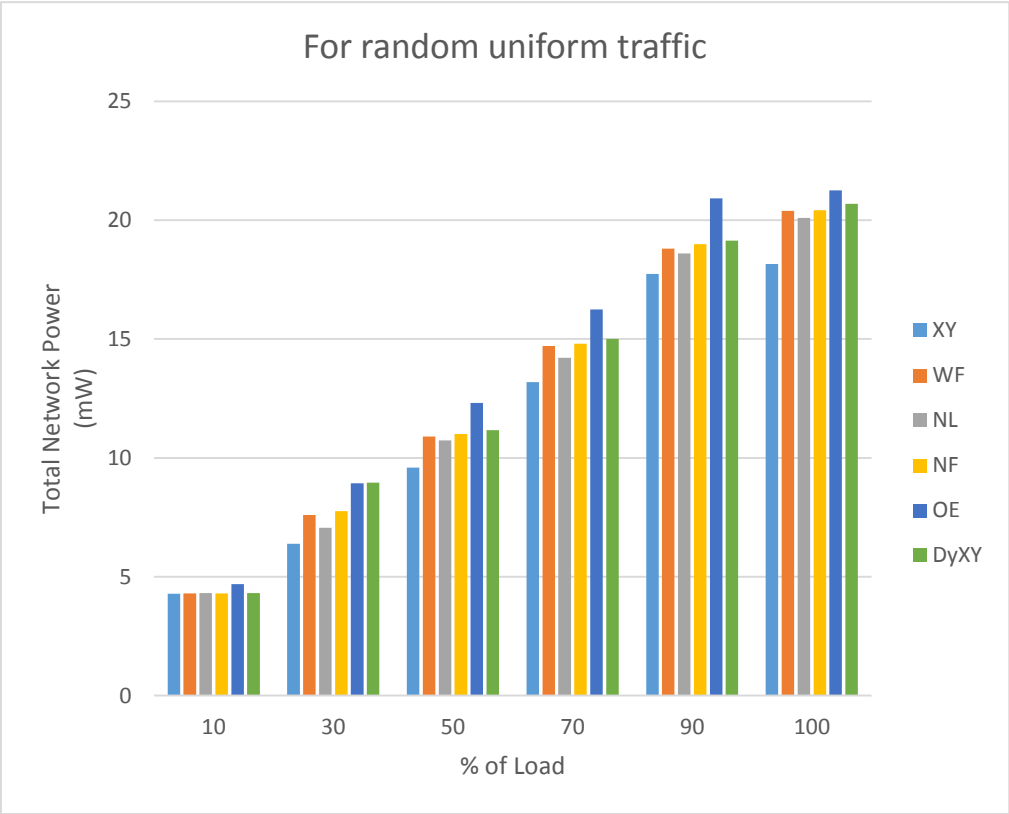


Figure 5.11 % of Load vs Total Network Power for Random uniform traffic

5.4.2 Simulation results for Bit reversal traffic pattern

Table 5.4 % of Load vs Average Throughput for Bit reversal traffic pattern

% of Load	*Average Throughput (Gbps)*					
	XY	*WF*	*NL*	*NF*	*OE*	*DyXY*
10	0.1523	0.3011	0.5614	0.4236	0.5937	0.6092
30	1.1012	1.9631	2.219	1.9812	3.925	4.2157
50	3.08712	5.6181	6.195	5.6611	11.9635	12.021
70	4.7398	8.8932	9.865	8.901	16.0125	16.450
90	5.1096	10.221	11.663	10.591	18.991	19.611
100	5.963	12.342	13.347	11.951	19.563	20.571

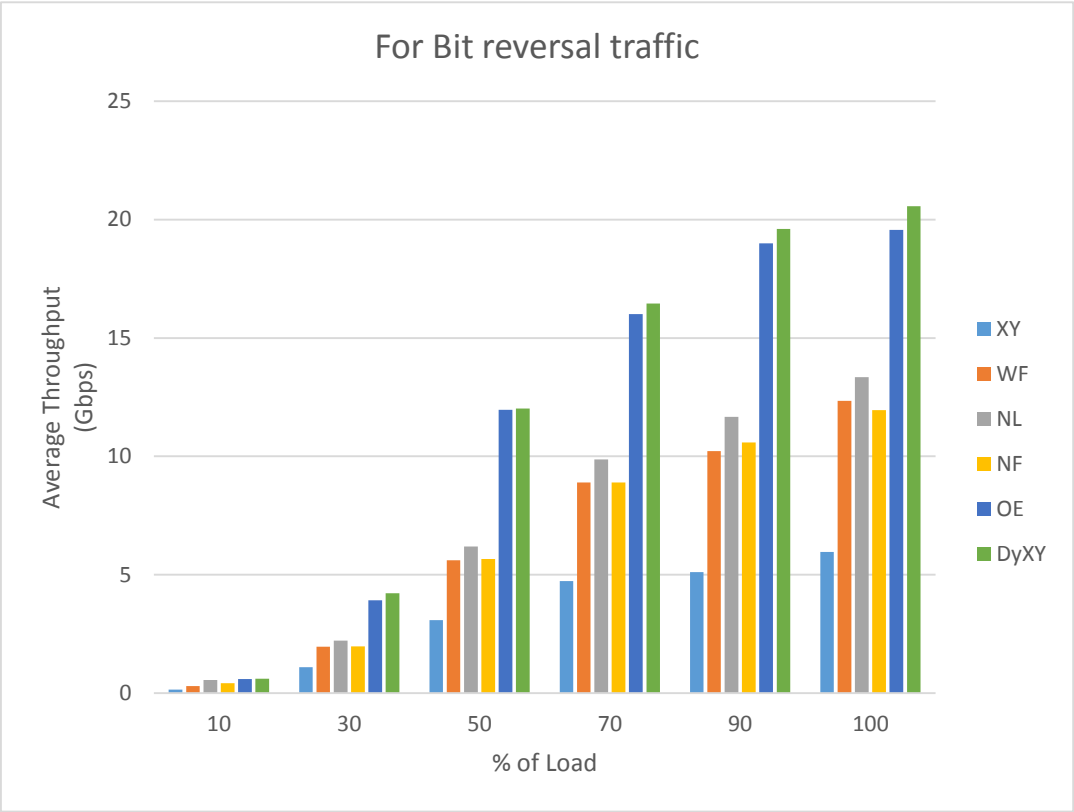


Figure 5.12 % of Load vs Average Throughput for Bit reversal traffic pattern

Table 5.5 % of Load vs Average Latency for Bit reversal traffic pattern

% of Load	*Average latency (Clock cycles per packet)*					
	XY	*WF*	*NL*	*NF*	*OE*	*DyXY*
10	62.132	64.819	62.510	64.276	66.862	65.251
30	66.215	69.368	67.211	69.924	70.566	68.813
50	69.365	70.651	69.169	71.238	72.815	70.597
70	81.329	79.549	74.389	78.912	74.393	73.127
90	87.037	85.172	81.157	85.681	79.517	78.390
100	93.112	90.217	88.035	89.936	87.919	84.541

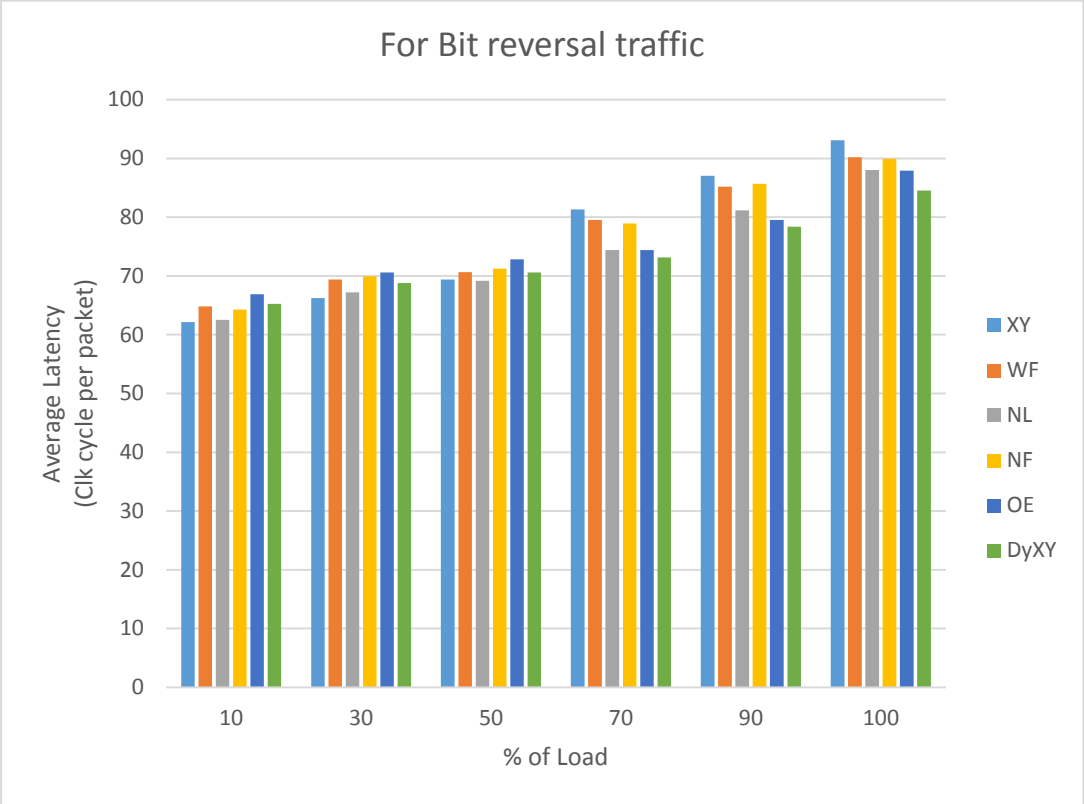


Figure 5.13 % of Load vs Average Latency for Bit reversal traffic pattern

Table 5.6 % of Load vs Total Network Power for Bit reversal traffic pattern

% of Load	*Total Network Power (mW)*					
	XY	*WF*	*NL*	*NF*	*OE*	*DyXY*
10	3.7523	3.7544	3.7544	3.7544	4.98431	4.35414
30	5.2173	6.7937	6.2109	6.6181	7.5293	6.9712
50	7.3461	8.298	8.045	8.332	9.595	8.5611
70	10.5877	11.96	11.827	11.99	13.690	12.12
90	13.1745	14.6	14.333	14.8	15.9	15.1
100	15.3036	16.2	16.019	16.3	17.2833	16.8266

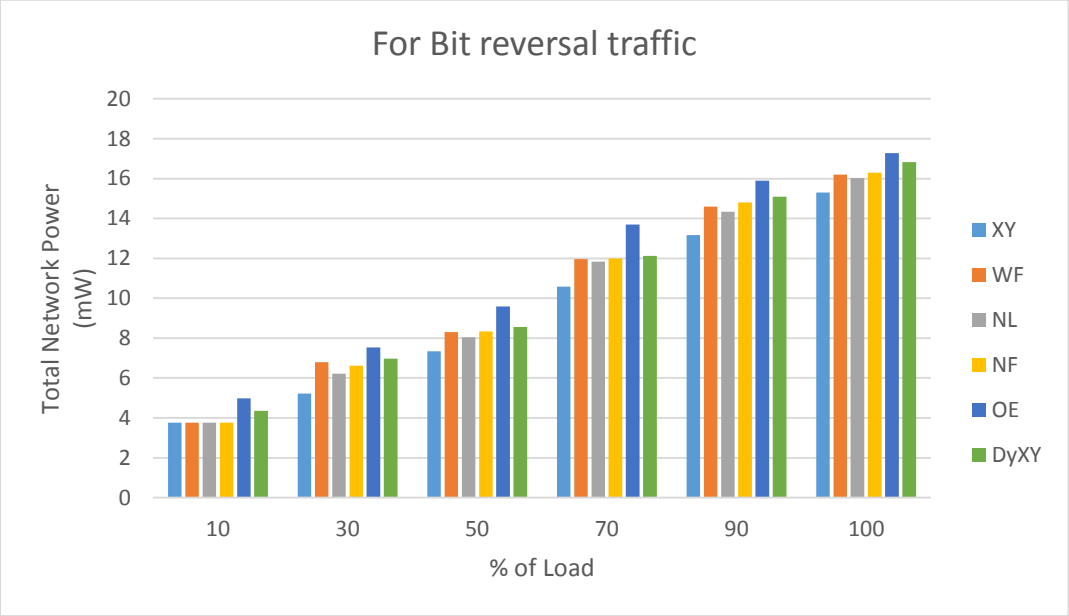


Figure 5.14 % of Load vs Total Network Power for Bit reversal traffic pattern

5.4.3 Simulation results for Bit shuffle traffic pattern

Table 5.7 % of Load vs Average Latency for Bit shuffle traffic pattern

% of Load	*Average Latency (Clock cycles per packet)*					
	XY	*WF*	*NL*	*NF*	*OE*	*DyXY*
10	67.214	71.275	69.320	72.916	75.466	73.613
30	70.616	74.169	72.923	76.245	77.329	74.391
50	75.367	80.649	76.680	79.341	78.246	75.099
70	86.873	86.646	83.649	87.582	80.293	79.001
90	95.911	89.189	87.395	91.321	85.189	83.696
100	110.012	92.129	90.694	93.913	88.101	86.872

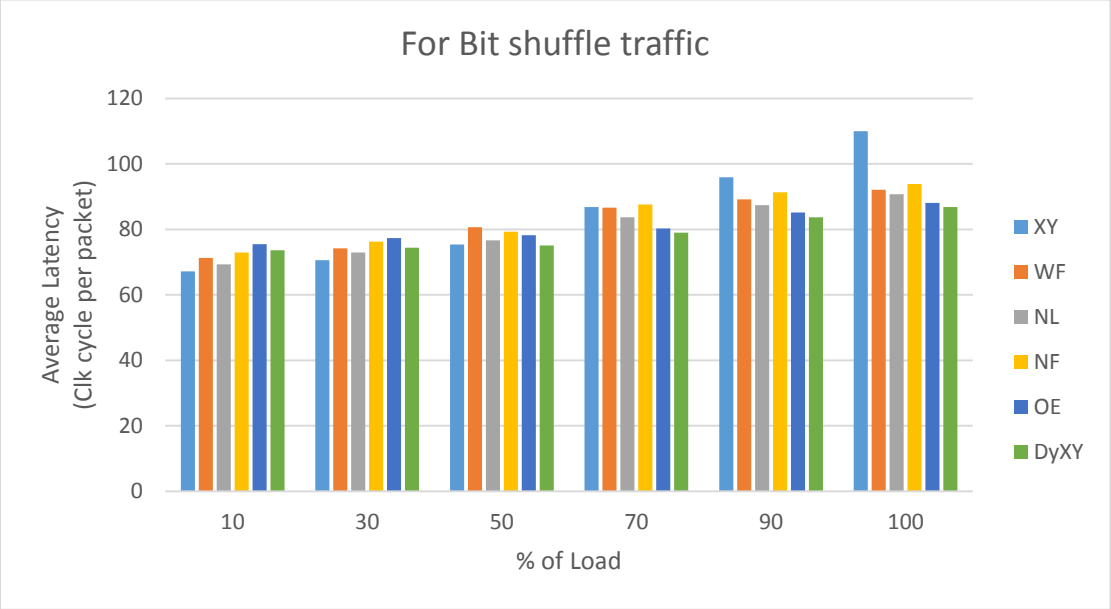


Figure 5.15 % of Load vs Average Latency for Bit shuffle traffic pattern

Table 5.8 % of Load vs Average Throughput for Bit shuffle traffic pattern

% of Load	*Average Throughput (Gbps)*					
	XY	*WF*	*NL*	*NF*	*OE*	*DyXY*
10	2.10193	2.356	2.6274	2.469	4.961	5.252
30	3.09065	3.935	4.692	4.216	6.367	7.725
50	5.664	6.392	7.295	6.313	13.9012	14.532
70	5.925	6.989	7.806	7.032	15.091	15.147
90	7.534	8.561	9.421	8.887	18.015	18.949
100	8.211	9.025	10.119	9.468	24.895	25.365

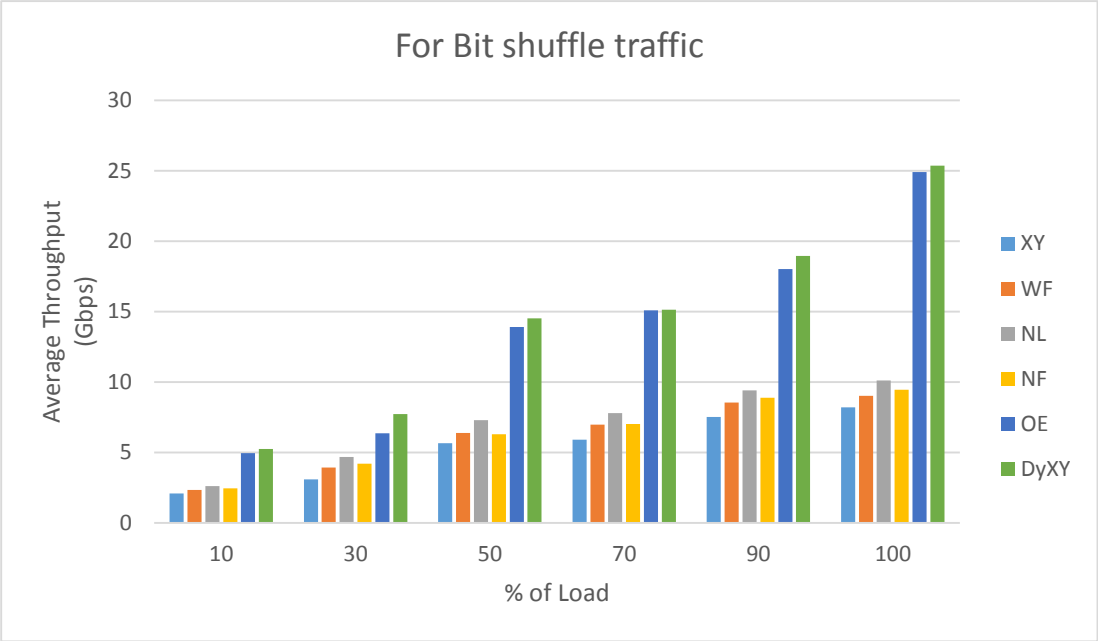


Figure 5.16 % of Load vs Average Throughput for Bit shuffle traffic pattern

Table 5.9 % of Load vs Total Network Power for Bit shuffle traffic pattern

% of Load	*Total Network Power (mW)*					
	XY	*WF*	*NL*	*NF*	*OE*	*DyXY*
10	3.3282	4.918	4.312	4.8292	4.90186	4.63829
30	4.135	5.325	5.621	5.7039	6.929	6.01462
50	7.439	8.691	8.393	8.9335	10.214	9.2758
70	10.307	11.801	11.624	11.914	13.238	12.380
90	12.836	14.036	13.913	14.029	15.857	14.761
100	13.5579	14.914	14.678	14.845	16.9738	15.0451

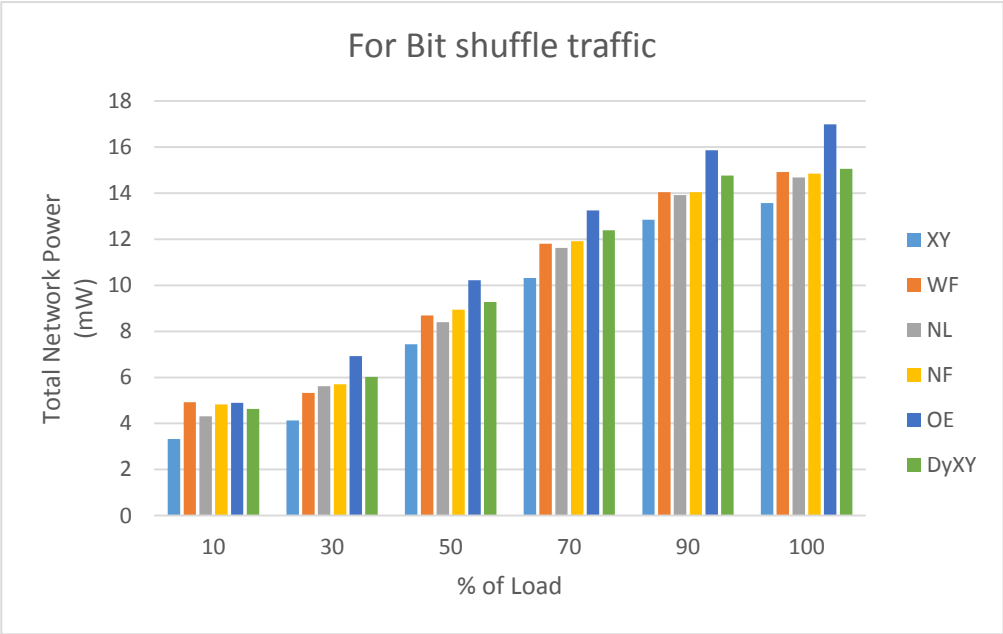


Figure 5.17 % of Load vs Total Network Power for Bit shuffle traffic pattern

5.5 Discussion

In NIRGAM simulator % of Load (% of maximum bandwidth utilized) is varied and according to that the effect on performance parameters (Average Latency, Average Throughput and Total Network Power) is observed for warm-up time 800 clock cycles and total simulation time 50000 clock cycles. In this experiment % of Load is varied from 10% to 100% and in Table 5.1, 5.2 and 5.3 shows the simulation results of average latency (in clock cycle per packet), average throughput (in Gbps) and total network power (in mW) respectively for XY, West First routing, North Last routing, Negative First routing, OE, DyXY routing for random uniform traffic pattern. Table 5.4, 5.5 and 5.6 shows the simulation results of average latency (in clock cycle per packet), average throughput (in Gbps) and total network power (in mW) respectively for XY, West First routing, North Last routing, Negative First routing, OE, DyXY routing for bit reversal traffic pattern. Table 5.7, 5.8 and 5.9 shows the simulation results of average latency (in clock cycle per packet), average throughput (in Gbps) and total network power (in mW) respectively for XY, West First routing, North Last routing, Negative First routing, OE, DyXY routing for bit shuffle traffic pattern. Figure 4.5, 4.6 and 4.7 shows the histograms of Table 5.8, 5.9 and 5.10 respectively. Performance metrics is the ratio between average throughput and average latency. More the “P” betters the Routing Algorithm.

P = Performance Metrics (Per channel basis) = Average Throughput/Average Latency

In Random uniform traffic pattern

For XY Routing (50% Load) P = (5.90696 / 66.4783) = 0.0885

For West First Routing (50% Load) P= (7.311 / 68.5387) = 0.1066

For North Last Routing (50% Load) P= (7.9612/67.7663) = 0.1174

For Negative First Routing (50% Load) P= (7.521 / 69.0121) = 0.1089

For OE Routing (50% Load) P = (9.336 / 73.661) = 0.1267

For DyXY Routing (50% Load) P = (10.186 / 67.2065) = 0.1515

In Bit reversal traffic pattern

For XY Routing (50% Load) P = (3.08712 / 69.365) = 0.04450

For West First Routing (50% Load) P= (5.6181 / 70.651) = 0.07951

For North Last Routing (50% Load) P= (6.195 / 69.169) = 0.089563

For Negative First Routing (50% Load) P= (5.6611 / 71.283) = 0.07941

For OE Routing (50% Load) P = (11.9635 / 72.815) = 0.1642

For DyXY Routing (50% Load) P = (12.021 / 70.597) = 0.17027

In Bit shuffle traffic pattern

For XY Routing (50% Load) P = (5.664 / 75.367) = 0.07515

For West First Routing (50% Load) P= (6.392 / 80.649) = 0.7925

For North Last Routing (50% Load) P= (7.295 / 76.680) = 0.09513

For Negative First Routing (50% Load) P= (6.313 / 79.341) = 0.07956

For OE Routing (50% Load) P = (13.9012 / 78.246) = 0.1776

For DyXY Routing (50% Load) P = (14.532 / 75.099) = 0.1935

From the above results and discussion we are concluding that DyXY performs better than other algorithms in random uniform, bit reversal and bit shuffle traffic pattern. Among partial adaptive routing algorithms i.e. west first, north last and negative first routing algorithms North Last algorithm shows best performance in latency, throughput and power point of view. Power efficiency of DyXY is more than OE routing. Power efficiency of routing algorithms is in the order shown i.e. Deterministic routing > Partial Adaptive routing > Adaptive routing. XY routing algorithm is the most power efficient algorithm but it suffers from performance degradation in heavy network load.

Chapter 6

Performance Evaluation of a Fault Tolerant Routing Algorithm in NoC

6.1 Introduction to Routing faults in NoC

VLSI industry introduces CMOS technology. It results in scale down in SoC. Many systems can be introduced in a single chip. Many processors and storage devices can be introduced into a single chip. All these elements are important for the increased performance in the network on chip. NoC is more sensitive to various intrusions. Due to these intrusions, two types of faults occurred i.e. I) Permanent faults II) Transient faults. These permanent faults are occurred by aging effects like electro-migration and poor technology. During fabrication this kind of problem may occur. Transient faults include soft upsets. It also includes cross talk and coupling noise. Basically here we are concerned about permanent faults which occurred due to failure of a router or a processing element. This failure is classified into two types i.e. i) Node failure ii) Link failure. In node failure the whole router is considered as faulty. It occurs due to the problem in architecture of the router e.g. arbiter, Queue etc. But link fault indicates fault is in the links (input or output channels) which are connected to other routers. When there is node failure, this will indicate all the links of the router is faulty. This node failure can't be repaired or compensated online in the process of routing. Data packets can't traverse through that node. But if there is link faults data packets can manage to find another router with the help of other functional links. To avoid these above mentioned problem a Network on Chip must have fault tolerance capability.

6.2 Fault tolerant Routing algorithm

For a fault tolerant NoC we need a fault tolerant routing algorithm. This kind routing algorithm should improve the performance of NoC as well as reduce the packet or network latency. It should also be concerned about power consumption. There are various kinds of fault tolerant algorithms are available [24]. They are i) Detour routing algorithm ii) Backtracking routing algorithm. But they have more power consumption issues. They are not at all power efficient. So they are not preferable for NoC. Another commonly used fault tolerant routing algorithm are i) Link state based routing ii) Flooding based routing iii) Combination of both. According to flooding based routing many copies of a packet is sent through different links of network. As a result network load will increase heavily and it can cause network congestion. Instability in network may occur. It will increase the power consumption. Dynamic smart routing like DyAD or turn model routing like North Last can give good results as compared to these flooding and link based routing in latency,

throughput and power consumption point of view. But the problem is that their performance will degrade heavily if the number of faults will increase beyond certain limits [26].

6.3 DyXY based Fault tolerant Routing Algorithm

But in this chapter we are going to discuss about a fault tolerant routing algorithm which is based on dynamic XY routing algorithm which has a advantage of routing the packets by avoiding congested links and it is also power efficient. Latency and throughput values are also acceptable. This algorithm can distribute the traffics ejected from the nodes evenly or uniformly throughout the network to avoid congestion problem.

6.3.1 Router Architecture for this Routing algorithm

A 2D Mesh NoC consists of tiles with a number of M X N, where M is number of rows and N is number of rows. A tile consists of a processing element and a network switch or router. Here PE is connected to the router by the resource network interface. The router has four bidirectional links which both input and output channels with buffers. Data are transmitted in the form of packet or flits between different tiles. The route between the source and destination is decided by the routing algorithm. A simple router structure suitable for fault-tolerant routing algorithm is shown in (Fig. 1). The router has two main parts: i) the input port and the output port. The input port includes Input Arbiter, Input Buffer, and Input Switch. The output port is composed of Output Switch and Penalty Table. In this router the two ports are multiplexed by internal control unit. In Figure 6.1, the subscript number in every signal represents different ports, 0-4 representing PE, Port N, Port S, Port E, and Port W, respectively.

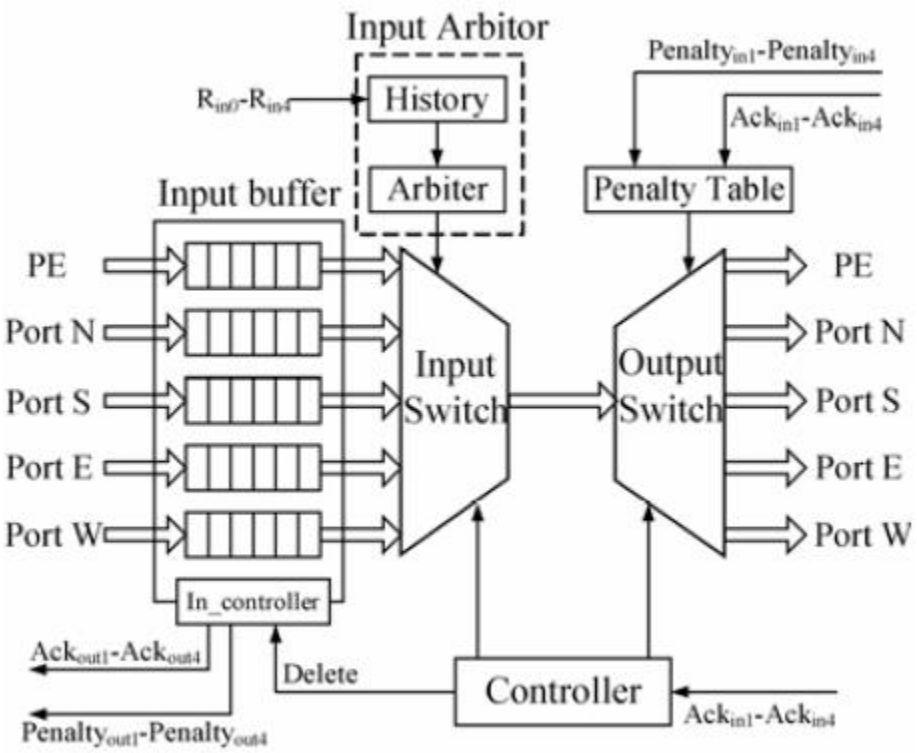


Figure 6.1 Router Architecture for FT_DyXY Routing Algorithm

6.3.2 Principle of Algorithm

The approach of this routing algorithm is shown. Here the source node is (0,0) and destination node is (3,3). According to XY routing, the data packet P will first traverse to (1,0). Then the packet is moved to from (1,0) to the next X-direction node i.e. (2,0). Again after checking the X-coordinate of destination node the packet will move from (2,0) to (3,0). Node (3,0) is in the destination column. Then packet will move in Y-direction. After checking the Y-coordinates the packet will move from (3,0) to (3,1). After that packet will move from (3,1) to (3,2) then (3,2) to (3,3). The XY routing path is shown in solid line. Suppose this NoC has a faulty node (2,0). If the routing is done by XY routing the packet reach the destination node. According to the fault-tolerant routing algorithm, the data packet P will first traverse to (1,0) by comparing the X-coordinate of destination. Then the packet has to move to from (1,0) to the next X-direction node i.e. (2,0). But the node (2,0) is found to be faulty. So according to the algorithm that packet will move to (1,1) and then check for the less congested node. Let (1,2) has less congestion value than (2,1). So the

packet will move to (1,2). Like that it reached the destination (3,3) by checking the congestion value by multi-level congestion control mechanism which is the special characteristics of DyXY routing algorithm. So the path of P will be (0, 0)→(1, 0)→(1, 1)→(1, 2)→(1, 3)→(2, 3)→(3, 3) shown as the broken line in Fig.2

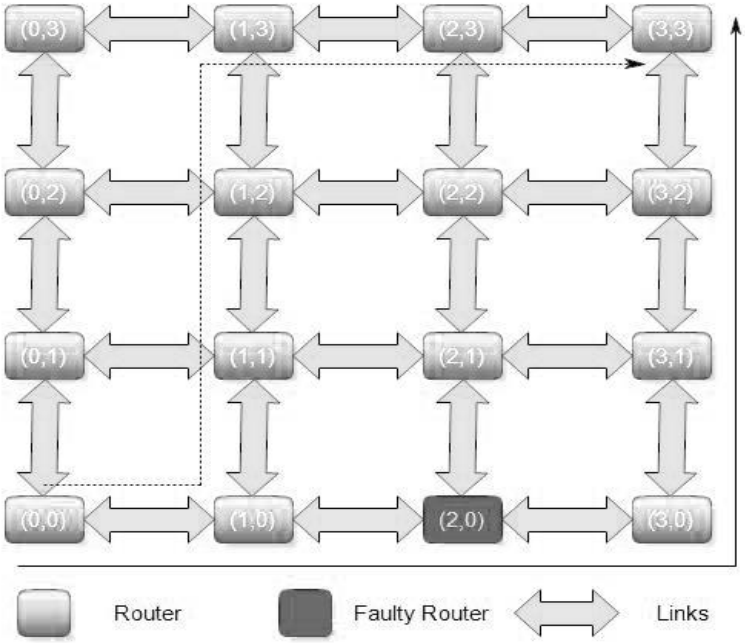


Figure 6.2 Example of Fault tolerant routing

Like that final route is calculated. When NoC is fault free, this fault tolerant routing algorithm will simply follow the dynamic XY routing policy by using its multi-level congestion control mechanism to avoid congested links. Else it choose a route by considering the fault position and the stress values of routers connected to the current router. According to this algorithm, every link connected to the router having faults or link is marked as unsafe, and the packet cannot be routed through that faulty links or unsafe links. To determine the dynamic faults which occurs in between routing process a mechanism is used. According that mechanism when the sender node sends a packet to another node, it will also wait for the receiving node to send an acknowledgement signal. If it will not receive that signal it will send to that node. If number of sent packets is greater than a predefined threshold limit, then that link will be marked as unsafe and next time no node will send any packets through that link. Considering the severe congestion around faults, a congestion control mechanism based on multi-level congestion scheme is designed in our algorithm. Every router owns a penalty table or stress value table where the penalty values or the stress values are stored for each output channels. The corresponding penalty

value is generated by its neighboring routers, and it represents the amount of packets in the input buffers of the routers. In our algorithm, according to the penalty value, we define three different congestion levels as shown in Table 6.1.

Table 6.1 Table for determining congestion level

Buffer occupancy percentage of the neighboring router	Congestion Level
20% to 40%	Low
40% to 80%	Moderate
80% above	Severe

Different levels have different control methods:

1. When there is no congestion in the network which has to found by checking the buffers of neighboring router, then routing is done with XY routing policy.
2. When there is low congestion values in the network, then the penalty values are compared and port having least penalty value will be selected for routing.
3. When there is moderate and severe congestion level in the network, the difference between the penalty values will be measured. This will indicate the effect of distributing packets. If the difference is more than the threshold which is predefined, then the port having less penalty value will be chosen. Else Simply XY routing algorithm will be followed.
4. When output ports are having different congestion level, then according to the priority the ports will be selected. First priority is given to the “NO CONGESTION” level. Then priority will be given to “LOW CONGESTION” level. Lastly ports having “MODERATE” and “SEVERE” congestion level will be served.

As a result there exist a effective control over the packets and like that hotspots will be neglected around the faulty nodes.

6.3.3 Implementation of routing algorithm

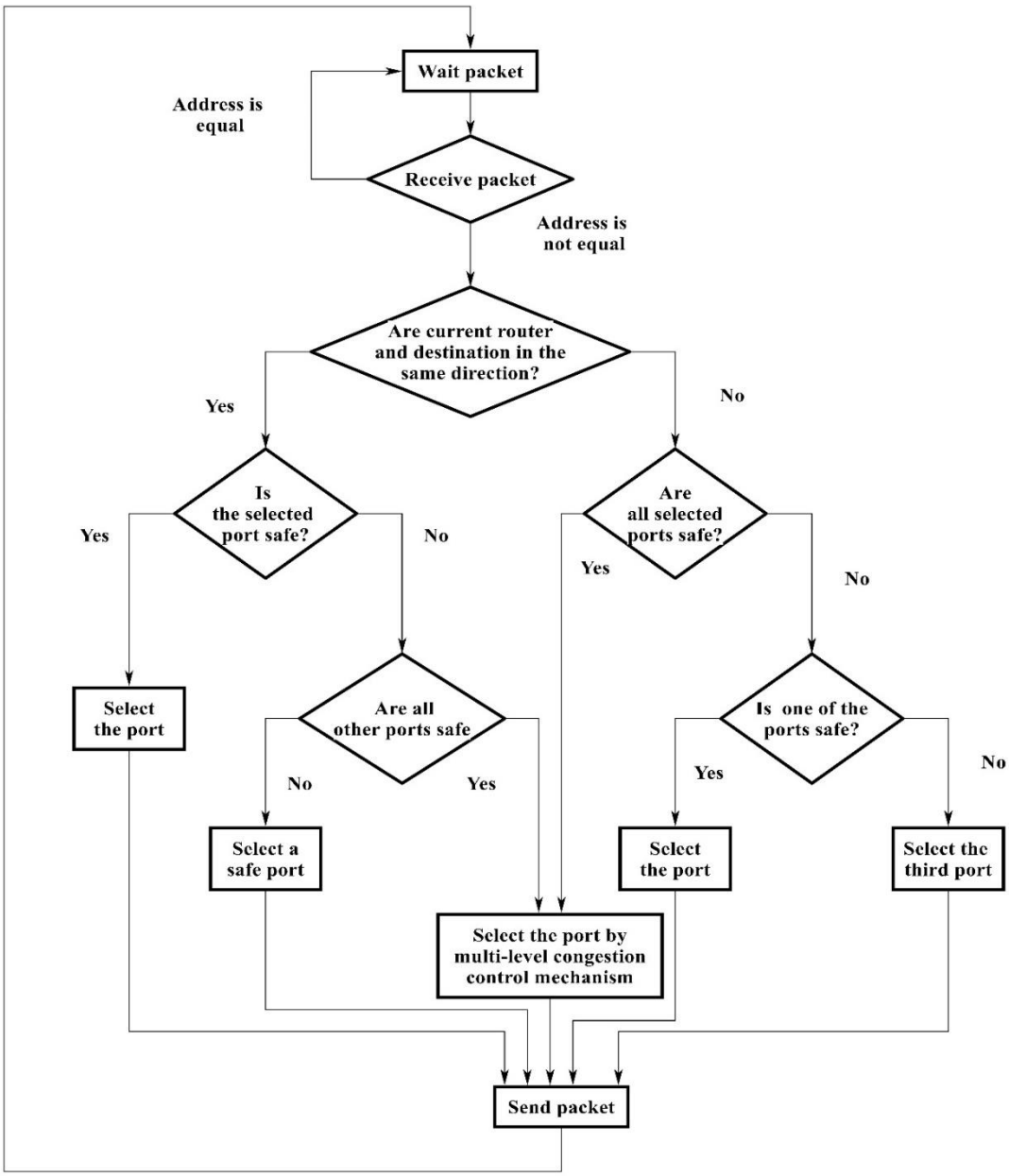


Figure 6.3 Flow diagram of Fault tolerant routing

6.4 Simulation Results

Simulation results for Random uniform traffic pattern

Table 6.2 Number of faults vs % of Packet loss For Random uniform traffic pattern

No. of faults	*% of Packet Loss*						
	XY	***WF***	***NL***	***NF***	***OE***	***DyXY***	***FT_DyXY***
2	49.05	40.61	38.28	42.36	42.95	45.67	22.8
4	57.85	48.33	45.91	47.32	49.157	52.112	24.39
6	74.36	58.97	54.54	61.19	64.28	66.81	38.88

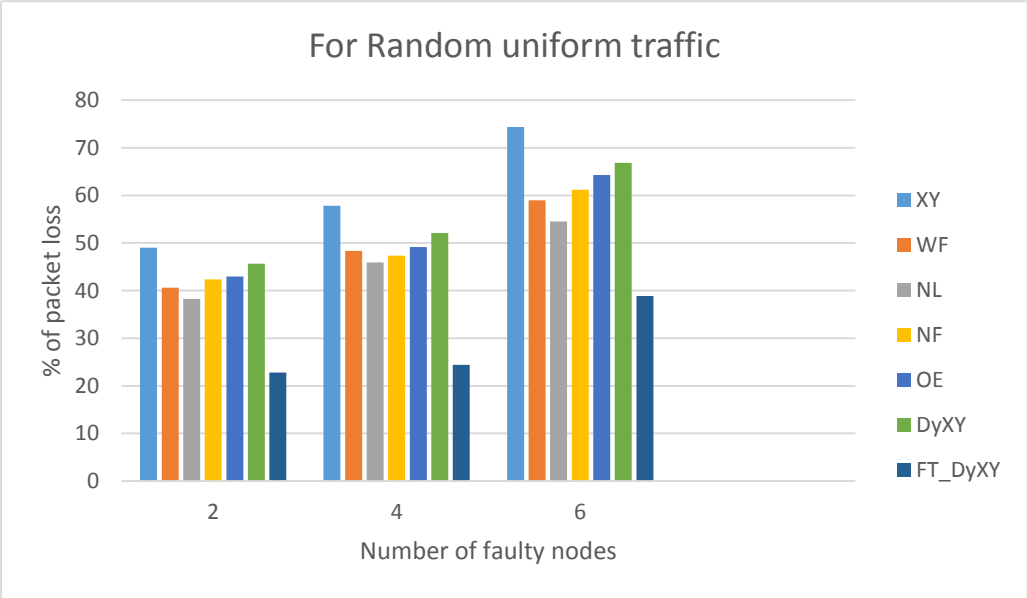


Figure 6.4 Number of faults vs % of Packet loss For Random uniform traffic pattern

FOR BIT REVERSAL TRAFFIC PATTERN

Table 6.3 Number of faults vs % of Packet loss for Bit reversal traffic

No. of Faults	*% of Packet Loss*						
	XY	***WF***	***NL***	***NF***	***OE***	***DyXY***	***FT_DyXY***
2	23.4	8.56	4.64	7.35	9.15	12.96	0
4	53.96	34.65	29.57	31.119	35.91	38.23	0.311
6	71.39	49.12	42.69	45.73	48.38	57.11	1.561

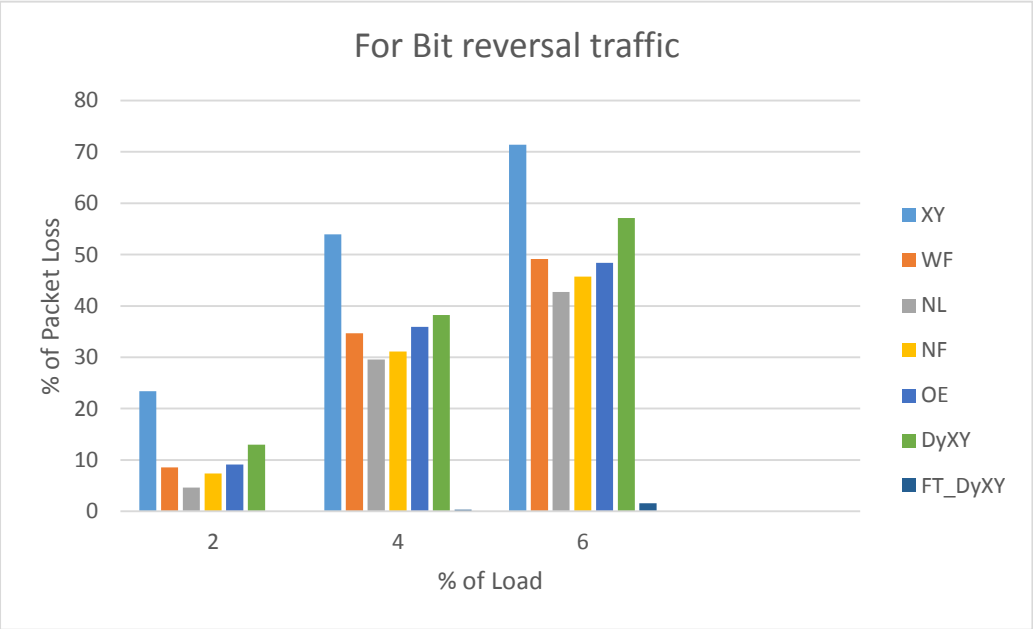


Figure 6.5 Number of Faults vs % of Packet loss for Bit reversal traffic

BIT SHUFFLE TRAFFIC PATTERN

Table 6.4 Number of faults vs % of Packet loss for Bit shuffle traffic

No. of Faults	*% of Packet loss*						
	XY	***WF***	***NL***	***NF***	***OE***	***DyXY***	***FT_DyXY***
2	18.65	3.41	3.12	4.63	3.98	10.69	0
4	22.91	11.67	9.11	13.15	12.61	17.38	0.152
6	41.19	22.36	20.43	21.16	22.11	30.15	1.075

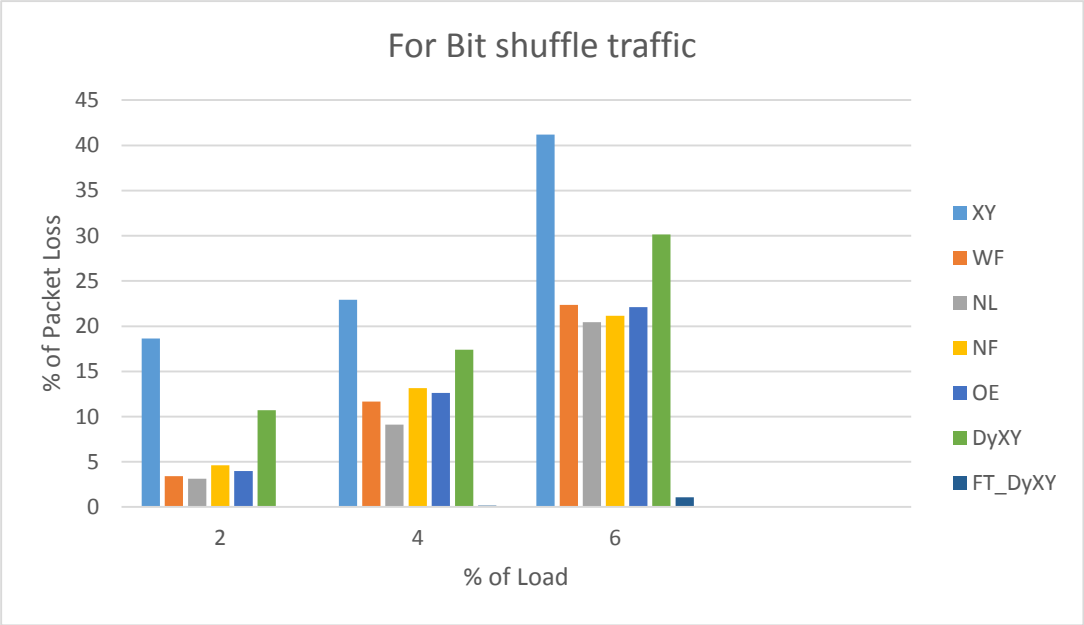


Figure 6.6 Number of Faults vs % of Packet loss for Bit shuffle traffic

6.5 Discussion

In NIRGAM simulator Number of faulty routers is varied and according to that the effect on performance parameter i.e. Packet drop (or % of Packet Loss) is observed for warm-up time 800 clock cycles and total simulation time 50000 clock cycles with 50% network load for three traffic patterns i.e. random uniform, bit reversal and bit shuffle. In this experiment Number of faulty nodes are varied from 2 to 6 and in Table 6.2 shows the simulation results of % of Packet Loss for XY, West First routing, North Last routing, Negative First routing, OE, DyXY routing and Fault tolerant DyXY routing for random uniform traffic pattern. Table 6.3 shows the simulation results of % of Packet Loss for XY, West First routing, North Last routing, Negative First routing, OE, DyXY routing and Fault tolerant DyXY routing for bit reversal traffic pattern. Table 6.4 shows the simulation results of % of Packet Loss for XY, West First routing, North Last routing, Negative First routing, OE, DyXY routing and Fault tolerant DyXY routing for bit shuffle traffic pattern. Figure 6.4, Figure 6.5 and Figure 6.6 shows the histograms of Table 6.2, Table 6.3 and Table 6.4 respectively.

According to the above results it is clear that XY routing algorithm will suffer from heavy performance degradation if number faults in the network will increase. Dynamic algorithms like DyXY, DyAD will also have performance degradation. But the turn model based algorithms have satisfactory performance in fault tolerance point of view. But North last routing algorithm has best fault tolerance capability. DyXY routing algorithm has best performance than discussed deadlock-free routing algorithms. If DyXY will be made fault tolerant, then it will definitely perform better in latency, throughput point of view and it will have less packet drop for all discussed traffic patterns.

Chapter 7

Conclusion and Future work

7.1 Conclusion

In this thesis we have studied about different routing algorithms like XY, OE, DyAD, West first, North last, Negative first, DyXY routing algorithms. The common capability of all these algorithms are their deadlock free nature. We also discussed about a fault tolerant routing algorithm which is based on one of the deadlock free routing algorithms i.e. DyXY. We compared their performance with varying % of network load and we observed the effect on the network parameters i.e. Average Latency, Average Throughput and Total Network power. After that we evaluated the performance of the fault tolerant routing algorithm based on DyXY routing by varying number of faults in the network and observed effect on packet drop (or % of packet loss). OE routing algorithm has less overall average latency than XY routing algorithm. In average throughput case we can see the complete dominance of OE over XY routing algorithm. But DyAD routing algorithm consistently outperforms XY and OE routing in latency and throughput case. But in total network power case OE has more power consumption than XY for all load conditions. So XY routing algorithm is more power efficient than OE routing algorithm. From Table III it is proved that DyAD routing algorithm is more power efficient than both XY and OE routing. The Performance metrics shows that OE routing algorithm is better routing algorithm than XY routing algorithm and DyAD routing algorithm is better than both XY and OE routing algorithm in performance aspects i.e. latency, throughput and total network power. DyXY performs better than other algorithms in random uniform, bit reversal and bit shuffle traffic pattern. Among partial adaptive routing algorithms i.e. west first, north last and negative first routing algorithms North Last algorithm shows best performance in latency, throughput and power point of view. Power efficiency of DyXY is more than OE routing. Power efficiency of routing algorithms is in the order shown i.e. Deterministic routing > Partial Adaptive routing > Adaptive routing. XY routing algorithm is the most power efficient algorithm but it suffers from performance degradation in heavy network load. XY routing algorithm will suffer from heavy performance degradation if number faults in the network will increase. Dynamic algorithms like DyXY, DyAD will also have performance degradation. But the turn model based algorithms have satisfactory performance in fault tolerance point of view. But North last routing algorithm has best fault tolerance capability. DyXY routing algorithm has best performance than discussed deadlock-free routing algorithms. If DyXY will be made fault tolerant, then it will definitely perform better in latency, throughput point of view and it will have less packet drop for all discussed traffic patterns.

7.2 Future works

The above conclusions are just fit for a 2-Dimension a 3x3 and 4x4 mesh topology NoC. These routing algorithms are only applicable to regular networks. We need to study different routing algorithms for irregular topologies. We need to study other routing algorithms which can also give better results for different parameter variation.

REFERENCES

[1] P. Gartz, “Regional Congestion Awareness for Load Balance in Networks-on-Chip,” in *14th International Symposium on High-Performance Computer Architecture*, Austin, 2006.

[2] G. Ascia, “Neighbors-on-Path: A New Selection Strategy for On-Chip Networks,” in *IEEE ESTIMedia*, Hong Kong, 2006.

[3] M. Palesi, “Application Specific Routing Algorithms for Network on Chip,” in *IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS*, March 2009.

[4] A. Mejia, “Region based Routing: A Mchanism to Support Efficient Routing Algorithms in NoCs,” in *IEEE TRANSACTIONS ON VLSI SYSTEMS* , 2009.

[5] M. Palesi, “Bandwidth-aware routing algorithms for networks-on-chip platforms,” in *IET Computers & Digital Techniques*, 2009,Vol. 3, Iss. 5, pp. 413–429 413.

[6] S. Rodrigo, “Efficient implementation of distributed routing algorithms for NoCs,” in *IET Computers & Digital Techniques*, 2008.

[7] R. Manevich, “Centralized Adaptive Routing for NoCs,” in *IEEE COMPUTER ARCHITECTURE LETTERS*, 2010.

[8] Jose´ Flich, “A Survey and Evaluation of Topology-Agnostic Deterministic Routing Algorithms,” in *IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS*, Chicago, March 2012.

[9] W. J. Dally, “Route packets; not wire,” in *DAC 2001*, Las vegas, 2001.

[10] S. P. a. N. Dutt, “Network on Chip,” in *On-Chip Communication Architectures*, Morgan Kaufmann, 2010, p. 439.

[11] J. K. Singh ,A.K. Swain “Performance evalulation of different routing algorithms in Network on Chip,” in *PrimeAsia 2013*, Vishakhpatnma, 2013.

[12] T. L. Ville Rantala, “Network on Chip Routing Algorithms,” citeseerx, 2006.

[13] M. A. Yogita A. Sadawarte, “Comparative Study of Switching Techniques for Network-on-Chip Architecture,” in *ICCCS’11*, Rourkela, 2011.

[14] Z. Lu, “Using Wormhole Switching for Networks on Chip: Feasibility Analysis and Microarchitecture Adaptation,” Royal Institute of Technology, Sweden, 2005.

[15] A. Jantsch, Networks on Chip, London: Kluwer Academic Publishers, 2003.

[16] S. Y. Jose Duato, Interconnection Networks: An engineering approach, New York: Morgan Kaufmann Publishers, 2010.

[17] D. L. selvakumar, “Analysis & Design of Network in Reusable Sub-Systems,” *International Journal of Engineering and Technology,* vol. 1, no. 2, pp. 50-62, 2009.

[18] L.-S. P. Natalie Enright Jerger, On-Chip Networks, Madison: Morgan & Claypool Publishers, 2009.

[19] William James Dally, Principles and Practices of Interconnection Networks, San fransisco: Morgan Kaufmann, 2004.

[20] P. Y. K. C. Terrence Mak, “Adaptive Routing in Network-on-Chips Using a Dynamic-Programming Network,” *IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS,* vol. 58, no. 8, pp. 3701-3716, 2011.

[21] U. o. S. UK, “NIRGAM: A Simulator for NoC Interconnect Routing and Application Modeling,” 2010. [Online]. Available: nirgam.ecs.soton.ac.uk.

[22] L. H. J. W. Wang Zhang, “Comparison Research between XY and Odd-Even Routing Algorithm of a 2-Dimension 3X3 Mesh Topology Network-on-Chip,” in *Global Congress on Intelligent Systems*, Hong Kong, 2009.

[23] G.-M. Chiu, “The Odd-Even Turn Model for Adaptive Routing,” *IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS,* vol. 11, no. 7, pp. 729-738, 2000.

[24] J. Hu, “DYAD - Smart Routing for Networks-on-Chip,” in *DAC 2004*, California, 2004.

[25] L. M. N. Christopher J. Glass, “The Turn Model for Adaptive Routing,” ACM digital library, East Lansing, 2005.

[26] Rohini, “Design and Implementation of Minimal adaptive West first algorithm for NoC Router Architecture,” in *International Journal of Current Engineering and Technology*, Dharwad, 2013.

[27] M. Li, “DyXY - a proximity congestion-aware deadlock-free dynamic routing method for network on chip,” in *Design Automation Conference, 2006*, San Francisco, 2006.

[28] M. A. G. Shubhangi D Chawade, “Review of XY Routing Algorithm for Network-On-Chip Architecture,” in *IRNet Transactions on Electrical and Electronics Engineering*, Jaipur, 2009.

[29] J. Wu, “A Fault-Tolerant and Deadlock-Free Routing Protocol in 2D Meshes Based on Odd-Even Turn Model,” *IEEE TRANSACTIONS ON COMPUTERS,* vol. 52, no. 9, pp. 1154-1169, 2003.

[30] M. F. Yusuke Fukushima, “Fault-Tolerant Routing Algorithm for Network on Chip without Virtual Channels,” in *IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems*, Tohuko, 2009.

PUBLICATIONS

[1] **J K Singh**, A.K. Swain, T.N.K. Reddy and K. K. Mahapatra, "Performance evalulation of different routing algorithms in Network on Chip," IEEE PrimeAsia Proc., pp. 180–185, Dec. 2013.

[2] T.N.K. Reddy, A.K. Swain, **J.K. Singh** and K.K. Mahapatra, "Performance assessment of different Network on chip topologies" IEEE ICDCS Proc., Mar. 2014.