# ABSTRACT

The performance of the system is heavily coupled by the efficiency of underlying communication and processing architecture. Growing need of high performance computing and decreasing size of technology has enhanced the growth of multicore era. They are more focused on communication then computation. And urge to design and develop an efficientcommunication architecture has forked a new dimension in computer architecture i.e. Network on chip (NoC).

Bus-based system involves arbitration delay. Crossbar involves scalability issue and point to point increases in complexity in routing wires.Hence we connect the cores through network on th Chip. However, NoCs have been proved to be reliable, scalable and exible but still there are a lot of challenges related to fault tolerance, quality of service, power and area that need to be addressed.In this report, adaptive routing algorithms which have been developed to improved the performence of NoC in terms of fault tolerance and congestion awareness are explained.These algorithms are cost-effective and efficient.

Firstly, we propose a cost-effective fault tolerant routing algorithm for irregular 2D mesh without using tables. We exploit one hop visibility of LBDR to eliminate tables. This algorithm handles one or many single link faults within 2D mesh and uses reconfigured paths (minimal and/or non-minimal), if links fail. We use turn model based approach to avoid deadlock. Since our method does not need virtual channels to achieve deadlock freedom, it is more area and power effective.

Then, we present low-cost, congestion-aware, non-minimal and fully adaptive (CHARM) routing algorithm for 2D and 3D meshes that offers high degree of adaptiveness by permitting cycles in channel dependency graph while remaining deadlock free. CHARM uses only one, two and two virtual channels along the X, Y and Z dimensions, respectively, thus becomes low-cost algorithm. It is a a reconfigurable, deadlock free and cost-efficientfault tolerant routing algorithm without using VCs.

To overcome limitation of dimension order routing in spidergon, an adaptive routing algorithm has been proposed which allows the packet to take across link at any place according to current congestion level. Our scheme is minimal path based and packet take across link only once if required. If a packet want to take across link and that link is congested, then it can take left or right link and go for across link at any other intermediate node.

# CHAPTER 1

# INTRODUCTION TO NETWORK ON CHIPS

## 1.1 Motivation

Advances in Very Large Scale Integration (VLSI) fabrication technology allow complex chips designs containing billions of transistors on a single chip. As the component size shrinks, more space is available on the chip. Available chip space increases the scale of integration of electronic devices on a single chip. A complete system with a very large number of components including analog devices can be implemented on a single chip, i.e., System-on-Chip (SoC).

Technology advancement and shrinking size of transistor has increased the number of compo-nents on chip. As per Moore's law, the number of transistors on integrated circuits doubles approximately every 18 months. There is millions of transistor available on chip. With growing complexity, design paradigm is shifting toward multiple cores on single chip instead of a single complex processing element. Therefore focus has been shifted from computation to communica-tion [1] since area, performance and power consumption of SoC mainly depends on underlying interconnect architecture.

For years, systems have been designed on bus architecture. Bus is used as a shared medium of communication as shown in Figure 1.1. But that architecture has problems such as

- Bandwidth of the bus.
- Arbitration logic has multiple levels of complex logic to guarantee some fairness in bus allocation.
- Wire delays and capacitance cause physical problems.

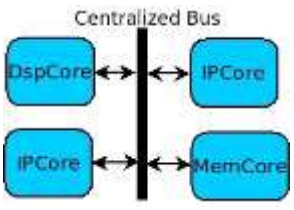


Figure 1.1: Bus Based Communication

In multi-core architectures, bus-based systems scale only to a small number of processors. This limited scalability is because bus traffic quickly reaches saturation as more cores are added to the bus, so it is hard to attain high bandwidth [2]. The power required to drive a long bus with many cores arbitrating onto it grows exponentially. In addition, a centralized arbiter adds arbitration latency as core counts increase requiring a large area and consuming high power. To address these problems, bus designs includes segmentation, distributed arbitration, split transactions.

With a small number of nodes, dedicated point to point as shown in Figure 1.2 wiring can be used to interconnect them. However, the use of dedicated wires is problematic as we increase the number of components on-chip. The amount of wiring required to directly connect every compo-nent will become prohibitive. The complexity of design increases as more e ort on placement and routing of wires. More clock cycles are wasted in driving the wires instead of logic gates.Hence complexity and cost of communication grows as the number of components increases.

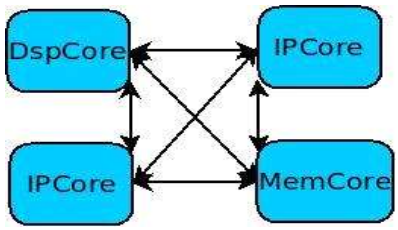


Figure 1.2: Point to Point Communication

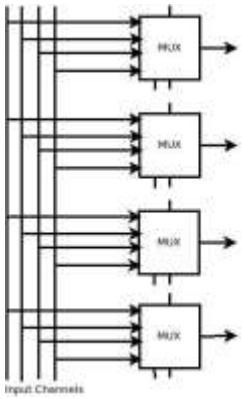

Figure 1.3: 4 X Crossbar

Crossbars as shown in Figure 1.3 address the bandwidth problem of buses, and complexity issue of point to point interconnection, and have been used for on-chip interconnects for a small number of nodes. They are non-blocking but circuit switching based network and are mostly unicasting. Hierarchical crossbars, where cores are clustered into nodes and several levels of smaller crossbars provide the interconnection,are used as alternative to buses and crossbars. However, crossbars scale poorly for a large number of cores. It require large area as number of cores grow and consume high power.

Network on Chip as shown in Figure 1.4 represent a scalable solution to on-chip communica-tion [3], due to their ability to supply scalable bandwidth at low area and power overheads. Designs with small number of cores using buses and crossbars are considered the simplest vari-ants of networks-on-chip. Second, on-chip networks are very efficientin their use of wiring, multiplexing. NoC allows different communication ows on the same links allowing for high bandwidth. Finally, on-chip networks with regular topologies have local, small interconnects that are fixed in length. Network on Chip, a network based approach to interconnect all com-ponents of SoC [4]has been proposed as an alternative to overcome drawbacks of bus and point to point based classical interconnects.

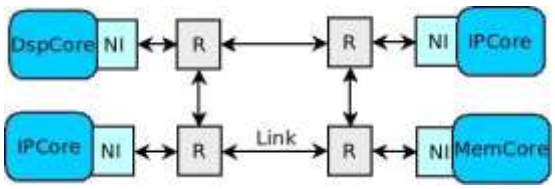


Figure 1.4: Network On Chip

## 1.2 NoC Basics

### 1.2.1 Topology

Network-on-chip comprises of routers, links and network Interface (NI) as shown in Figure 1.4. Router are are connected together by links. NI is used to interface router to ip-cores. Multiple arrangements of the routers have been proposed in academia and industry.

Any arrangement of the routers is known as the Topology. The routers are shared resource for any interconnect network. Selection of topology depends on things like the packaging technology, bandwidth requirement and latency. There is no \Perfect" topology. Different topology is chosen under different constraints and requirements. There are multiple topologies possible for NoC such as Mesh, Cube (Hypercube), Tori, K-ary N-cube and other topologies. A overall classi cation of topology is described in Figure 1.5

The topology that exhibits geometrical property, their interconnection is defined by mathematical foundations are called regular topology. The mathematical foundations paves way for sound and efficinetperformance of the network. They are easier to analyze and implement. Topology that are formed by arranging as per particular need are called irregular. Application specific NoC arrange router as per proximity and need of application and hence are irregular topologies. They exploit the locality of application to improve the performance. Arrangements can be both power and area efficient.

In regular topologies when each router is attached with ip-cores then they are called direct topology. However when the NoC comprises of router which only act as intermediate nodes in facilitating communication are called indirect topologies. Fat-tree, Butter y are some indirect topologies used in NoC design. In direct topology, a router act as terminal and intermediate both. It can act as source and sink. The resources are utilized more effectively. When all router of a regular topology have same number of input and output ports then it is called uniform topology. Spidergon, torus, HPC Mesh are some examples of uniform topology. While a network containing non-uniform router is called non-uniform topology. Mesh comprises router with two port(corner routers), three port (border router) and four port routers hence is a non-uniform topology.

#### 1.2.1.1 Parameters of a Topology.

In order to compare different networks, we need some parameters for the topologies so that we can compare their relative performance. Some of the most widely used network topology parameters [2]are the following.

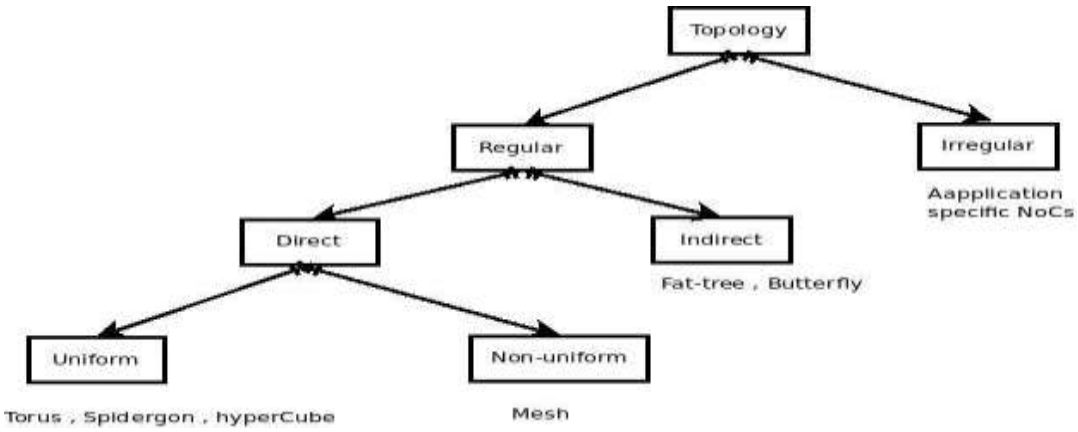


Figure 1.5: Topology in Network On Chip

**Routing Distance** : This is effectively the number of links that any packet needs to traverse in order to go from source to destination. It is neither best-case nor the worst-case link count but the number of links for any pair of nodes.

**Diameter of the Network** : Diameter of the network is defined as the maximum routing distance between any two points. This is an important parameter because we always want to minimize the maximum routing distance. For the mesh topology, the diameter is

$$2\sqrt{N} - 2$$

where N is the number of nodes in the mesh.

**Average Distance** : Ava rage distance is the average routing distance between each pair of nodes. We compute the total routing distance between each pair and divide it by the number of pairs.

**Minimum Bisection Bandwidth** : Minimum bisection bandwidth is defined as the bandwidth of a minimal cut though the network such that the network is divided into two equal sets of nodes. For mesh topology, minimum bisection bandwidth is $2\sqrt{N}$ where N is the number of nodes in the mesh. Here we assume that the links are bidirectional.

**Path Diversity** : Path diversity is property of topology which explains the presence of multiple shortest path for a pair of source and destination. It means multiple shortest path are available for traveling from a source to destination.

### 1.2.2 Routing

Routing logic decides which direction the packet should take in order to reach the destination. Each packet must contain enough information required to reach the destination. Depending on the design decisions, packets routing information is put on the packet. Minimum information needed by the routing logic to decide the next direction is the destination address in which case we do not worry about the sender and no response is to be sent. Routing algorithms are very critical for several reasons. Good topology selection gives us potential performance possibilities but good routing algorithm can help us achieve that potential. Good routing algorithms balances the networks load across the network even if there is a non-uniform traffic generation. They are broadly classifed into three different groups based on basis destination, routing decisions and adaptability as shown in Figure 1.6.

- **Destinations**

  - **Unicast routing** : This is based on one to one communication and message is sent to only one destination at a time. Only one copy of packet is in the network any time. Resource utilization is optimum.

  - **Multicast routing** : It broadcast to all the nodes attached and allowed, and message is sent to all the nodes. Multiple copy of packet is created in network. It creates large traffic overhead ad performance is affected.

- **Routing Decisions**

  - **Centralized routing** : Routing is done at the centralized and dedicated node and distributed to rest of the nodes in the network.Each node store path to other nodes. As centralized node need to distribute routing paths to other nodes, it increases the extra overhead in traffic and memory requirements.

- **Source routing** : In this method route is determined at the source itself and attached to the message header. All along the path header information is used to traverse the packet.

- **Distributed routing** : Routing is determined at all the hops through which it is traveling. It can take care of network congestion state without much overhead.

- **Adaptability**

  - **Deterministic and Oblivious Routing** :These algorithms always choose the same path for a given source-destination pair even if there are multiple paths existing be-tween the pair. They do not consider the network state while computing the path, so they don`t perform load balancing. But because of the advantages of ease of imple-mentation and deadlock free behavior, they are quite common in practice. Oblivious routing too does not depend on the state of the network. These algorithms include deterministic algorithm as subset, it does not take the state of the network state in consideration.

  - **Adaptive Routing** : As the name suggests, Adaptive routings consider the present network state and then by looking at the traffic and hotspots, try to route the packets around the congestion spots. The network state variables that could be considered are node state, link state, buffer queue lengths etc. The routing path which is generated is different based on the current network status. This help to nd the best path possible in case of high congestion. Further adaptive routing is divided into two parts. Partially adaptive algorithm follows some (partial) path which corresponds to shortest path. Fully adaptive algorithm can route packets in all paths which are shortest route.

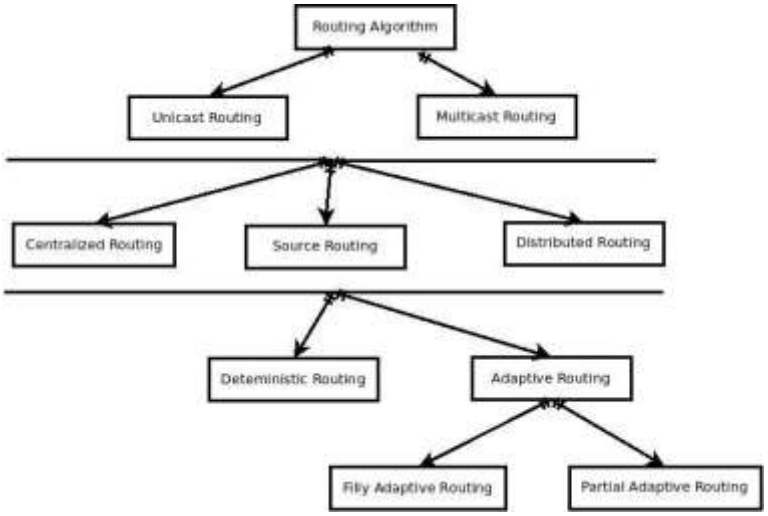


Figure 1.6: Routing in Network On Chip

.

### 1.2.3 Switching Techniques for NoC.

It determines how network resources are allocated for data transmission, i.e. how and when the input channel is connected to the output channel selected by the routing algorithm.Switching is broadly classifed as shown in Figure 1.7 :

- **Circuit switching** :The communication between a source and destination has two phases : circuit establishment phase and message transmission phase. A physical path from the source to the destination is reserved prior to the transmission of data by injecting a routing probe, which contains destination address and some control information.

- **Packet Switching** :It consists of protocols in which messages are divided into packets before they are sent. Each packet is then transmitted individually and can even follow different routes to its destination. Packet switching is then abstracted into three parts.

  - **Store and Forward** : Each mode receives a complete packet and then sends it out. It is based on the assumption that a packet must be received in its totally before any routing decision can be made and the packet start sending to the destination.

- **Virtual cut through** : Each node starts to send the packet without waiting for whole packet to arrive. In fact, the message does not even have to be buffered at the output and can cut through to the input of the next router before the complete packet has been received at the current router. This switching technique is referred to as vir-tual cut-through switching (VCT). In the absence of blocking, the latency experienced by the header at each node is the routing latency and propagation delay through the router and along the physical channels. The message is effectively pipelined through successive switches. If the header is blocked on a busy output channel, the complete message is buffered at the node. Thus, at high network loads, VCT switching behaves like packet switching [5].

- **Wormhole switching** : The packets are split to its which are snaked along the route exactly in the pipeline and also here transmission of different packets cannot be interleaved or multiplefixed freely over one physical channel without additional architectural support. Every router has small buffers for one or a few its per channel. The primary di erence between wormhole switching and VCT switching is that, the unit of ow control is a single it and, so small buffers can be used. In case of blocking, the small buffer sizes at each node cause the message to occupy buffers in multiple routers, recessively blocking other messages. This nature of wormhole complicates the issue of deadlock freedom.

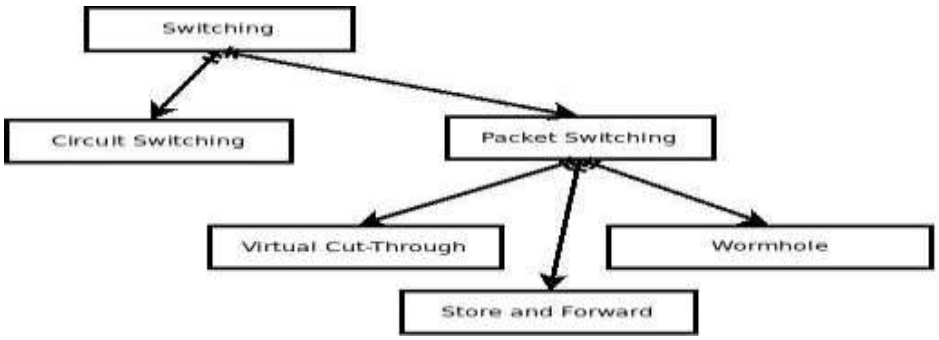


Figure 1.7: Switching In Network On Chip

### 1.2.4 Flow Control Methods for NoC

Various resources in network-on-chip are channels, buffers, switch and control states. Flow control can be viewed as problem of resource allocation or contention resolution. Flow control de nes synchronization and transmission protocol for a unit of information. The unit of ow control refers to that portion of the message whose transfer must be synchronized. This unit is defined as the smallest unit of information whose transfer is requested by the sender and acknowledged by the receiver. The request/acknowledgment signaling is used to ensure successful transfer and the availability of buffer space at the receiver.

Flow control is responsible for regulating the traffic in a certain way that maximizes the band-width. In Network-on-chip, Flow control makes sure that network does not drop data as retrans-mission is not a obvious choice.

- **Buffer less flow Control**
  - Circuit switching

- **Buffered flow control**
  - Packet Buffer
  - Flit Buffer
    - Link Based
      - ✓ Credit Based Back-Pressure
      - ✓ Stop and Go
      - ✓ Request Acknowledgement based
    - End-to-end
      - ✓ Explicit backward Notification

Flow control can be either link based in which we regulate the traffic flowing through each link. These mechanism depend on information coming from neighboring nodes only. While end-to-end ow control involves control the round trip ow control from source to destination ensuring that we do not ood the network. Information from each node is

forwarded implicitly or explicitly to source or destination. It is also responsible for allocation of resources in optimum way to achieve the highest performance fraction of the ideal performance.

## 1.3 Layered Architecture of NoC

NoC can be described as abstraction of layer as OSI layers describe the functioning of net-work.The network is the abstraction of the communication among components and must satisfy quality-of-service requirements such as reliability, performance, and power consumption. Above that the limitation of unreliable signal transmission and communication delays on wires raises few more issue. These layer de ne architecture and functionality of the NoC. Each layer have dif-ferent functionality and handles property of different component. The layer are basically formed on the basis of the component and their function.Three layer of abstraction in NoC are Software Layer, Architecture and Control Layer and Physical Layer as shown in Figure 1.8 [4].

- **SOFTWARE LAYER :**

This layer deal with top most abstraction i.e. System Applica-tion level. At this level,most of the network implementation details are still hidden. The data is represented as message. The details included in this layer are source and destina-tion of the messages, injection rate, application generating data and hardware on which application is running. This layer also differentiate between memory access data and com-putational data. This layer have knowledge of functionality and property of ip-cores,DSP cores,Memory,ASCI etc.This layer is the top most layer of the NoC.There are follflowing function that will be done at this layer :

- **Design methodology and abstraction :**

Design methodology de nes how comfortably system level NoC characteristic can be changed at instantiation time. There are wide range of parameters such as number of slots in the switch, pipeline stages in the links, number of ports of the network, and others. This is very useful for co-exploration on the basis of architectural parameter a ecting directly with ip-cores of the SoC.

The aim of modeling is to understand clearly required architecture of NoC and analyze trade-o s relations among power, area, design-time etc; while adhering to application requirements on one side and technology constraints on the other side. Modeling NoC has three eld modeling environment, abstraction levels, and result analysis.

Application mapping is important aspect of this layer. For assisting application map-ping, it includes allocator, scheduler and synchronizer. Allocator provides the re-sources needed and in turn to minimize the resource con icts for bandwidth, buffer, channel etc in the network. The scheduler optimize the resource utilization by allo-cating threads to cores such that minimum resource is used in network. Synchronizer deals with concurrency between the messages which are communicating. These system are part of operating system if we have single core.

- **Architecture Domain:**

The most important architecture parameter are topology, routing algorithm, flow control techniques, switching techniques, buffering scheme, switch architecture, arbitration techniques etc. These parameter decides the perfor-mance of the NoC. We apply various design principles such as clustering and com-position to enhance their performance. Dealing with run time fault comes under the handling run time network condition such as fault and congestion.

- **Traffic characterization :**

(A)latency-critical : Latency critical traffic is traffic with strict latency demands for critical interrupts, memory access, etc. These often have small payload. (B)Data streams : It have high payload and demand QoS in terms of bandwidth. Usually it is large fixed bandwidth, which may be jitter critical e.g. MPEG data, DMA access,etc.

- **ARCHITECTURE AND CONTROL LAYER :**

This is also architectural layer of the NoC,where this is similar to the network layer,transport layer and data link layer of OSI model. This is a layer where network interface and router works. As named interface act bride between router and ip-cores. It assembles its into packets and dissemble packets into its. Router route the packets through right channel reach its destination.

- **Network Interface :**

It act as intermediate of the in network to the cores on which application are running. Generally work at transport layer. Important functions are encapsulation, service management. This part has a socket which connect the ports. We can reuse the ip-cores. Here message are divided into packets,and its encapsulated and send to the transport layer by attaching a transport header.

- **Router:**

It has two main function namely routing and selection of optimum path. Routing function provides the set of output channels at particular node basis on the current node, destination node and congestion status where selection function select the channel from that set on basis of the availability of the channel, is it free or busy.

- **PHYSICAL LAYER :**

This is same as OSI physical layer which includes the links and ports, links as physical medium to connect ports and sends the message to the destination. Links connects the output channel from the nodes to input channels of next node. Each channel has address space , these is called the port which is some number to assign that link to the message. At this layer synchronization, encoding and reliable transfer trough medium are prime functionality.

## 1.4 Industrial NoC Systems

Network on chip is a new research area. Its use in commercial processor is still in beginning phase. Hence only chip have been designed with network on chip based system. Some of the chips are :

**IBM Cell** : The Cell architecture is a joint e ort between IBM, Sony and Toshiba to design a power-efficinetfamily of chips targeting game systems, but that are general enough for other domains as well. Cell is a product that is in most game consoles in the market today. It is a 90nm, 221 sqr. mm chip that can run at frequencies above 4GHz. It consists of one IBM 64

bit PowerPC Architecture core and eight Synergistic Processing Elements (SPEs) that are single instruction-multiple-data (SIMD)-based. These nine nodes are interconnected with an on-chip Element Interconnect Bus (EIB) that overlays bus access semantics on four ring networks, with a maximum bisection bandwidth of over 300GBytes/s7.

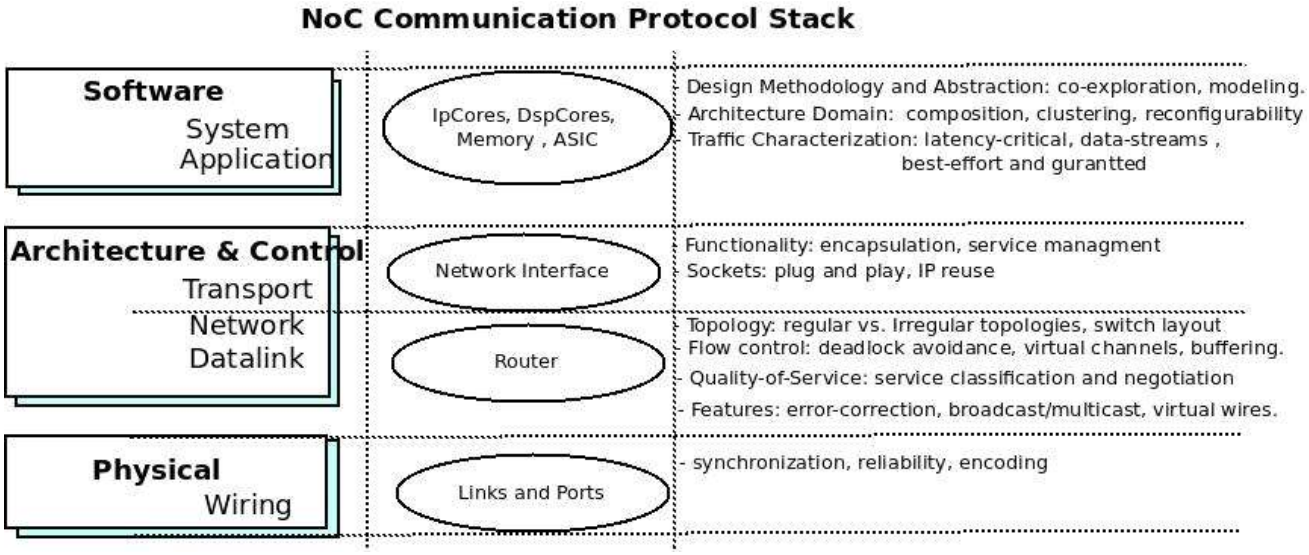


Figure 1.8: Communication Protocol Stack

**Intel TeraFLOPS** :

The TeraFLOPS processor is a research prototype chip that is targeted at exploring future processor designs with high core counts. It is a 65nm, 275mm2 chip with 80 tiles running at a targeted frequency of 5GHz. Each tile has a processing engine (PE) connected to an on-chip network router. The PE consists of two single-precision oating-point multiply-accumulator (FPMAC) units; 3KB of single-cycle instruction memory (IMEM), and 2KB of data memory (DMEM). The router interface block (RIB) interfaces the PE with the router, and any PE can send or receive instruction and data packets to or from any other PE.Two FPMACs in each PE providing 20 GigaFLOPS of aggregate performance, partnered with a maximum bisection bandwidth of 320GBytes/s in the on-chip network enable the chip to realize a sustained performance of 1012 oating-point operations per second (1.0 TeraFLOPS) while dissipating less than 100W.

**TileraTILE64 andTILE64Pro** :

TheTILE64 andTILE64Pro architectures are products from Tilera targeted towards high-performance embedded applications such as networking and real-time video processing.

These chips support a shared memory space across the 64 tiles (TILE64Pro has additional support for cache coherence termed Dynamic Distributed Cache), with each tile consisting of L1 and L2 caches and a 3-issue VLIW processor core connected to the four mesh networks. The TILE64 chip at 90nm, 750MHz, has 5MB of on-chip cache and on-chip net-works that provide a maximum bisection bandwidth of 2Tb/s with each tile dissipating less than 300mW.

**ST Microelectronics STNoC** : The STNoC is a prototype architecture and methodology targeted to replace the widely-used STBus in MPSoCs. It is targeted towards the unique demands of MPSoCs on the on-chip network fabric : automated synthesis of network design, compatibility with heterogeneous, diverse IP blocks from other vendors and prior knowledge of traffic demands. It has been applied to the Morpheus chip targeting 90nm, with eight nodes on the on-chip network connecting an ARM9 core, an array of ALUs, a reconfigurable array, embedded FPGA,and memory controllers.

# CHAPTER 2

# Related Work

## 2.1 Turn Model and Deadlocks in Routing Algorithms [6]

Routing algorithms are very important component of router. It decides the path a packet is going to take in coarse of its journey. Working of routing algoritham a cts the overall performance of NoC. Any optimum routing algorithm shows following property.

- DeadLock Freedom.
- LiveLock Free.
- No Starvation.
- Uniform Load Balancing.

Average communication latency is highly dependent on the chosen routing technique. Deadlock can be one of the major challenges for any routing technique which inde nitely postpones the delivery of a packet. Most of the fault tolerant routings use either turn models [7, 8, 9, 10, 11] or virtual channels (VCs) [12, 13, 14] based strategies to achieve deadlock freedom. However, attaching VCs to a physical channel is less expensive than adding one full physical channel, but this solution is still expensive, particularly in low area and power NoCs. The main area costs associated with VCs are the extrafficontrol lines to implement VCs and additional buffer space required for each VC. additional area overhead is due to switching and control hardware for multiplexing and transaction protocols. In addition to these area related costs, adding VCs also increases communication latency of a router [15, 16]. Thus, in NoCs where it is required to keep area and latency low, VC based approaches are not suitable candidates for low cost constraints.

Many deadlock free routing algorithms has been proposed in the literature that do not require any VCs for deadlock avoidance. XY routing algorithm for 2D mesh topology first forwards the packet along x direction and then along y direction. It restricts turns from y

direction to x direction as shown in Fig. 2.1a. The turn models are introduced by Glass and Ni [7]. They published three elegant partially adaptive routing techniques namely west- first, north-last and negative- first for n-dimensional meshes.

These routing algorithms provide deadlock freedom by prohibiting one turn in each abstract cycle (clockwise and counter-clockwise) as shown in Fig. 2.1. Chiu [9] proposed one new Odd-Even (OE) turn model which has higher degree of adaptiveness than Glass's turn model. In this model, ES (East to South) & EN turns are not allowed on the nodes located at even columns, and NW & SW turns are not allowed at nodes located at odd columns. This model is basically combination of east-last restrictions at even columns and west- rst restrictions at odd columns .

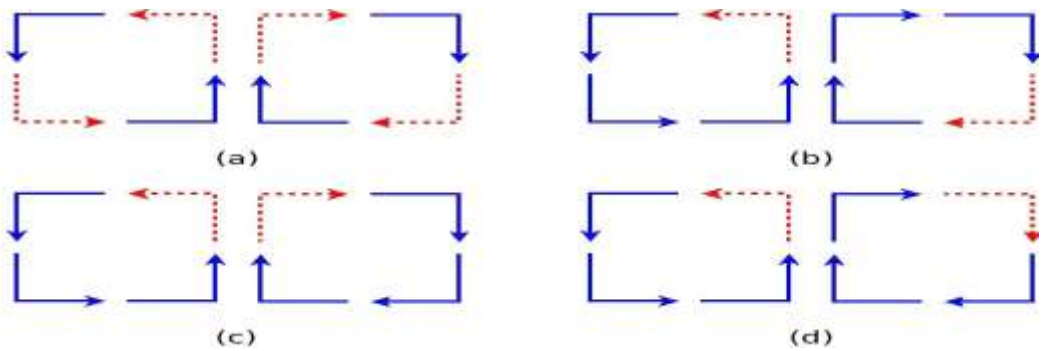


Figure 2.1: Turn Models (a) xy (b) west- first (c) north-last (d) negative- first (solid lines for permitted turns and dash lines for restricted turns)

Hu and Marculescu [17] proposed a routing algorithm called DyAD, that combines low latency advantage of deterministic routing at low loads and high throughput advantage of adaptive routing. When the network is not congested, the DyAD router is operated under deterministic mode. It switches to adaptive mode as the network becomes congested.

Turn model produces partially adaptive routing algorithms. Another popular method of design-ing deadlock free routing algorithms with higher adaptiveness, is to add virtual channels to the network. Several minimal/non-minimal and fully adaptive routing schemes, based on a small number of virtual channels, have been proposed in [12, 18, 14, 19, 20, 21, 22, 23, 24, 25].

## 2.2 Fault-tolerant Routing Algorithms for 2D Mesh [6]

On-chip networks have been aggressively discussed and researched in recent past as a dominant communication paradigm for complex Multi Processor System on Chips (MPSoCs) and Chip Multi Processors (CMPs) because of their scalability, energy-efficiency, reusability and reliabil-ity [4]. In an MPSoC or CMP architecture, reliability of NoC is affected by transient and/or permanent faults. Faults a ect the functionality and can degrade the performance of on-chip networks. Thus, it has become essential to design on-chip networks which can tolerate faults. In an MPSoC or CMP architecture, reliability is affected by transient faults (alpha particles, cosmic rays, coupling noise and crosstalk) and permanent faults (defective open or short cir-cuits, electro-migration), produced by the low yield and process variations that are likely in nano-technology or deep sub-micron processes.

Faults can be tolerated by using many methods and majority of them are based on fault tolerant routing algorithms. Routing algorithms can be implemented as either table-based or algorithmic routing [26, 27]. In algorithmic routing mechanism, an algorithm is executed using software or special hardware circuits (generally combinational logic) employing FSM to compute appropriate router port. It is generally suitable for one topology and one routing method for that topology. The algorithmic mechanism is often efficinetin the terms of speed and area than table-based routing [26]. Table-based mechanism is used to deal with regular as well as irregular topologies. In this mechanism, each router keeps a table to store one output port channel (deterministic routing) for each destination from the current node in the network. The bene t of this mechanism is that it supports any topology and any routing method, even fault tolerant and congestion aware routing algorithms. The table based methods cannot scale well since the table size increases with size of the network and may become impractical for large NoCs.

In the application specific platforms where communication transactions (concurrent/non-concurrent) among IP cores are known in advance, it is quite possible to use compression techniques [28] to reduce size of tables instead of straight forward table-based implementation. However, it is not a generic scenario in multi core systems.

In [29], authors proposed efficinetimplementation of distributed routing algorithms for partial 2D meshes (irregular meshes) without routing tables. In this strategy, they suppress the need of tables on every router and enable the distributed implementation of

any routing algorithm. This method is known as LBDR. It uses three bits at each output port of a router and a few logic gates.

In [8], a fault-tolerant routing technique is introduced that can tolerate only single-node failures in mesh-based NoCs. It cannot handle link failures. Boppana and Chalasani [30] proposed a deterministic fault tolerant routing algorithm that can tolerate faulty nodes and links. It uses additional VCs to avoid deadlocks, thus, results in area overhead. Masoumeh Ebrahimi et al. [25] presented a minimal and defect-resilient routing method to route packets in the presence of one faulty link using two VCs. David Fick et al. [31] proposed a table based implementation of routing algorithm to handle faulty components.

## 2.3 Congestion-Aware Routing Algorithms in 2D mesh and 3D mesh

Congestion control is a critical issue in on chip networks as it a ects overall network performance. Congestion can be alleviated with balanced traffic load across various links. Adaptiveness and non-minimality of a routing algorithm is used to balance traffic load while maintaining deadlock freedom. Turn model produces partially adaptive routing [7, 9, 17, 8, 10] schemes that restrict some routing turns to achieve deadlock freedom thus resulting in low adaptiveness. Deadlock free fully adaptive routing schemes require additional virtual channels to achieve high degree of adap-tiveness. Several minimal/non-minimal fully adaptive routing algorithms have been proposed in [12, 18, 14, 19, 20, 21, 22, 23, 24] using additional virtual channels.

A routing algorithm determines output channel for a packet to reach its destination. A major challenge in designing any routing algorithm is deadlock-freedom. Deadlock is an anomalous network state in which network resources form a circular hold-and-wait dependency relation that inde nitely postpones the delivery of a packet.

The routing schemes introduced in [18, 12] produce an equivalent fully adaptive and minimal routing algorithm called double-y for 2D mesh. It requires one and two virtual channels in X and Y dimensions, respectively. In [14], authors introduced maximally adaptive double-y routing algorithm (Mad-y) which is an improvement over double-y network based schemes [12, 18] and it allows more number of routing turns by making better utilization of virtual channels to increase adaptivity. Ming Li et al. [19] proposed congestion-aware and dynamic routing algorithm DyXY that determines output channel using congestion status of input buffers of next hop routers. CATRA [23], DAR [21], RCA [20] and DBAR

[22] are congestion-aware routing schemes that use local and non-local congestion information to route the packets using extra hardware.

Minimal routing algorithms suffer from low degree of adaptiveness, which is not su cient in dis-tributing the network traffic, even if they accurately detect the status of congestion. Ebrahimi et al. [24] proposed non-minimal routing scheme for 2D mesh. It provides better adaptiveness than [14] with same number of virtual channels. However, it imposes some unnecessary restric-tions on routing turns, which can be removed to increase path diversity.

Routing method presented in [32] uses four, four, and two virtual channels along the X, Y and Z dimensions, respectively for deadlock avoidance. Dahir et al. [33] presented partially adaptive routing method which is 3D extension of 2D odd-even turn model [9]. Ebrahimi [34] proposed a fully adaptive routing algorithm for 3D NoCs that uses two, two and four virtual channels along the X, Y and Z dimensions, respectively. Deadlock avoidance method adopted in well-known planar adaptive routing method [18] uses one, three and two virtual channels along the X, Y and Z dimensions, respectively. It is fully adaptive within a 2D plane only.

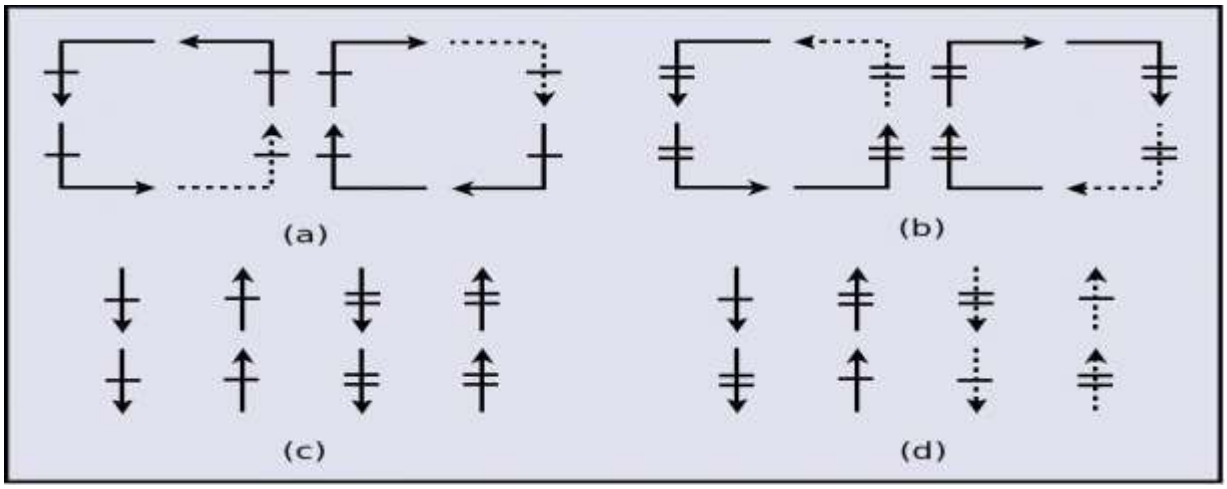


Figure 2.2: Mad-y

## 2.4 Routing Algorithms on Spidergon

Routing algorithms proposed so far for spidergon are very simple since they take the advan-tage of symmetry and simplicity o ered by spidergon topology. To the best of our knowledge, acrossFirst (aFirst), acrossLast (aLast) [35], aEqualized [36] and dynamic stress

de ection rout-ing (DSWR) [37] are the proposed algorithms for this architecture. aFirst, aLast and aEqualized are deterministic, minimal and source-based routing. DSWR is table based fault-tolerant adap-tive routing.

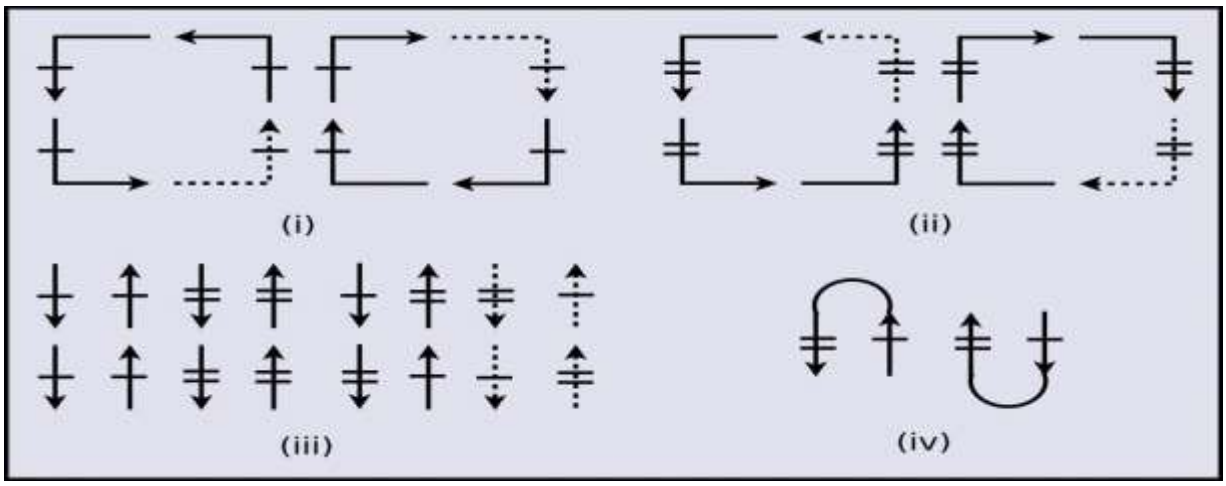


Figure 2.3: LEAR

In aFirst routing, if across link is to be traversed then it is taken as first hop and after that packet moves on left or right links. aLast traverse left or right link first and if across link is need to be traversed then it is taken as last hop. Across link is required to be traversed if destination lies in diagonally opposite region. A major limitation of above mentioned algorithms is that across link can be used only at first or last place and only for once even though multiple paths are available.

aEqualized routing algorithm proposed to overcome the limitations imposed by deterministic routing scheme and combines both aFirst and aLast algorithms. This algorithm mainly target networks with few nodes behaving as hot-spot nodes. Having prior knowledge of traffic pattern, it tags routers to run either aFirst or aLast routing algorithm. This is considered as main limi-tation of this approach because traffic behavior of NoC may change during run time which could result in a performance degradation.

Dynamic wormhole defection routing (DSWR) [37] is table based approach which calculates and stores path between each source and destination after mapping spidergon on 2xN mesh. It is fault tolerant and adaptive based upon concept of de ection routing [38]. But in prospect of low cost spidergon NoC it adds computational complexity and enhance the need of memory. Scalability is another issue that arises with table based approach.

# CHAPTER 3

# Reconfigurable Distributed Fault Tolerant Routing Algorithm for On-Chip Networks

## 3.1 Introduction

Network on chip (NoC) is emerging as a promising solution to overcome bus bottleneck for future multi core chips. Fault tolerance and quality of service issues are potential challenges for NoCs. In this paper, we propose a cost-effective fault tolerant routing algorithm for irregular 2D mesh without use of routing tables. We use one hop visibility of Logic Based Distributed Routing (LBDR) to eliminate routing tables. This algorithm handles one or multiple single link faults within 2D mesh and uses reconfigured paths (minimal and/or non-minimal), if links fail. We use turn model based approach to avoid deadlocks. Since our method does not require virtual channels to achieve deadlock freedom, it remains area and power efficient. On-chip networks have been aggressively discussed and researched in recent past as a dominant communication paradigm for complex Multi Processor System on Chips (MPSoCs) and Chip Multi Processors (CMPs) because of their scalability, energy-efficiency, reusability and reliability [4]. In an MPSoC or CMP architecture, reliability of NoC is affected by transient and/or permanent faults. Faults a ect the functionality and can degrade the performance of on-chip networks. Thus, it has become essential to design on-chip networks which can tolerate faults.

Faults can be tolerated by using many methods and majority of them are based on fault tolerant routing algorithms. Routing algorithms can be implemented as either table-based or algorithmic routing [26, 27]. In algorithmic routing mechanism, an algorithm is executed using software or special hardware circuits (generally combinational logic) employing FSM to compute appropriate router port. It is generally suitable for one topology and one routing method for that topology. The algorithmic mechanism is often efficinetin the terms of speed and area than table-based routing [26]. Table-based mechanism is used to deal with regular as well as irregular topologies. In this mechanism, each router keeps a table to store one output port channel (deterministic routing) for each destination from the current node in the network. The benefit of this mechanism is that it supports any topology and any routing method, even fault tolerant and congestion aware routing algorithms. The table based methods cannot scale

well since the table size increases with size of the network and may become impractical for large NoCs.

In the application specific platforms where communication transactions (concurrent/non-concurrent) among IP cores are known in advance, it is quite possible to use compression techniques [28] to reduce size of tables instead of straight forward table-based implementation. However, it is not a generic scenario in multi core systems.

In [29], authors proposed efficinet implementation of distributed routing algorithms for partial 2D meshes (irregular meshes) without routing tables. In this strategy, they suppress the need of tables on every router and enable the distributed implementation of any routing algorithm. This method is known as LBDR. It uses three bits at each output port of a router and a few logic gates.

Average communication latency is highly dependent on the chosen routing technique. Deadlock can be one of the major challenges for any routing technique which inde nitely postpones the delivery of a packet. Most of the fault tolerant routings use either turn models [7, 8, 9, 10, 11] or virtual channels (VCs) [12, 13, 14] based strategies to achieve deadlock freedom. However, attaching VCs to a physical channel is less expensive than adding one full physical channel, but this solution is still expensive, particularly in low area and power NoCs. The main area costs associated with VCs are the extrafficontrol lines to implement VCs and additional buffer space required for each VC. Thus, in NoCs where it is required to keep area and latency low, VC based approaches are not suitable candidates for low cost constraints.

Many deadlock free routing algorithms has been proposed in the literature that do not require any VCs for deadlock avoidance. XY routing algorithm for 2D mesh topology rst forwards the packet along x direction and then along y direction. It restricts turns from y direction to x direction as shown in Fig. 2.1a. The turn models are introduced by Glass and Ni [7]. They published three elegant partially adaptive routing techniques namely west- first, north-last and negative-first for n-dimensional meshes. These routing algorithms provide deadlock freedom by prohibiting one turn in each abstract cycle (clockwise and counter-clockwise) as shown in Fig. 2.1. Chiu [9] proposed one new Odd-Even (OE) turn model which has higher degree of adaptiveness than Glass's turn model. In this model, ES (East to South) & EN turns are not allowed on the nodes located at even columns, and NW & SW turns are not allowed at nodes located at odd columns.

In [8], a fault-tolerant routing technique is introduced that can tolerate only single-node failures in mesh-based NoCs. It cannot handle link failures. Boppana and Chalasani [30] proposed a deterministic fault tolerant routing algorithm that can tolerate faulty nodes and links. It uses additional VCs to avoid deadlocks, thus, results in area overhead. Masoumeh Ebrahimi et al. [25] presented a minimal and defect-resilient routing method to route packets in the presence of one faulty link using two VCs. David Fick et al. [31] proposed a table based implementation of routing algorithm to handle faulty components. In this paper, we present a reconfigurable, deadlock free and cost-efficinetfault tolerant routing algorithm without using VCs.

## 3.2 LDBR Basics

### 3.2.1 LDBR Overview

With the advances in deep sub-micron technology of VLSI, probability of failing of links has also increased. Therefore, topologies which are derived from 2D mesh having some manufactur-ing defects or faults have come into existence. The new routing methodology (LBDR) has the potential of handling such practical irregular topologies derived from initial 2D mesh without using routing tables. It can realize almost all deadlock free routing techniques efficiently. Dead-lock freedom of any routing technique can be ensured by enforcing some routing restrictions in LBDR implementation. A routing restriction at a router prohibits packet to take turn at next hop router during its travel from source to destination. For example, Fig.3.1 shows the routing restrictions corresponding to xy routing algorithm for 3X3 mesh.

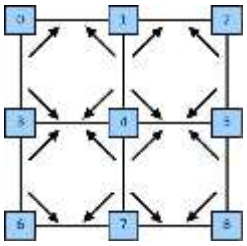

Figure 3.1: Routing Restrictions for xy Routing for 3 X 3 Mesh

In LBDR, each router is designed to keep two different sets of bits.

1. Routing Bits: Routing bits stand for the routing restrictions which are forced by underlying routing technique. These are used to a ord one hop visibility from the current router and restrict a packet to take few turns. For example, a routing bit ($R_{xy}$) states that a packet departing from x direction of the current router can take turn towards y direction on next router or not. If $R_{xy}$ bit is set, it means that packet can take the specified turn on next router.

2. Connectivity Bits: Connectivity bits stand for the connectivity of the current router within current network topology. Connectivity bits are designated with $C_x$, to de ne the connec-tivity at the x output port of the router. If $C_x$ is set, it means that the current router is having connectivity with neighbor in x direction.

For 2D mesh topology, this uses two routing bits and one connectivity bit for each output port. These bits are first computed o -line for a particular combination of topology and routing tech-nique. Table 3.1 shows the routing and connectivity bits corresponding to xy routing restrictions as shown in Fig. 3.1.

### 3.2.2 Next Hop Computation Logic

This implementation selects north as next hop (nextHop), if current router has connectivity in north ($C_n$=1) and if

Table 3.1: Routing and Connectivity Bits for LBDR Implementation of xy Routing for 3 3 Mesh

| Router ID | Rne | Rnw | Ren | Res | Rse | Rsw | Rwn | Rws | Cn | Ce | Cw | Cs |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | | | |
| 0 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 1 |
| 2 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| 3 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 |

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 4 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| 5 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1 |
| 6 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 |
| 7 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 8 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 0 |

1. destination is in north direction, or

2. destination is north east quadrant and $R_{ne}$=1, or

3. destination is north west quadrant and $R_{nw}$=1

## 3.3 Proposed Method

In this section, we present a deadlock free, reconfigurable, fault tolerant and deterministic routing algorithm which is designed to handle one or multiple single link faults within 2D NoC mesh topology. We modify and extend LBDR to implement our proposed method.

### 3.3.1 Link Failure Classification and Deadlock Freedom

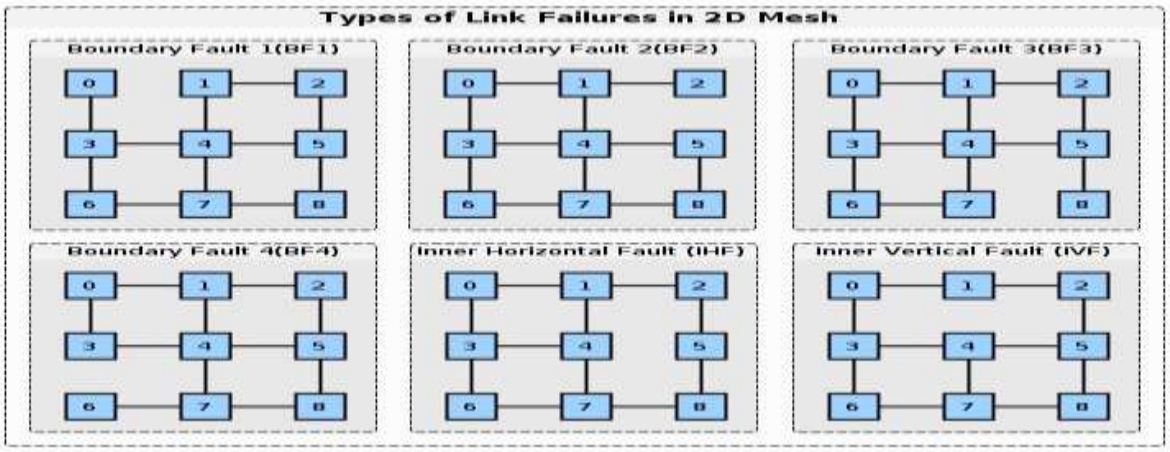


Figure 3.2: Types of Link Failure in 2D Mesh

We can classify link failures into six different categories depending upon their locations in mesh as shown in Fig. 3.2. According to Dally [39], Channel Dependency Graph (CDG) can be used to ensure deadlock freedom in the case of boundary faults (BF1, BF2, BF3 and BF4). CDG is a graph in which nodes and edges are network channels and dependencies between network channels, respectively. By constructing CDG for boundary of BF1 as shown in Fig. 3.3a, we can

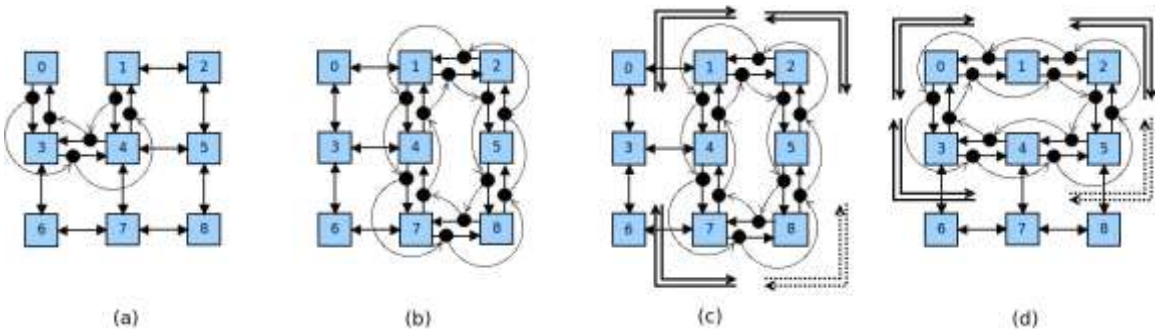


Figure 3.3: Channel Dependency Graphs for (a) Boundary of BF1 (b) Boundary of IHF (c) Deadlock Avoidance in IHF (d) Deadlock Avoidance in IVF (dashed line shows restricted turn)

see that it does not have any cycle. Similarly, we can prove that boundaries of faults BF2, BF3 and BF4 will not have any cycle in their CDGs. Thus, these all boundaries are deadlock free.

However, when we construct CDG for boundary involved in IHF, it contains cycles as shown in Fig. 3.3b. Hence, boundary of IHF is not deadlock free. To break cycles in IHF's CDG, we have used new routing restrictions as shown in Fig. 3.3c. These restrictions prohibit a packet from taking EN and SW turns in IHF boundary. Similarly, we can prove that boundary of IVF with new routing restrictions is also cycle free as shown in Fig. 3.3d, thus, deadlock free. We have used xy and new routing restrictions in non-faulty and faulty regions, respectively, to prevent deadlock in our proposed method.

### 3.2.2 Reconfiguration of Routing paths

In the absence of link failures, proposed method works as deterministic xy routing. But if a fault occurs, xy algorithm alone is unable to find paths between certain pairs of nodes. Thus, we need to reconfigure some paths to achieve deadlock freedom and it may result in some non-minimal paths.

Table 3.2 and Table 3.3 show reconfigured paths in IHF and IVF boundaries, respectively, ac-cording to new routing restrictions. Similarly, we can recon gure paths for remaining boundaries of faults (BF1, BF2, BF3 and BF4). We de ne a recon guration method for all routers that are part of any boundary of fault. This recon guration method is activated as soon as a built-in self test (BIST) unit detects a fault. We assume that faults are detected by BIST unit. After fault detection, we change the routing and connectivity bits for concerned routers according to above discussion.

### 3.3.3 Fault Tolerant Routing Implementation

We propose a deterministic fault tolerant routing algorithm based on xy routing for 2D mesh. LBDR implementation does not support irregular topologies that contain non-minimal paths (previously minimal in regular 2D mesh). Rodrigo et al [40] proposed a method LBDRdr that uses Deroute option in LBDR to support such cases. Since Deroute option results in extra bits(8) per router, we completely eliminate it and provide a new Misroute option without using additional bits.

Table 3.2: Reconfigured Paths for IHF Boundary

| Path No | Old Path According to *xy* Routing in 2D Regular Mesh | New Path According to Proposed Method in Irregular 2D Mesh | New Path in Irregular 2D Mesh | New Path in Irregular 2D Mesh compared to Regular 2D Mesh |
|---|---|---|---|---|
| 1 | $4 \rightarrow 5$ | $4 \rightarrow 1 \rightarrow 2 \rightarrow 5$ | Minimal | Non-minimal |
| 2 | $5 \rightarrow 4$ | $5 \rightarrow 2 \rightarrow 1 \rightarrow 4$ | Minimal | Non-minimal |
| 3 | $4 \rightarrow 5 \rightarrow 2$ | $4 \rightarrow 1 \rightarrow 2$ | Minimal | Minimal |
| 4 | $5 \rightarrow 4 \rightarrow 1$ | $5 \rightarrow 2 \rightarrow 1$ | Minimal | Minimal |
| 5 | $4 \rightarrow 5 \rightarrow 8$ | $4 \rightarrow 7 \rightarrow 8$ | Minimal | Minimal |
| 6 | $5 \rightarrow 4 \rightarrow 7$ | $5 \rightarrow 2 \rightarrow 1 \rightarrow 4 \rightarrow 7$ | Non-minimal | Non-minimal |
| 7 | $7 \rightarrow 8 \rightarrow 5$ | $7 \rightarrow 4 \rightarrow 1 \rightarrow 2 \rightarrow 5$ | Non-minimal | Non-minimal |
| 8 | $7 \rightarrow 8 \rightarrow 5 \rightarrow 2$ | $7 \rightarrow 4 \rightarrow 1 \rightarrow 2$ | Minimal | Minimal |

Table 3.3: Reconfigured Paths for IVF Boundary

| Path No | Old Path According to *xy* Routing in 2D Regular Mesh | New Path According to Proposed Method in Irregular 2D Mesh | New Path in Irregular 2D Mesh | New Path in Irregular 2D Mesh compared to Regular 2D Mesh |
|---|---|---|---|---|
| 1 | $4 \rightarrow 1$ | $4 \rightarrow 3 \rightarrow 0 \rightarrow 1$ | Minimal | Non-minimal |
| 2 | $1 \rightarrow 4$ | $1 \rightarrow 0 \rightarrow 3 \rightarrow 4$ | Minimal | Non-minimal |
| 3 | $2 \rightarrow 1 \rightarrow 4$ | $2 \rightarrow 1 \rightarrow 0 \rightarrow 3 \rightarrow 4$ | Non-minimal | Non-minimal |
| 4 | $5 \rightarrow 4 \rightarrow 1$ | $5 \rightarrow 2 \rightarrow 1$ | Minimal | Minimal |
| 5 | $0 \rightarrow 1 \rightarrow 4$ | $0 \rightarrow 3 \rightarrow 4$ | Minimal | Minimal |
| 6 | $3 \rightarrow 4 \rightarrow 1$ | $3 \rightarrow 0 \rightarrow 1$ | Minimal | Minimal |
| 7 | $3 \rightarrow 4 \rightarrow 5 \rightarrow 2$ | $3 \rightarrow 0 \rightarrow 1 \rightarrow 2$ | Minimal | Minimal |
| 8 | $4 \rightarrow 5 \rightarrow 2$ | $4 \rightarrow 3 \rightarrow 0 \rightarrow 1 \rightarrow 2$ | Non-minimal | Non-minimal |

Our routing implementation is divided into the follflowing three steps.

1. Compute Relative Position of Destination Router: Using current router ($X_c$, $Y_c$) and destination router ($X_d$, $Y_d$) coordinates, we compute relative position of destination. We use two variables ($d_1$ and $d_2$) to encode it. If $d_1$ is set to NULL and $d_2$ is set to WEST , it means that destination is in west direction. If $d_1$ and $d_2$ both are NULL, it means destination is Local IP Core.

2. Invoke Recon guration Method on Occurrence of Fault: If BIST detects any fault, we invoke recon guration method (described in Section 3.3.2) to reset routing and connectivity bits.

3. Next Hop Computation: We compute next hop (nextHop) using LBDR. If it is unable to compute next hop (nextHop is set to NULL), we use Misroute option. Algorithm 1 describes Misroute option for IHF boundary. It computes next hop direction only for routers (4, 5 and 7) depending on their locations on IHF boundary and input

direction (ip_dir) of the incoming packet. We explain nextHop computation with follflowing two examples.

- If router 4 receives a packet coming from west direction (ip_dir is WEST) and targets for router 5, LBDR does not produce any next hop (nextHop is set to NULL). Because there is no connectivity between router 4 and router 5 ($C_e$ is 0). Thus, we compute north as next hop using Misroute option

- If router 7 receives a packet coming from west direction (ip dir is WEST) and targets for router 5, LBDR does not produce any next hop (nextHop is set to NULL). Although router 7 is having connectivity toward $C_e$ and $C_n$, but routing restrictions $R_{en}$ (north turn on next router in east direction) and $R_{ne}$ (east turn on next router in

Algorithm 1 : Next Hop Computation on IHF Boundary using Misroute Option

Input: ip_dir, $d_1$, $d_2$

Output: nextHop

Procedure:

if (nextHop == NULL) then

/* Compute next hop for Node 4 of IHF Boundary */ if (ip_dir == WEST|| ip_dir == SOUTH) then

if ($d_1$ == NULL && $d_2$ == WEST) then

nextHop = NORTH

end if

end if

/* Compute next hop for Node 5 of IHF Boundary */ if (ip_dir == EAST) then

```
if (d2 == WEST) then
   if (d1 == NULL|| d1 == SOUTH) then
      nextHop = NORTH
  end if
   end if
   end if

/* Compute next hop for Node 7 of IHF Boundary */ if
(ip_dir == WEST) then
   if (d1 == NORTH && d2 == EAST) then
      nextHop = NORTH
   end if
   end if
   end if
```

north direction) are set to zero by recon guration method. Thus, we compute north as next hop using Misroute option because destination router is NE quadrant.

Since Misroute option for IHF always redirects a packet in north direction only, it does not require additional bits. Similarly, we can compute west as next hop direction for routers (1, 2 and 4) of IVF boundary using Misroute option.

## 3.4 Experimental Set Up and Result Analysis

Experimental Setup In this section, we evaluate proposed method using a cycle accurate NoC simulator NIRGAM (NoC Interconnect Routing and Application Modeling). NIRGAM is developed using systemC. We have implemented our method by modifying NIRGAM for link failures. All simulations are performed on 7X7 mesh using wormhole switching mechanism. The input channel buffer and packet sizes are set to 6 its and 8 its respectively. Each simulation was run for 15000 cycles with warm-up sessions of 5000 cycles and generates traffic for 10000 cycles. As a performance metric, we use communication latency per channel defined as number of cycles between arrival and departure of packet through a router. In each plot, xy plane is used to represent channels and z axis is used to plot latency in number of cycles. We randomly inject one fault in each simulation for 100 times so that

recon guration method is invoked repetitively in one simulation to produce exhaustive average results. We compare introduced scheme (with faults) with xy routing (without faults) for uniform and hot spot traffic models.

Uniform traffic Model In the uniform traffic model, each IP core generates packets and sends them to other IP cores using a uniform probability distribution. Figures 4.5 and 4.6 show the average communication latencies per channel basis for both xy and proposed method at high traffic load. At low traffic load, results are almost similar, thus, we have not shown them. But when we increase traffic load, it can be observed that the xy routing outperforms our method as expected because xy algorithm (without faults) incorporates relatively long term and more global information about uniform traffic characteristic. Since, our method uses non-minimal paths to tolerate faults, latencies are increased as compared to xy.

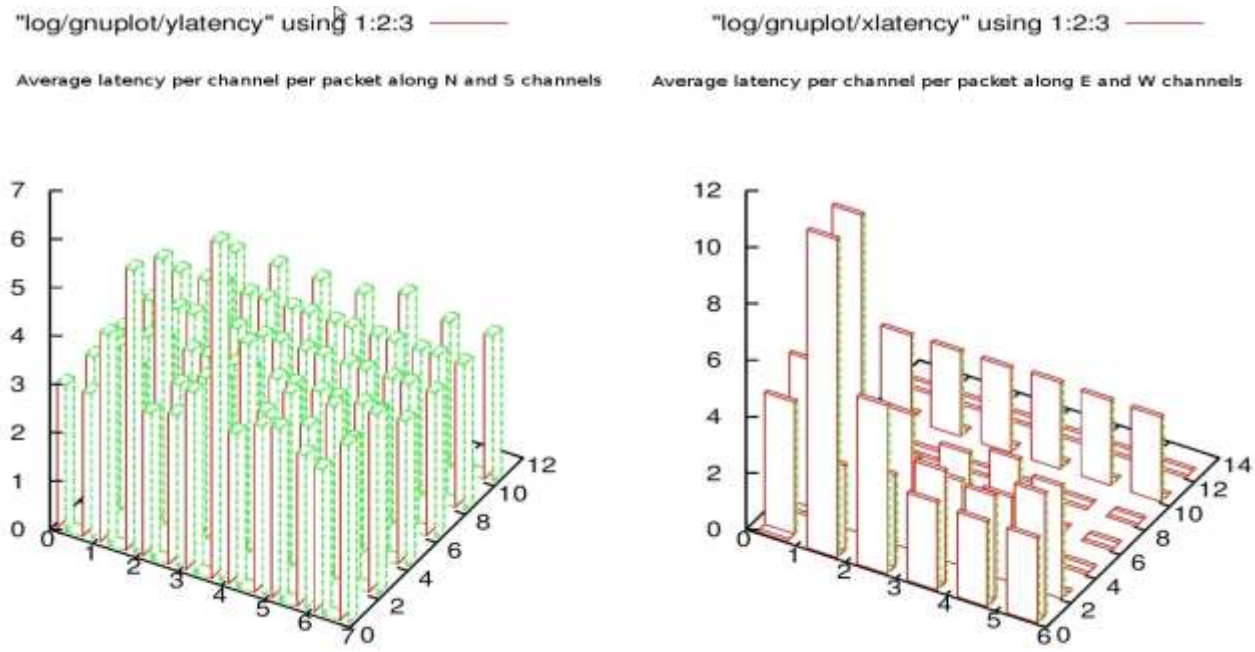


Figure 3.4: xy algorithm without faults

Hot Spot traffic Model Hot spot traffic model is considered as a more realistic traffic pattern, in which a few hot spot nodes receive extra packets in addition to the regular uniform traffic. We set node 10 as hot spot with .3 probability of getting addition traffic. Figures 4.7 and 4.8 show the average communication latencies for hot spot traffic at high load. The performance of xy and our methods is affected by hot spot traffic at high load as compared to uniform traffic.

In the absence of faults, proposed method works as xy routing. Results indicate that latencies of proposed method (with faults) are greater than the latencies of xy routing (without faults), because our method follows some non-minimal paths to tolerate faults whereas xy algorithm always follows minimal paths in the absence of faults. However, if faults occur, xy algorithm cannot nd paths between certain pairs of nodes, thus, fails.

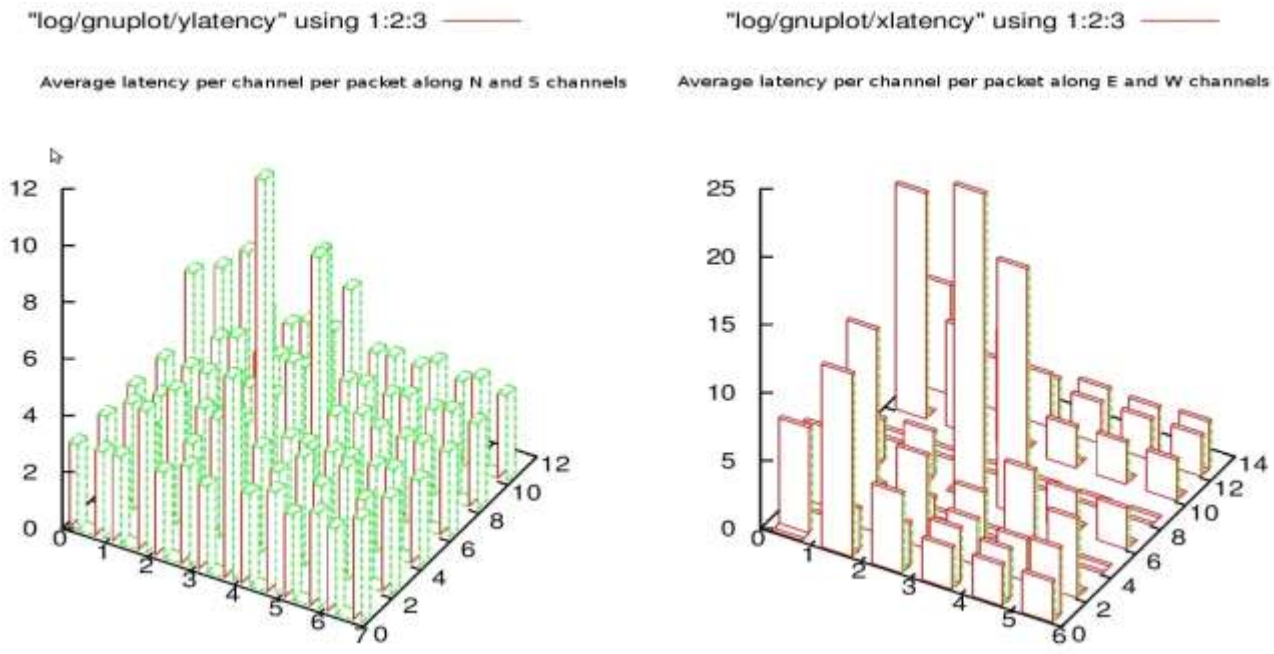


Figure 3.5: proposed algorithm with faults

# Chapter 4

# A low-cost Congestion-aware and Highly Adaptive Routing Method for 2D and 3D On Chip Networks

## 4.1 Introduction

Three-dimensional very large-scale integration (3D VLSI) is emerging as a promising solution to physical limitations of semi-conductors manufacturing processes [41]. In 3D-Integration, multi-ple 2D dies are vertically stacked on top of each other with Through-Silicon-Vias (TSVs). The main advantage of 3D ICs is the considerable reduction in the count and length of global inter-connects, which leads to increase in performance. On the other hand, on-chip interconnection networks are aggressively discussed and researched in past few years as a prominent and promis-ing communication paradigm for complex Multi Core Systems on Chips (MCSoCs) and Chip Multi Processors (CMPs) architectures because of their energy-efficiency, parallelism, increased predictability, scalability, reusability and reliability [4]. NoC efficiency and performance is highly dependent on the underlying routing algorithm we choose. Higher the degree of adaptiveness (path diversity) of routing algorithm, lower is the probability for a packet to enter faulty or con-gested area. Thus, the focus of this research is to enhance the performance of routing algorithm by increasing degree of adaptiveness (allflowing additional minimal paths), use of less number of virtual channels for deadlock and providing congestion awareness.

On the basis of inclusion of current network information in routing decision, routing algorithms are classifed as either deterministic or adaptive. Deterministic routing algorithms always choose the same path between a given pair of source and destination nodes without considering current network status, even if there exists multiple possible paths. On the other hand, adaptive routing algorithms use current state (traffic and/or link status) of network in making routing decision to avoid congested or faulty links/regions of the network. It offers better throughput and latency than deterministic routing, especially under bursty and hot-spot traffic patterns [4].

Network congestion may lead to increased power consumption and communication delay, thus degrades the performance of on-chip networks. However, the performance of NoC can

be im-proved by routing packets along non-minimal path through less congested areas and distributing traffic load across the network. Minimal routing methods guarantee the shortest route between source to destination but, it is imprudent to neglect the promising performance improvements provided by non-minimal routing schemes. For example, if all output channels corresponding to minimal routing paths are faulty; routing the packets along a non-minimal route may be the only alternative. We have shown that the proposed routing algorithm results in signi cant performance improvement compared to existing routing schemes under certain traffic patterns designed for NoCs. We have also shown that deadlock avoidance method used by our routing algorithm is cost efficinetas it uses less number of virtual channels compared to other well known adaptive routing methods [34, 32, 18].

The rest of our paper is organized as follow. Section ?? reviews the related work. In Section 4.2, we discuss our proposed work. Section 4.3 evaluates and compares the e ectiveness of proposed method using synthetic and realistic traffic profiles. Finally, Section ?? concludes with results analysis and future work

## 4.2 Proposed Work

In this section, we present CHARM routing method for 2D and 3D meshes. We have proposed a novel turn model, which is a major improvement in existing 2D turn models used by several routing algorithms [24, 42, 43, 44, 19, 45] including Mad-y [14], which is base of all these al-gorithms. We have also extended our proposed 2D turn model for 3D, which can be further extended for n-dimensional mesh as well.

### 4.2.2 Background

In mesh networks, routing turns can be of three types: 0-degree (packet switches from one virtual channel to another virtual channel in same direction), 90-degree (packet switches from one virtual channel to another virtual channel in different dimension) and 180-degree (also known as U-turn in which packet switches from one virtual channel to a virtual channel in opposite direction of same dimension). For minimal routing, 180-degree turns cannot be used. The proposed method CHARM provides more minimal and non-minimal paths between source and destination by allflowing some routing turns which were prohibited in Mad-y. However, these permitted routing turns create cycles in channel dependency graph, but we

have shown that proposed routing method is deadlock free using Duato's well known theorem [46]. Duato has proved that cycles can be allowed in channel dependency graph provided that extended channel dependency graph is acyclic.

Goal of proposed method is to enhance the capability of existing virtual channels in Mad-y to route packets minimally or non-minimally around hotspot and congested areas of the network. Since our method is based on Mad-y and LEAR algorithms, we rst brie y describe these

### 4.2.2 Mad-y and LEAR Algorithms

The turn model representation is an easy and effective way to describe routing algorithms and their routing restrictions. Figures ?? and 2.3 show turn model representation of Mad-y and LEAR routing algorithms, respectively. Both Mad-y and LEAR algorithms use double-y network that uses one and two virtual channels along X and Y dimensions, respectively.

In order to achieve deadlock freedom, Mad-y imposes some constraints on routing turns. It pro-hibits four 90-degree turns (N2-W , S2-W , E-N1 and E-S1) as shown in Figures ??(i) and ??(ii). It also prohibits two 0-degree turns (S2-S1 and N2-N1) as shown in Figure ??(iii). Since it is minimal routing algorithm, it does not permit any 180-degree turn. We can deduce follflowing restrictions from above constraints. A packet is allowed to take turn towards east using 90-degree turns (N1-E and S1-E) and 0-degree turns (N1-N2 and S1-S2) only when it has not already routed to east. Because, after using east channel, the packet cannot use N1 or S1. A packet is allowed to take 90-degree turns (W -N2 and W -S2) from west and 0-degree turns (N1-N2 and S1-S2) only when it does not need to take west turn further. Because, the packet cannot be forwarded towards west after using N2 or S2. A packet cannot use N2 or S2 at source node, if it needs to be forwarded west.

LEAR permits some 180-degree turns, as shown in Figure 2.3(iv) in addition to the turn restric-tions used by Mad-y. Both Mad-y and LEAR routing algorithms are proved deadlock free on the basis of work of Dally and Seitz [39]. They show that a routing algorithm is deadlock free if all channels can be assigned numbers such that the algorithm always forwards each packet along channels with strictly increasing (or decreasing) order.

### 4.2.3 Proposed Method on 2D Mesh

An acyclic channel dependency graph requirement to avoid deadlocks places unnecessary, thus avoidable restrictions on the routing turns in a routing algorithm. Mad-y routing method is proved deadlock-free using acyclic channel dependency graph [39]. Thus, its routing function cannot use all quali ed turns to forward packets through less congested areas. The proposed method imposes substantially fewer restrictions on routing turns (specially on 90-degree) us-ing [46], thus it provides additional minimal and non-minimal paths between source and desti-nation than Mad-y and LEAR.

Figure 4.1 shows turn model representation of 2D-CHARM. A packet is permitted to use rst virtual channel at any time as shown in Figure 4.1(i). It can use second virtual channel only if it has already routed to negative direction of X dimension (west) as shown in Figure 4.1(ii). It is allowed to take only two 180-degree turns from west to east (W -E) and south to north (S2-N2) as shown in Figure 4.1(iv) only if it has completed routing in west and south directions, respectively. In short, in order to avoid deadlocks, 2D-CHARM imposes follflowing constraints on routing turns:

**1.** It prohibits two 90-degree turns (S2-W and N2-W ).

**2.** It allows 0-degree turns (S1-S1, N1-N1, S2-S2 and N2-N2) as shown in Figure 4.1(iii). It allows 0-degree turns (S1-S2, N1-N2, S2-S1 and N2-N1), as shown in Figure 4.1(iii), with some restrictions. It allows these restricted turns only when packet does not need to be forwarded further west.

**3.** It permits some 180-degree turns.

We can deduce follflowing restrictions from above constraints:

**1.** A packet is allowed to take 90-degree turns (W -S2 and W -N2) only when it does not need to take west turn further. This restriction is because of prohibited 90-degree turns (N2-W and S2-W ).

**2.** A packet cannot use N2 or S2 at source node, if it needs to be forwarded west.

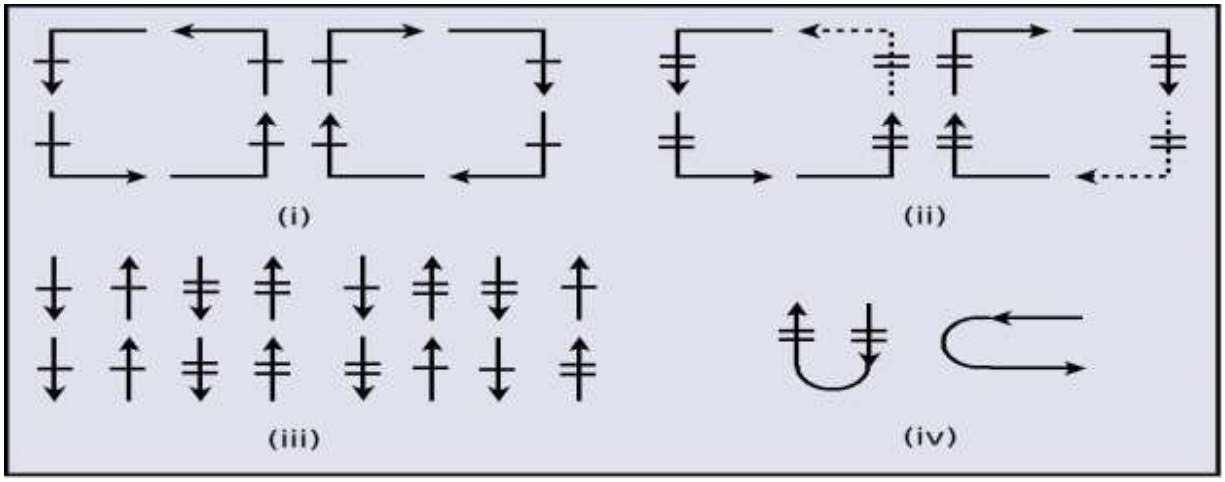


Figure 4.1: 2D-CHARM

### 4.2.4 Proposed Method on 3D Mesh

In this section, we extend 2D-CHARM for three dimensions. 3D-CHARM deploys double-yz network that uses one virtual channel along X dimension (+X: east, X: west), two virtual channels along Y dimension (+Y : north, Y : south) and Z dimension (+Z: up, Z: down) to achieve high degree of adaptiveness. Figures ?? and ?? show turn model representations for different planes (Y Z, XY and XZ) of 3D mesh topology. A packet is permitted to use rst virtual channel at any time as shown in Figures ??(i) and ??(ii). It can use second virtual channel only if it has already routed to negative directions (west and south) of all lower dimensions. It is allowed to take 180-degree turn from west to east, south to north and down to up only if it has completed routing in west, south and down directions, respectively.

For XY -plane, 3D-CHARM imposes same constraints as in 2D-CHARM (Section 4.2.3).

For Y Z-plane, 3D-CHARM imposes follflowing constraints on routing turns:

1. It prohibits four 90-degree turns (U2-S1, U2-S2, D2-S1 and D2-S2) as shown in Figures ??(iii) and ??(iv).

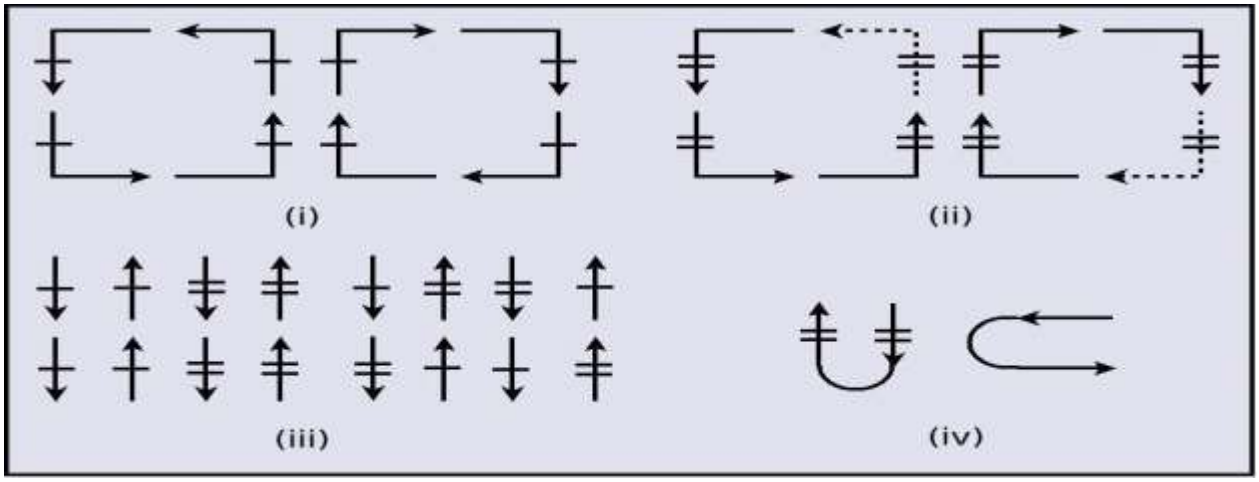


Figure 4.2: XY –plane

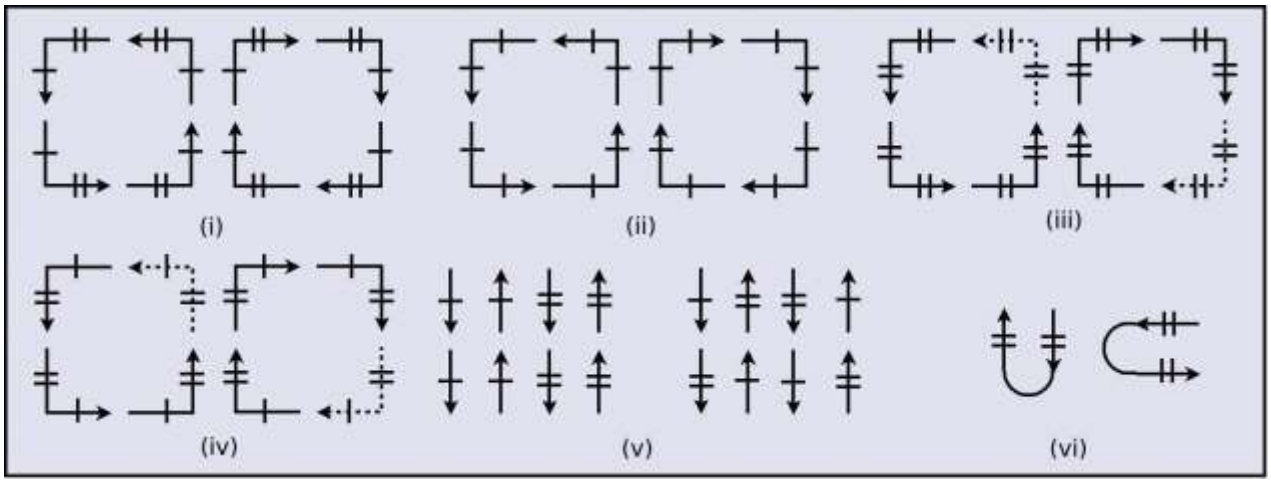


Figure 4.3: XZ-plane

2. It allows 0-degree turns (U1-U1, D1-D1, U2-U2 and D2-D2) as shown in Figure ??(v). 3D-CHARM allows 0-degree turns (U1-U2, U2-U1, D1-D2 and D2-D1) as shown in Fig-ure ??(v), with some restrictions. It allows these restricted turns only when packet does not need to be forwarded further west or south. Similarly, we can nd allowed and restricted 0-degree turns for north and south as well.

3. It permits some 180-degree turns as shown in Figure ??(vi).

We can deduce follflowing restrictions from above constraints:

1. A packet is allowed to take 90-degree turns (S2-U2, S1-U2, S2-D2 and S1-D2) only when it does not need to be forwarded to west or south further. This restriction is due to prohibited 90-degree turns (U2-S1, U2-S2, D2-S1 and D2-S2).
2. A packet cannot use U2 or S2 at source node, if it needs to be forwarded west or south.

Constraints on routing turns for XZ-plane can be deduced in similar fashion as in XY - plane.

Functionality of CHARM is divided into two phases: route computation phase and output channel selection phase. On the basis input channel (on which packet has arrived) and relative position of destination node with respect to current node, routing function computes a set of output channels using turn model explanation discussed above. By comparing turn models as shown in Figure ??, we can conclude that CHARM results in a larger set of output channels due to high degree of adaptiveness than Mad-y and LEAR.

Selection function selects one output channel from the set of output channels provided by the route computation function. Our selection function rst checks all quali ed output channels corresponding to shortest routes and forwards the packet to the output channel in which the corresponding next hop node has its congestion status ag set to zero (and possibly, correspond-ing to non-escape channels). If the congestion status ags of all next hop nodes on shortest routes are set to one, the congestion status ag of each eligible non-minimal route is inspected. If there exist such non-minimal routes, which are not congested, our method selects one of the output channel to forward the packet (and possibly, corresponding to non-escape channels). Our method prefers adaptive output channels over escape channels[1], because it results in increased probability of escape output channels being available when they are required to avoid deadlocks.

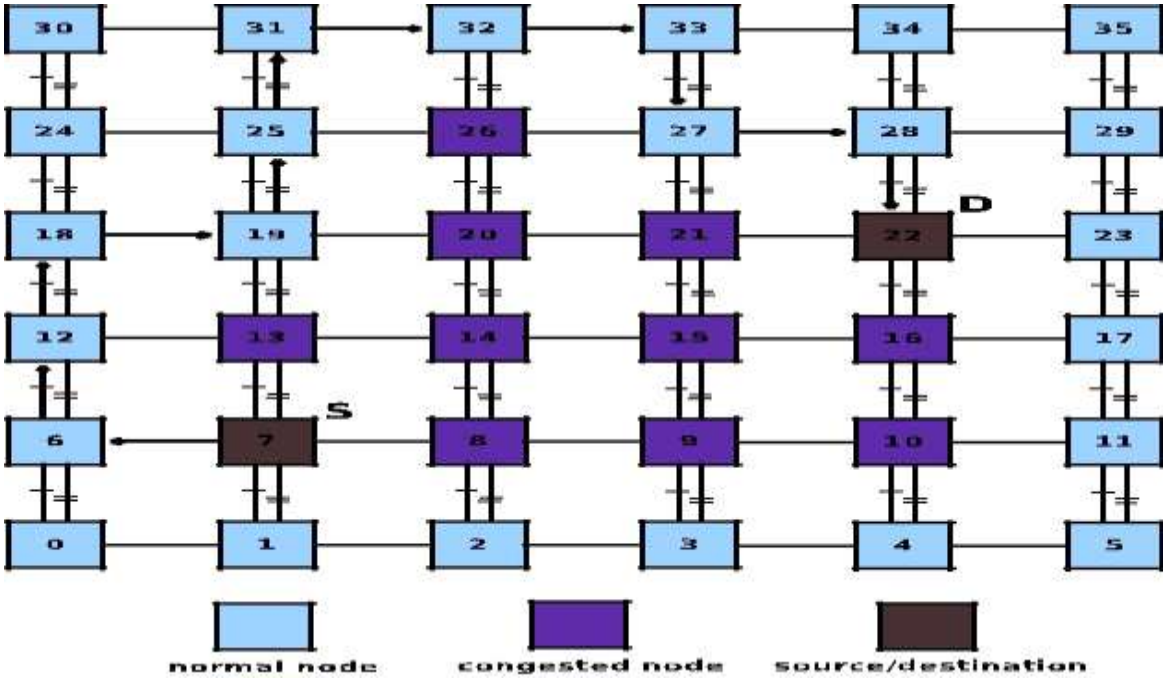


Figure 4.4: Example of CHARM method

Figure 4.4 shows an example of CHARM for a 6X6 mesh network. A source node 7 sends packet to destination node 22. CHARM routing function provides all six channels (E, N1, N2, W , S1 and S2) as output at node 7. All neighboring nodes on minimal node in paths (E, N1,N2) are in congestion region. If west channel is selected, packet has ve choices (N1, N2, S1, S2 and W ) on next node 6. Since node 6 is on west border, west cannot be selected as output dirrection.

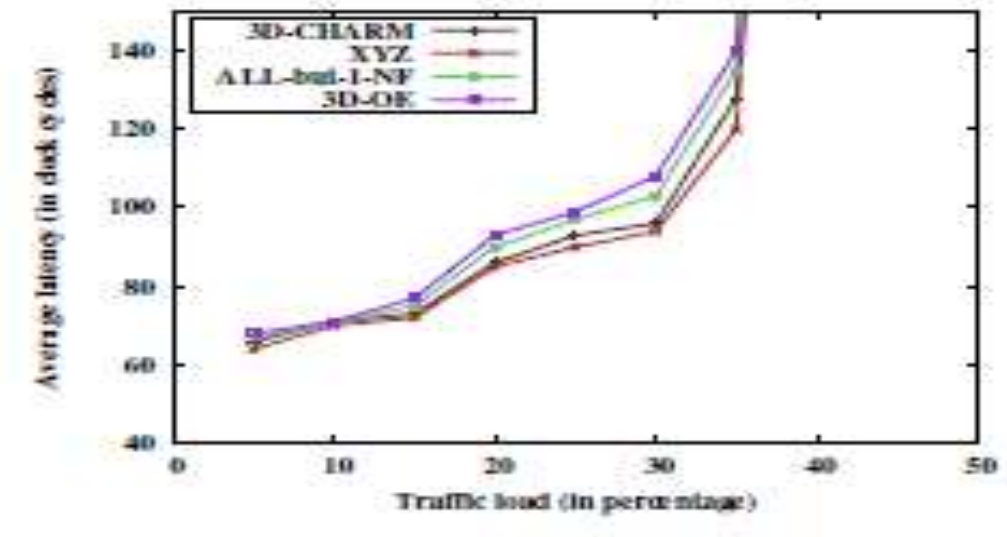


Figure 4.5: Average latency in 4 X 4 X 4 mesh at different loads under uniform traffic

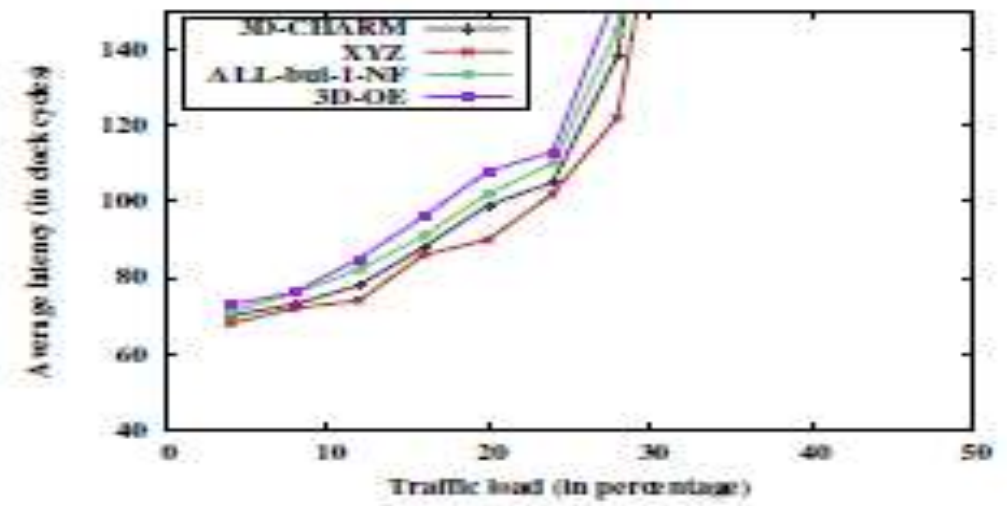


Figure 4.6: Average latency in 6 X 6 X 6 mesh at different loads under uniform traffic

direction. Now at node 6, packet has four choices (N1, N2, S1 and S2). CHARM selection function will give preference to non-congested minimal paths, it selects north direction at node 6. The same strategy is followed till the packet is received at destination. Specially at node 33 (or 28), CHARM routing function provides both S1 and S2 as output directions. Whereas other methods provide only E-S2 (Figure ??) as E-S1 is prohibited. This example shows CHARM method is capable of routing packets around congested region and distributing the traffic across the network.

### 4.2.5 Deadlock and Livelock Freedom

With deterministic routing, packets can be routed over single output channel at each node. Thus, it is mandatory to remove all cyclic dependencies between network channels in order to achieve deadlock freedom. In adaptive routing, packets often have several options for routing at each node. Thus, it is not mandatory to eliminate all cyclic dependencies between channels, provided that every packet can be forwarded on a route whose channels are not involved in cyclic dependencies. The channels involved in these acyclic routes are considered as escape channels from deadlocks (cycles).

The deadlock-freedom of CHARM is assured by using Duato's theory [46] stated as follows:

Theorem 1. (Duato's Theorem) For a given interconnection network I, a connected and adaptive routing function R is deadlock-free if there exists a routing subfunction $R_1$ R, that is connected and has acyclic extended channel dependency graph.

Following Duato's terminology, the route computation function of CHARM is denoted by R and the set of channels used by R is denoted by C. To assure deadlock freedom of CHARM, we rst identify the subset of channels $C_1$ C, that de nes routing subfunction $R_1$ R that is connected and has an extended channel dependency graph (ECDG) with no cycles arising from direct, indirect, direct-cross and indirect-cross dependencies. For CHARM, $C_1$ has all virtual channels except N1, S1, U1 and D1.

Lemma 1. The routing subfunction $R_1$ is connected.

Proof. $R_1$ routing subfunction with channel set $C_1$ is all-but-one-negative- rst [7] routing algorithm. Since non-minimal all-but-one-negative- rst routing is connected, so $R_1$ is connected. □

Lemma 2. Extended channel dependency graph of channel set $C_1$ with additional channel (N1,S1, U1 and D1) introduced by R, does not result in any cyclic dependencies.

Proof. There is no direct-cross dependency in ECDG of $C_1$ as routing function R does not add any new routing option between channels of $C_1$ directly. However, additional channels introduced by routing function R add new routing options between channels of $C_1$ indirectly, but it does not produce any indirect-cross dependency. Additional channels introduced by R can cause only indirect dependencies between west (south) channels as a packet can use west (south) channel and later can use west (south) channel of different row and column. But this indirect dependency does not introduce any cycle in ECDG of $C_1$. The ECDG for $C_1$ has no dependencies from a channel in north, east, south, up, or down directions to a channel in the west direction, so the west channels are always used before all other channels in $C_1$. Hence, these indirect dependencies introduce new dependencies between only the west virtual channels and create no cycles using only the west virtual channels. Similarly ECDG for $C_1$ has no dependencies from a channel in north, up, or down directions to a channel in the south direction, so the south channels are always used before other channels (north, up and down) in $C_1$. These indirect dependencies introduce dependencies from west to south or between south virtual channels. Hence, these indirect dependencies create no cycles using only the south virtual channels. Since there are no indirect and direct dependencies, which produce cycle in ECDG. Therefore ECDG of $C_1$ is acyclic.

Theorem 2. The proposed routing algorithm is deadlock-free.

Proof. It can be concluded from Lemma 1 & Lemma 2 and using Theorem 1 that proposed routing algorithm is deadlock-free.

Non-minimal routing algorithms are susceptible to livelock. The proposed routing algorithm is proved to be livelock free using follflowing theorem.

Theorem 3. The proposed routing algorithm is livelock-free.

Proof. From the discussion in Section 4.2, we can notice that whenever a packet is routed in the east direction, it is not allowed to route it back in the west direction. So, in the worst scenario, packet may reach to the west most column then it starts to move towards destination column. Similarly, whenever a packet is routed in the north direction, it is not allowed to route back in south. Similarly, whenever a packet is routed in the up direction, it is not allowed to route back in down. So, in the worst case, packet may reach to the down most XY

-plane then it starts to move towards destination node. In each dimension, only one 180-degree turn is allowed. Therefore, after a limited number of hops, the packet reaches to its destination node. Thus, proposed routing algorithm is livelock free.

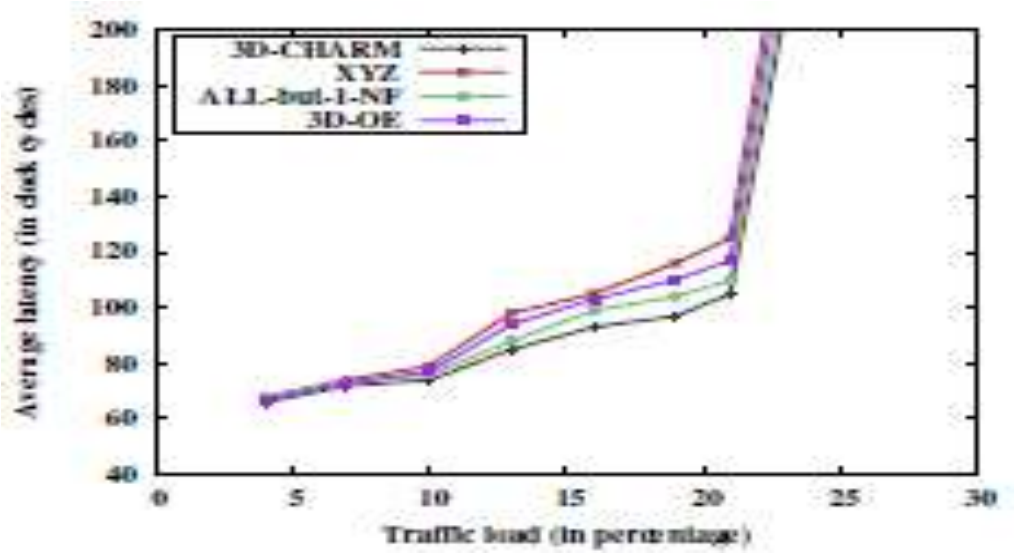


Figure 4.7: Average latency in 4 X 4 X 4 mesh at different loads under hot-spot traffic

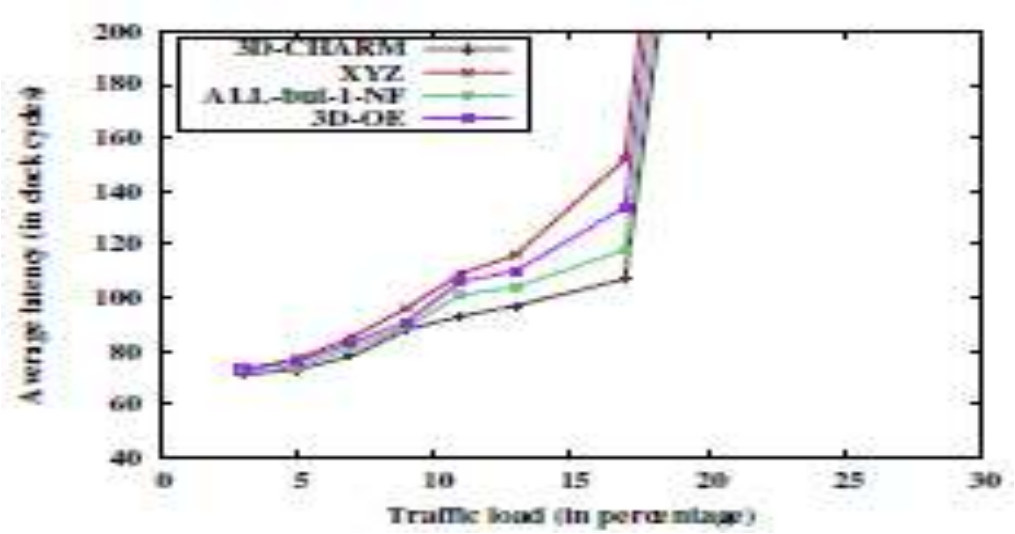


Figure 4.7: Average latency in 4 X 4 X 4 mesh at different loads under hot-spot traffic

## 4.3 Experimental Setup and Result Analysis

We have evaluated proposed routing method (2D-CHARM and 3D-CHARM) with real and synthetic traffic profiles. Simulation results for a 3D (say, 6 X 6 X 6) mesh and for a

larger 2D (say, 10 X 10) mesh have a similar behavior [7]. Thus, we have shown results for 3D-CHARM.

### 4.3.1 Experiment Setup

To evaluate effectiveness of the proposed routing method (3D-CHARM), we have implemented three other well-known routing methods. These methods include the dimension order routing (XYZ), 3D Odd-Even (3D-OE) [33], and all-but-one-negative- rst (ALL-but-1-NF) [7]. We eval-uate 3D-CHARM and all other routing methods on both synthetic and realistic traffic profiles. As synthetic traffic profiles, we consider both uniform (random) and non-uniform (hotspot and transpose) traffic patterns. For realistic communication profiles, we consider traces of some ap-plication suites extracted from benchmark suite E3S [?]. In order to perform all required set of experiments involving various routing methods and traffic profiles, we modi ed and extended [47], a cycle-accurate and open source SystemC based NoC simulator.

For all experiments, we consider 4 X 4 X 4 and 6 X 6 X 6 meshes. Packet size and input-channel buffer size for each virtual channel are kept constant for all experiments and set to 6 and 8 its, respectively. The simulator is warmed up for 5,000 cycles and afterwards; the average performance is measured over another 30,000 cycles, out of which traffic is generated over 20,000 cycles. Congestion threshold is set to 66% of total buffer size. As communication performance parameter, we consider latency (delay). It is defined as the time difference (in clock cycles) between header it injection from source router and tail it reception at destination router.

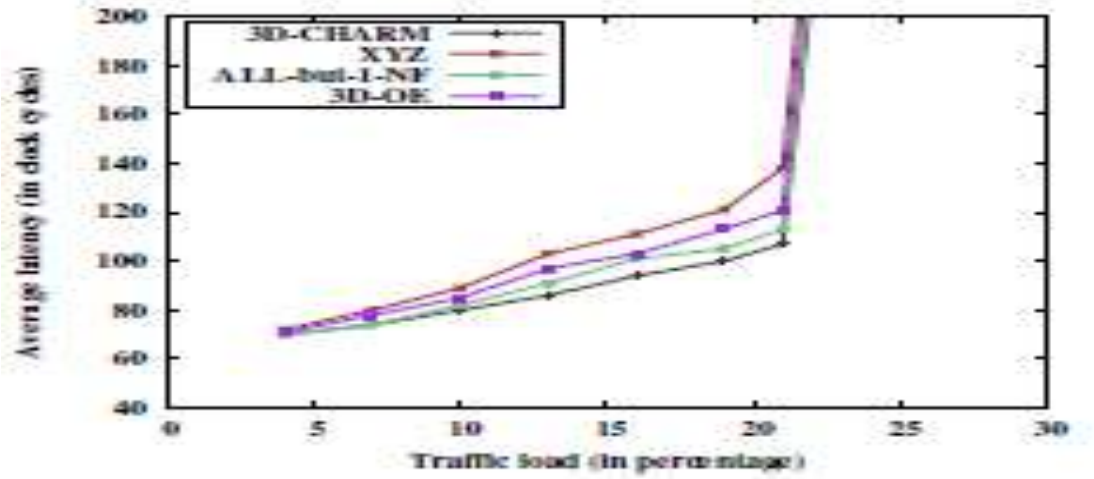


Figure 4.9: Average latency in 4 X 4 X 4 mesh at different loads under transpose traffic

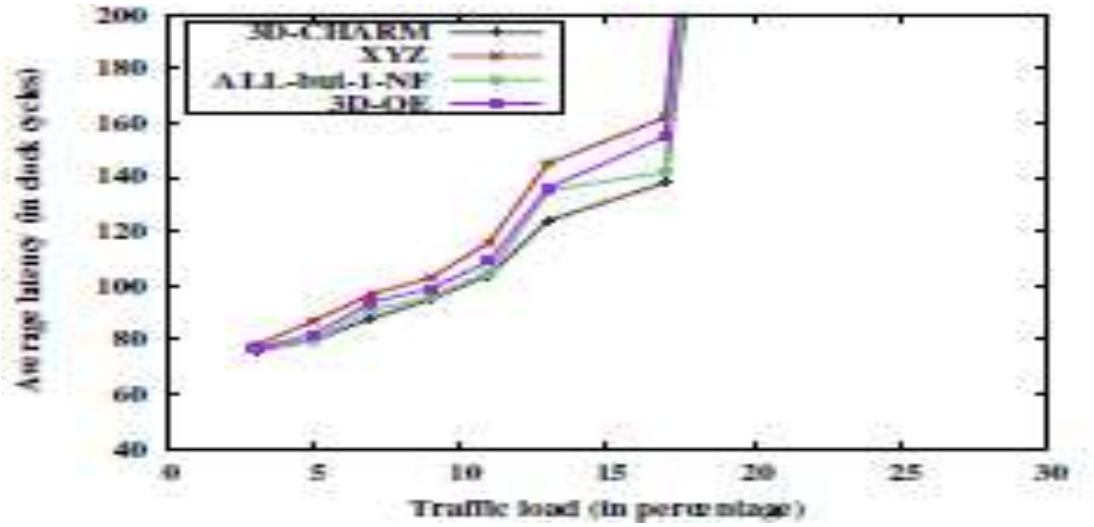


Figure 4.10 : Average latency in 6 X 6 X 6 mesh at different loads under transpose traffic

### 4.3.2 Performance Evaluation

Uniform traffic Profile Under uniform (random) traffic profile, a node sends several packets to every other node in the network with same probability using uniform probability distribution. The load latency graph for uniform traffic model is presented in Figures 4.5 and 4.6. As shown, all algorithms exhibit similar average latency at lower traffic loads for both 4 4 4 and 6 6 6 mesh NoCs. But with increased packet injection rate, it is observed that the dimension order routing (XYZ) performs much better than all other adaptive routing methods as expected. XYZ incorporates relatively long term and more global information about

uniform traffic load characteristic [7]. Since XYZ routes packets rst along X dimension and then in the other dimensions, it distributes packets as evenly as possible throughout the network in the long-term. On the other hand, adaptive schemes select output channels using local short-term information about the network. This selection decision for packets is advantageous only in immediate future and may tend to create \hot spots" region for other packets. In fact, these channel selections hinder the global long-term evenness of random traffic profile by creating zigzag routes. This results into increase contention and reduction in performance at higher packet injection rates. We observe that 3D-CHARM performs better than other adaptive routing methods at higher traffic loads.

Hot-spot traffic Profile Hot-spot is considered a more realistic traffic profile, because in most applications, tasks communicate frequently with small subset of tasks (memory nodes, I/O resources). Under this traffic pattern, some nodes are appointed as hot-spot nodes, which receive some additional hot-spot traffic besides their normal uniform traffic. For simulation, we set nodes 10, 23 and 38 as hot-spot nodes with 0:4 probability of getting additional traffic. Figures 4.7 and 4.8 show average latency under this model for both 4 4 4 and 6 6 6 mesh NoCs. It can be observed clearly that XYZ, in contrast with the random traffic profile, has a higher average latency than the three adaptive algorithms that can cope with congestion better. Because, when multiple traffic ows are oriented towards a small subset of \hot spot" nodes, a non-adaptive XYZ router will be compelled to forward them towards the same output direction, thus saturating the virtual channel queues. On the other hand, adaptive algorithms can direct packets, destined for same destination, to different output channels. It can also be observed that due to higher adaptiveness (both minimal and non-minimal), 3D-CHARM scheme achieves better average latency than other adaptive algorithms by avoiding \hot spots". 3D-CHARM method leads to smaller average latencies because it can more evenly distribute traffic in a congested network using additional paths both minimal and non-minimal than other routing algorithms.

Transpose traffic Profile Under transpose traffic profile, a node at position (X, Y , Z) only sends packets to another node at position (n 1 X, n 1 Y , n 1 Z), for a n n n 3D mesh. This traffic profile simulates the concept of transposing a matrix. This traffic profile results into a non-uniform distribution of traffic with heavy traffic ows for the central nodes creating network \hot spots". As shown by results in Figures 4.9 and 4.10, it is observed that at lower traffic loads, the routing algorithms behave similarly as \hot spots" are not created. But at higher injection rate, \hot spots" are created in center of the mesh. Again, 3D-CHARM method leads to lower average latencies because of its higher path diversity than other

adaptive algorithms. 3D-CHARM results into more evenly distributed traffic using additional paths (minimal and non-minimal).

Application traffic Profile To evaluate proposed work in a more realistic scenario, we con-sider E3S benchmark suite. We select four application suites automotive/industrial, networking, consumer and o ce-automation. This selection is intended to represent various applications used in the real-time embedded systems. Each application suite is represented by .tg le. We gen-erated communication task graphs for each .tg le by parsing it. A communication task graph represents communication pattern and communication volume among tasks. We assign each task to core (processor) using minimum execution time scheduler that executes it in the fastest time. Task mapping strongly depends on particular application traffic. A random mapping algorithm is used to compute the locations of cores within NoC to enable honest and intuitive comparisons among routing algorithms. We executed routing algorithms several times using random mapping and the average of simulation results is used. Figure 4.12 shows average packet latency normal-ized to XYZ routing. 3D-CHARM provides lower latency than other methods across all four application suites. 3D-CHARM shows the greatest performance gain on consumer application traces. Average performance gain of 3D-CHARM is up to 28% across all selected benchmarks vs XYZ and 11% vs other adaptive algorithms.

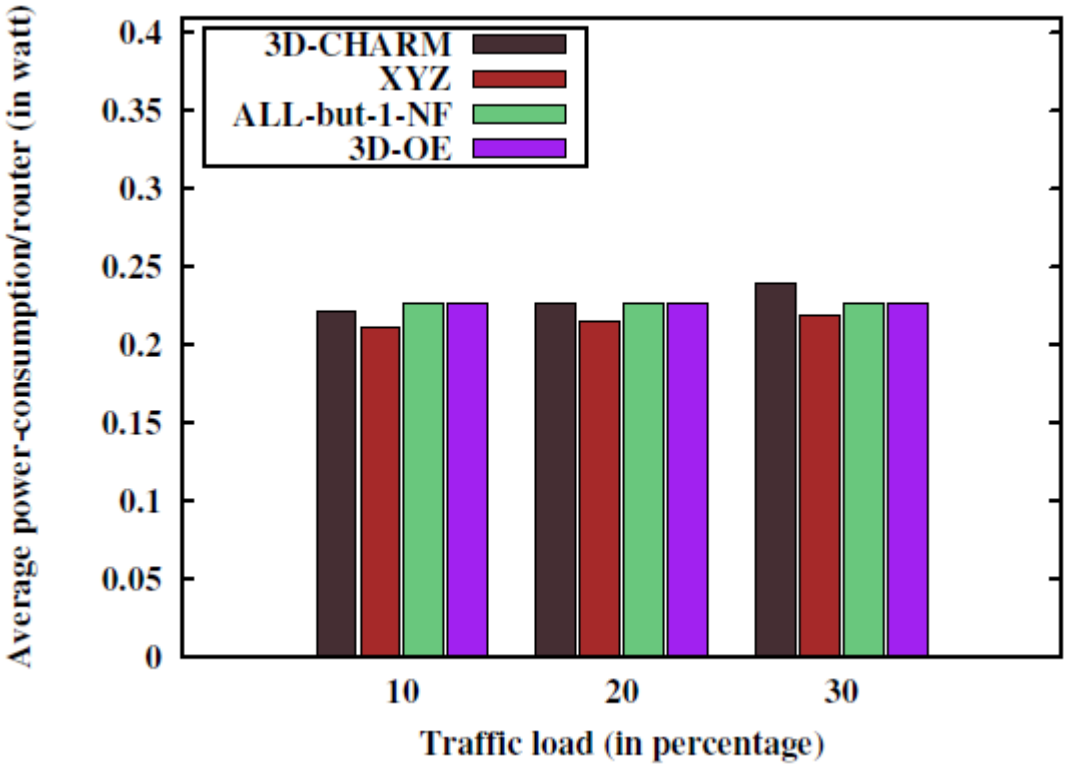


Figure 4.11: Power consumption results under hot-spot Traffic

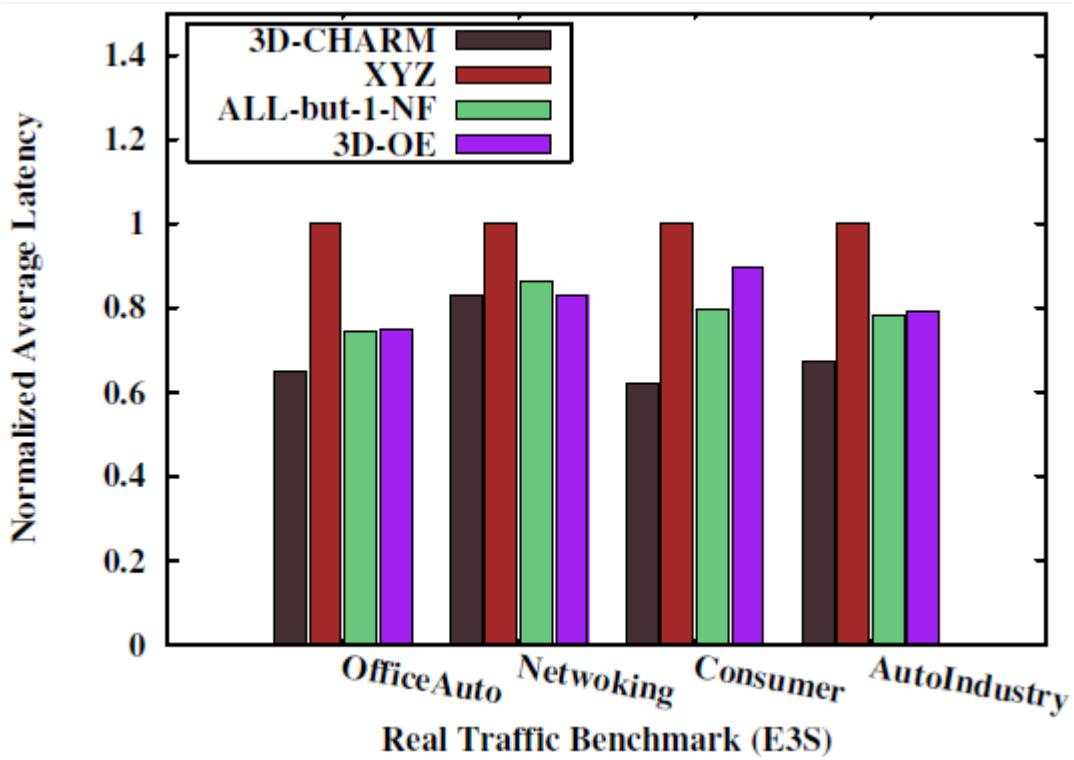


Figure 4.12: Performance for application traces

### 4.3.3 Power Estimation

We deploy an existing NoC power estimation tool ORION [48], that is integrated with NoC simulator [47]. It estimates total power consumption of a router into various sub-components: input buffers, router control logic (arbiter and crossbar) traversal and channels. Figure 4.11 illustrates average power consumption for hotspot traffic with different traffic loads. It can be observed that XYZ consumes less power for all traffic load. Because, it always routes packets through minimal paths. At lower traffic loads, 3D-CHARM perform better than other adaptive routing methods. It uses minimal paths due to small \hot spots" creation at lower traffic load. However, other adaptive methods perform better than 3D-CHARM at higher traffic loads. Proposed method uses non-minimal paths to alleviate congestion, which causes increase in hop count.

# Chapter 5

# Distributed Adaptive Routing for Spidergon NoC

## 5.1 Introduction

Spidergon is a popular NoC (Network-On-Chip) developed to realize cost effective multi-processor SoC (MPSoC) using a fixed optimized topology [35]. Increasing diversity of applications, quality of service requirements and deterministic routing schemes inhibit the performance by creating congestion bottlenecks. Proposed routing algorithm exploits the path diversity of spidergon NoC and selects the optimal path on the basis of congestion level (CL). CL depicts current traffic conditions. Proposed scheme is deadlock free and uses only one extra virtual channel to achieve improved communication. Proposed scheme is compared with native deterministic rout-ing schemes of spidergon NoC i.e. aFirst and aLast. Results demonstrate that the our algorithm distributes traffic evenly while reducing hot-spots. Considerable performance improvement is achieved on modified router architecture.

Technology advancement and shrinking size of transistor has increased the number of IP com-ponents on chip. With grflowing complexity, design paradigm is shifting toward multiple core on single chip instead of a single complex processing element [49]. Therefor focus has been shifted from computation to communication since area, performance and power consumption of SoC mainly depends on underlying interconnect architecture [1]. Network-on-Chip [50] [3] [4], a net-work based approach to interconnect all components of SoC has been proposed as an alternative to overcome drawbacks of bus and point to point based classical interconnects. NoC comprises of routers, links and network interface. Topology de nes the way routers are connected with each other physically using links, networks interface implements protocol to connects ip-cores and routers.

Regular topologies such as 2D-mesh o er good theoretical metrics but cannot be exploited due to nature of traffic in Multi-core SoC applications. But spidergon topology developed by ST Microelectronics provides good trade off between theoretical performance and implementation cost [35]. In spidergon an even number of nodes are connected in bidirectional ring with an extra link connected to diagonally opposite node. Spidergon has smaller number of edges and competitive network diameter as compared to 2D-mesh and fat-

tree for up to 60 nodes. For larger number of nodes, aggregation and hierarchy reduce network diameter and improves performance.

Spidergon is regular, point-to-point and constant degree vertex-symmetric network. Hence uni-formity and homogeneity o er simple and identical router implementation by reducing design and complexity. Moreover, the spidergon topology translates easily into a low-cost practical lay-out. Fig 5.1 is a possible equivalent planar representation in which physical connection between nodes only need to cross at one point in the chip [51]. In spite of homogeneous spidergon scheme, heterogeneity is introduced due to different size and aspect ratio of IPs.

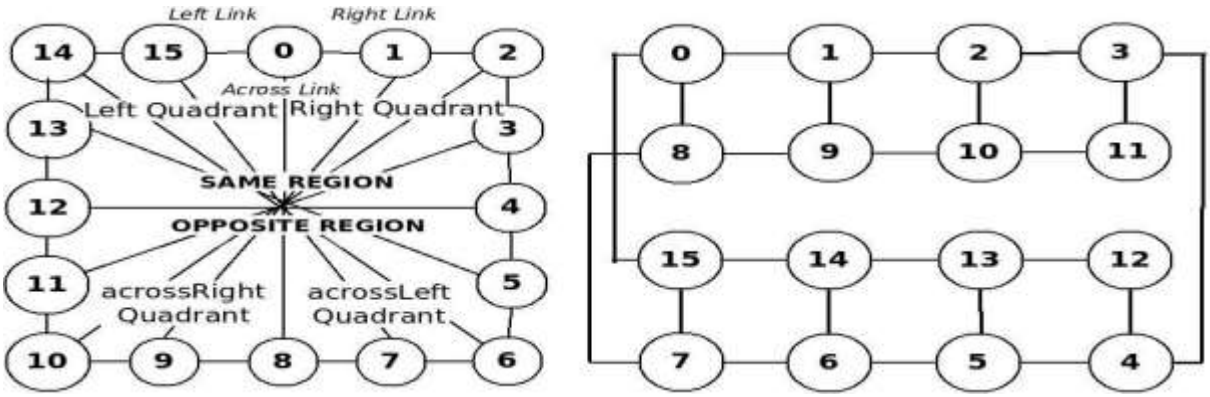


Figure 5.1: 16 Node Spidergon Topology and Connection Layout

Along with topology, choice of routing algorithm greatly a ect the performance of NoC. De-terministic routing algorithms are simple to implement but they do not consider the current network status and always generate a x path for a given pair of nodes. On contrary, adaptive routing algorithms provide more routing path diversity and distribute the traffic more evenly. Routing schemes proposed for spidergon topology are deterministic, source based and shortest path. In this paper we propose a novel adaptive routing algorithm for spidergon NoC architec-tures. Proposed routing is minimal and distributed in nature. It removes restriction on choice of across link. It allows to take across link at any intermediate node depending upon current network status instead of xing at rst or last and hence adding adaptivity to routing scheme.

## 5.2 Proposed Work

### 5.2.2 Motivation: Path Diversity in spidergon

Path is ordered set of channels between any two node. A path is minimal if number of channel in that path is minimum. If more than one minimal path exit between a pair of node, then topology become more robust. This property of topology is called as path diversity [2]. Spidergon exhibits strong path diversity. Our proposed routing scheme explore this property of spidergon to induce adaptivity in routing.

This section presents a brief analysis of routing path diversities available in spidergon topology and paths used by deterministic routing schemes (aFirst, aLast). For all of our analysis, we have chosen source destination pairs which are on opposite side of the ring, because for all other source destination pairs only one minimal path is available which is the one along the ring either in clockwise or anticlockwise as shown in Figure 5.1. For comparing path diversities of aFirst and spidergon topology, we have considered node 0 as source tile and all the nodes which are on opposite side of ring as destination node (node 5,6,7,8,9,10,11). As shown in Table 5.1, aFirst provides a single path for each destinations. On the other hand, spidergon topology provides almost double paths as compared to aFirst routing scheme except for the one shown in row 1 for which a single minimal path is available.

Table 5.1: Routing Path Diversity of aFirst and Spidergon Topology

| D | No. of available path in aFirst, route | No. of available path in spidergon, routes |
|---|---|---|
| 8 | 1,0-8 | 1,0-8 |
| 7 | 1,0-8-7 | 2,0-8-7,0-15-7 |
| 6 | 1,0-8-7-6 | 3,0-8-7-6,0-15-7-6,0-15-14-6 |
| 5 | 1,0-8-7-6-5 | 4,0-8-7-6-5,0-15-7-6-5,0-15-14-6-5,0-15-14-13-5 |
| 9 | 1,0-8-9 | 2,0-8-9,0-1-9 |
| 10 | 1,0-8-9-10 | 3,0-8-9-10,0-1-9-10,0-1-2-10 |
| 11 | 1,0-8-9-10-11 | 4,0-8-9-10-11,0-1-9-10-11,0-1-2-10-11,0-1-2-3-11 |

For comparing path diversities with aLast, we have considered node 0 as destination node and all the nodes which are on opposite side of ring as source node (node

5,6,7,8,9,10,11). As shown in Table 5.2, aLast provides a single path for all source destination pairs. On the other hand, topology provides almost double paths as compared to aLast routing scheme except for the one shown in row 1 for which a single minimal path is available.

Table 5.2: Routing Path Diversity of aLast and Spidergon Topology

| S | No. of available path in aLast, route | No. of available paths in spidergon, routes |
|---|---|---|
| 8 | 1,8-0 | 1,8-0 |
| 7 | 1,7-8-0 | 2,7-8-0,7-15-0 |
| 6 | 1,6-7-8-0 | 3,6-7-8-0,6-7-15-0,6-14-15-0 |
| 5 | 1,5-6-7-8-0 | 4,5-6-7-8-0,5-6-7-15-0,5-6-14-15-0,5-13-14-15-0 |
| 9 | 1,9-8-0 | 2,9-8-0,9-1-0 |
| 10 | 1,10-9-8-0 | 3,10-9-8-0,10-9-1-0,10-2-1-0 |
| 11 | 1,11-10-9-8-0 | 4,11-10-9-8-0,11-10-9-1-0,11-10-2-1-0,11-3-2-1-0 |

As per traffic pattern aEqualized tags certain nodes as aFirst and others as aLast. As a conclu-sion, aFirst and aLast and also aEqualized (combines aFirst and aLast) limit the path diversity provided by spidergon topology. In this paper we have exploited this path diversity provided by spidergon topology. In our scheme packets can take different routes depending on the current status of each link. As cyclic dependency may introduce deadlocks, two virtual channels are used for each link.

### 5.2.2 Proposed Routing Algorithm

As show in Figure 5.1 each node in spidergon is identi ed by a positional numerical value. A network of size N will have have id as 0, 1, 2, 3.....N-1. Spidergon consists of a bi-directional ring in both clockwise, and anti-clockwise directions. At any node We call clockwise links as Right link and Anticlockwise link as Left link. In addition to these links, each node is cross connected to diagonally opposite nodes, i.e. from node i,0< i < N to node (i + n)modN, we call it across link. Diameter of spidergon is defined as [N/4]. Distance between two tile is number of links in minimal path.

We divide the spidergon network into two region: same region or opposite region in context of any arbitrary node. U is any arbitrary node. A node V is said to be in same region if distance between U and V is less than or equal to diameter otherwise it is said to be in opposite region. We further divide it into four quadrants: left, right, acrossLeft and acrossRight. U is a arbitrary node and node A is connected directly to U using across link. V is said to be in left quadrant of U if it is in same region and is reached using left link. V is said to in right quadrant of U if it is in same region and is reached using right link. V said to in acrossLeft quadrant of U if it is in opposite region and in left quadrant of A. And V is is said to be right quadrant if it is in opposite region and in right quadrant of A.

Congestion level (CL) has been used as a measure of traffic load on a channel and hence depicting current traffic scenario. To keep router architecture simple we used a simple scheme using virtual channel. Congestion level of any link will be considered as high if requested virtual channel at next node in that particular output direction is not free, otherwise it is considered as low.

Algorithm 2 Algorithm: minCong(X,Y)

```
X,Y : directions

CL[ ] : Congestion Level of directions at a router
I/P : X, Y

O/P: X OR Y

if CL[X] _< CL[Y ] then
  return X

else
return Y

end if
```

Algorithm 3 Algorithm: Route

```
CUR : current Node, DST : destination node
C, A, L, R : Core, Across, Left, Right directions
I/P : CUR , DEST
O/P: C OR A OR L OR R

if DST is equal to CUR then
return C
else if DST is in Left Quadrant of CUR then
return L
else if DST is in Right Quadrant of CUR then
return R
else if DST is diagonally opposite of CUR then
return A
else if DST is in acrossLeft Quadrant of CUR then
return minConj(A,L)
else
return minConj(A,R)
end if
```

As per algorithm 3, current node id is compared with destination node id. If destination node id is equal to current node, the packet is routed to core for processing otherwise the quadrant and region of destination corresponding to current node is identified. It checks for quadrant in which destination lies if destination is in same region of current node. As per quadrant it routes the packet to left or right link. Across link is selected when destination node is diagonally opposite. When destination is in opposite region, it compares the congestion level of across link to left or right link as per destination quadrant using algorithm 2. When across link is congested, it moves the packet to left or right link so that it can take across link at any intermediate node whose across link is not congested. Figure 5.2 shows step by step ow of proposed routing algorithm.

Proposed routing scheme utilizes available minimal paths provided by spidergon topology and distributes the traffic accordingly. Packets take different routes depending on the current status of each link.It behaves like aFirst if destination is in same region or in

opposite region of ring and across link is free at rst place and like aLast if destination is in same region or opposite region of ring and across link is busy at all places along the path so it route packets rst along the ring in either clockwise or anticlockwise direction and at last it uses across link to deliver packet to its destination.

### 5.2.3 Deadlock and Livelock Avoidance

To ensure correct functionality of any routing algorithm, it must be made deadlock free. Deadlock is the situation where packets holding some resources (buffer or channel) request for resources held by some other packets in a circular way.Spidergon's deterministic schemes like aFirst and aLast restricts the location of taking across link to only at rst and last place respectively, to avoid circular dependencies between across channel and links along the ring (left or right) leading to deadlock situation. Our scheme does not restricts the location of taking across link to rst and last place, it allows it to take at any place along the minimal path. Furthermore circular dependencies between across link and the links along the ring may introduce deadlock situation like the one shown in Figure 5.3. To avoid these type of circular dependencies we have used two virtual channels (VC), along all the links. Out of two, one virtual channel is used as an escape channel. We forced packets to shift from normal virtual channel to escape virtual channel after taking across link. For example, packet in VC0 will be assigned VC1 after passing through across link, resulting in zero cyclic dependencies as shown in Figure 5.3. For avoiding circular dependencies along the ring channels in left and right directions, we have used multiple dateline VC selection algorithm [2].Livelock is avoided by restricting packets to use only minimal path to reach destination.

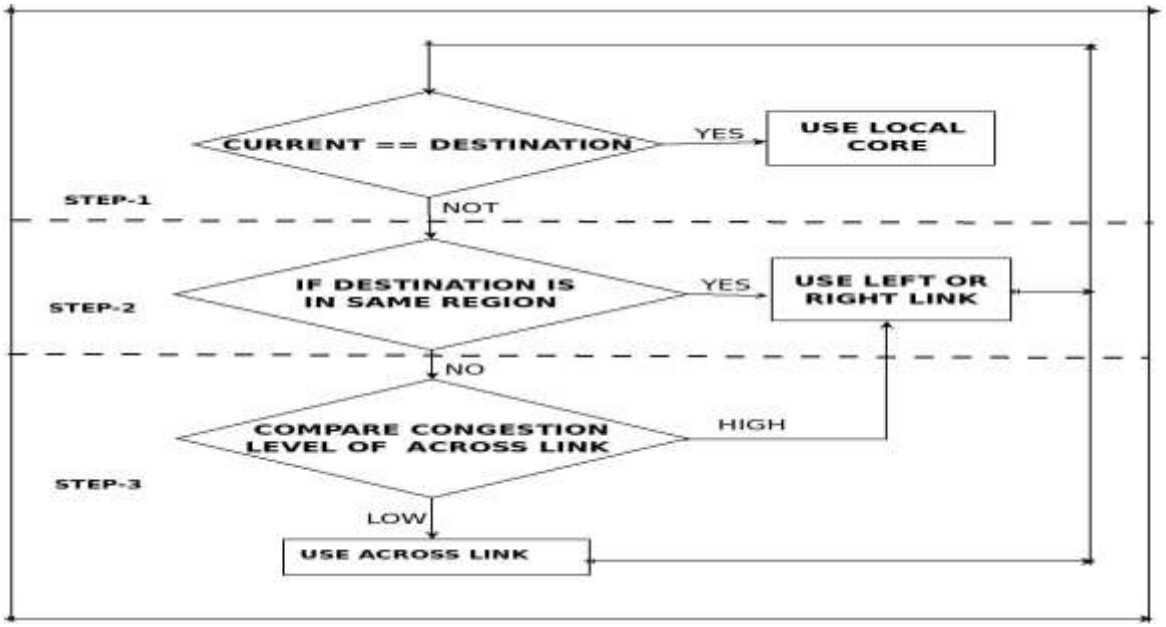


Figure 5.2: Proposed Method Flow

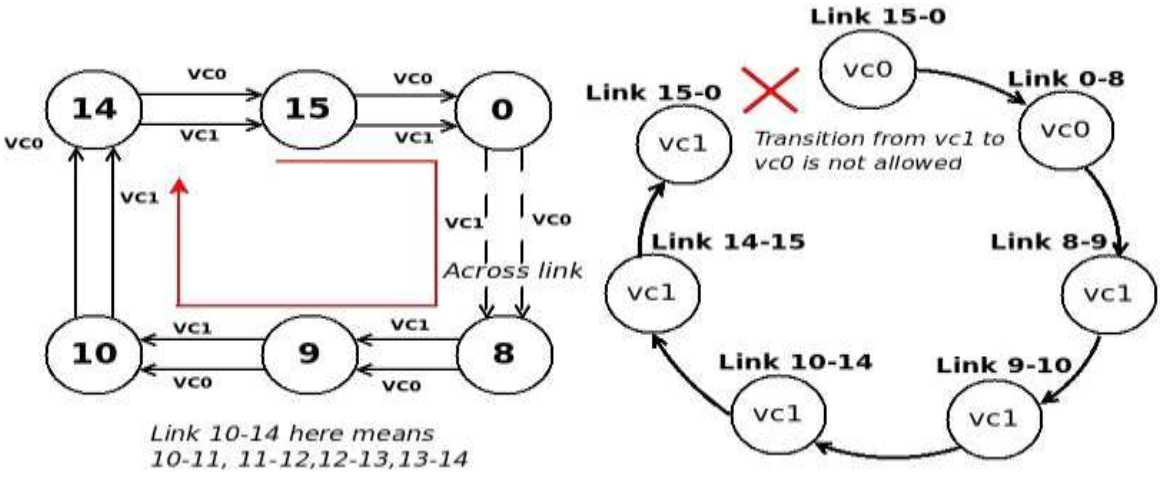


Figure 5.3: Deadlock Avoidance

### 5.2.4 Router Architecture

Figure 5.4 shows modi ed router architecture required for our approach. Two virtual channels are used in each direction. As packets destined for same tile can take either two directions at same time, Across - left or Across - right, we have used two virtual channels in

LOCAL direction to avoid contention for a single queue placed at network interface. It will give performance benefit for cases where node act as a hot-spot node.

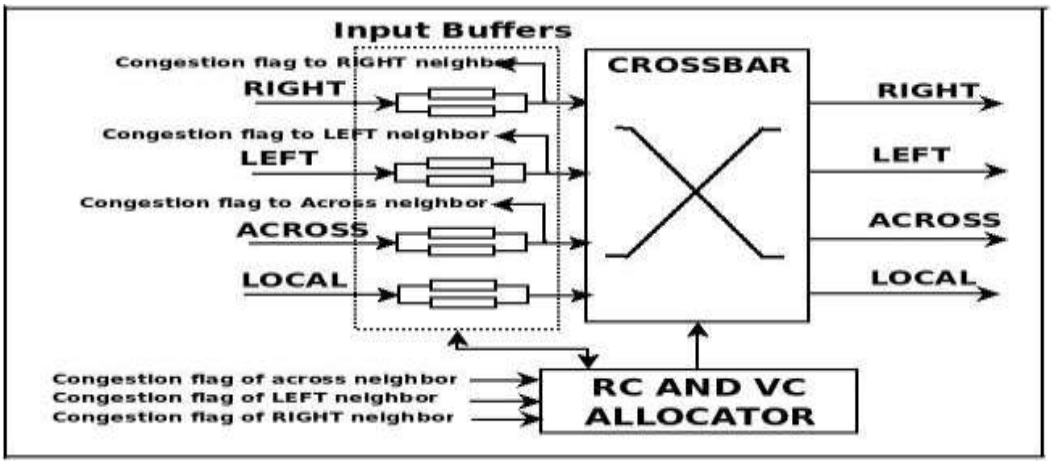


Figure 5.4: Proposed Router Architecture

## 5.3 Analytical Evaluation of Traffic Distribution

In this section we compared distribution of traffic generated by proposed scheme with aFirst and aLast. For comparing aFirst with proposed scheme we have chosen node 0 as source node and nodes which are on opposite side of ring as destination nodes (5, 6, 7, 8, 9, 10, 11). As shown in Figure 5.5, aFirst routing spreads most of the traffic over across link(shown in bold line) resulting in highly congested link and uneven distribution of traffic. On the other side, proposed scheme makes use of all available minimal paths to route the traffic and results in more even distribution of traffic across all links.

For comparing aLast with proposed scheme, source nodes 5, 6, 7, 8, 9, 10 and 11 are sending traffic to one of its opposite side destination, node 0. Similar to previous one, in Figure 5.5, aLast routing spreads most of the traffic over across link(shown in bold line) resulting in highly congested link and uneven distribution of traffic. On the other side, proposed scheme makes use of all available minimal paths to route the traffic and results in more even distribution of traffic across all links.

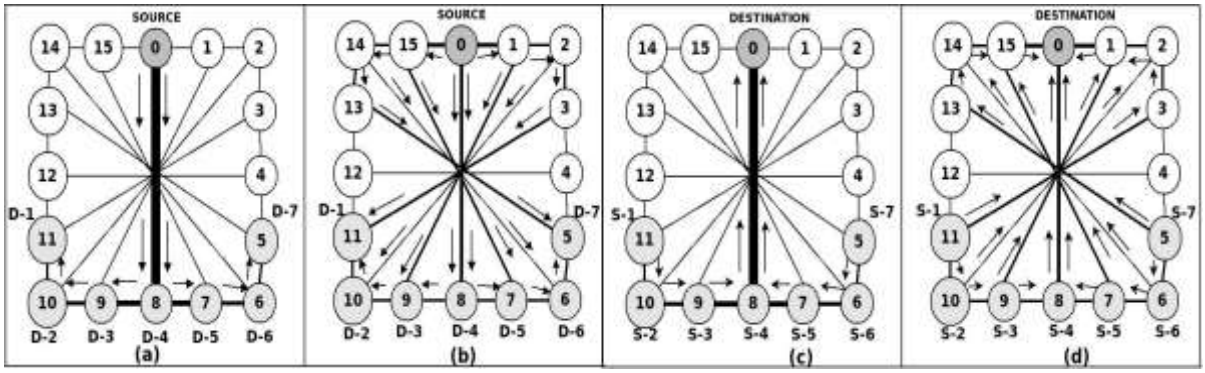


Figure 5.5: Channel Utilization (a)aFirst (b)Proposed (C)aLast (d)Proposed

## 5.4 Experimental Setup and Performance Evaluation

For simulation we have used NIRGAM [52] [47] (NoC Interconnect Routing and Application Modeling). NIRGAM is systemC based, extensible discrete event and cycle accurate simulator. We have used wormhole switching in 16 node spidergon topology. Packet is generated by nodes as constant bitrate (CBR) traffic pattern with packet size of 8 its. In our simulation experiments, Load parameter de nes o ered rate of packet injection but packet generation is also affected by architectural factors such bandwidth and resources availability.

### 5.4.1 Latency and Throughput Analysis

In this set of experiments, performance of network under different routing strategies (aFirst, aLast, proposed) and under uniform, non-uniform traffic patterns, respectively, is evaluated. In uniform traffic pattern, each node generates traffic for all other node with equal probability. traffic is evenly distributed across all links and each link is equally busy. Since all channels are equally congested so no improvement in latency is seen. As shown in Figure 5.6, under uniform traffic distribution average latency of proposed scheme is same as aFirst and aLast routings. However, in non-uniform traffic patterns, each node communicate with few nodes in network more frequently than other nodes, resulting in uneven traffic scenarios.

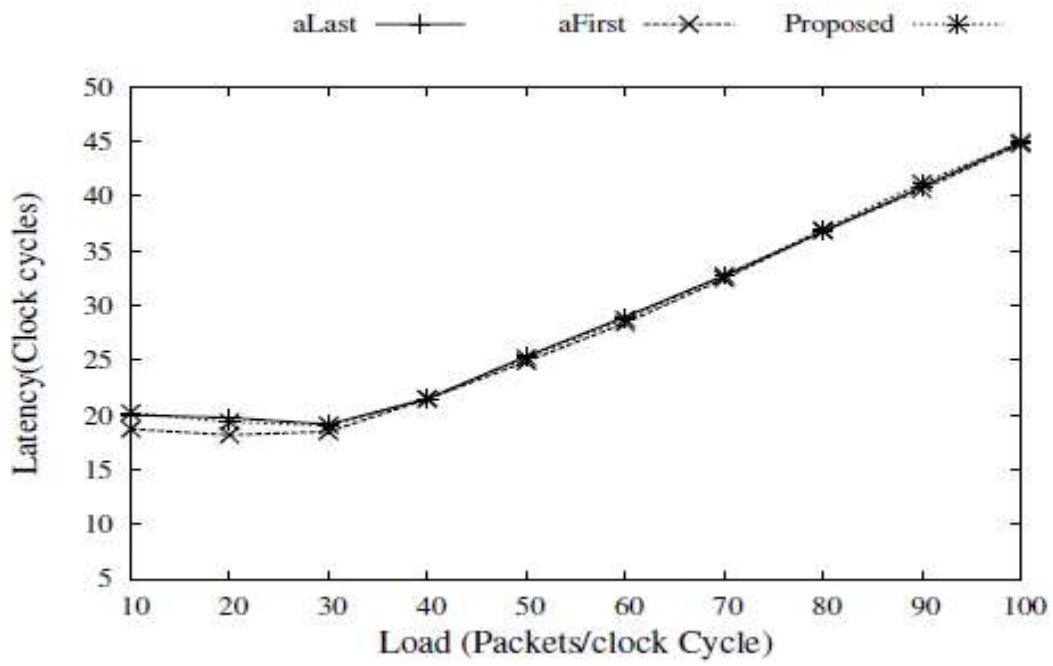


Figure 5.6: Latency Curve for Uniform Traffic

We considered hot-spot traffic for generating uneven traffic scenarios and evaluated performance under the same. In case of single hot-spot node, all nodes are sending packets to single desti-nation. aLast routing generates a bottleneck traffic as n/2 traffic of total traffic, needs to pass

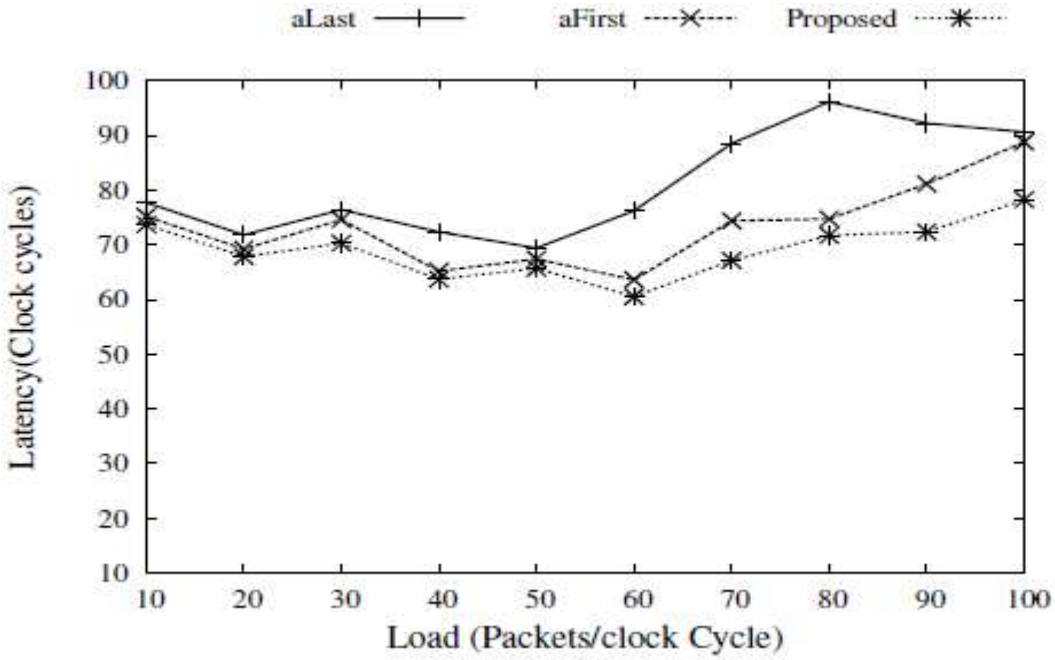


Figure 5.7: Latency Curve for Hot-spot Traffic

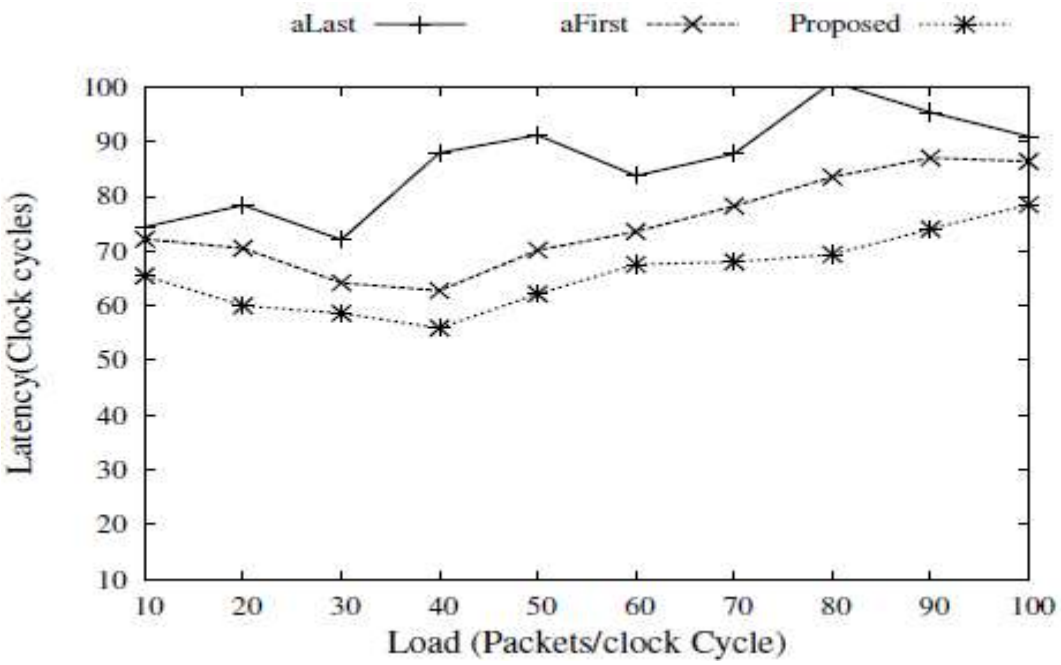


Figure 5.8: Latency Curve for Random Traffic

through a single across link whereas, other links remain unused. On the other side, aFirst, re-moves this bottleneck problem at some extent as all source nodes rst route traffic towards across link. But, in case of congestion it does not adapt as per the current network traffic, and increases traffic on clockwise and anticlockwise channel of opposite side node because of single determinis-tic path between each source destination pair. This scenario could be opposite for cases where a single hot-spot node generates traffic for rest of nodes. Proposed method addressed this problem by distributing traffic across all links. It takes advantage of status of current network traffic and sends traffic towards less congested output ports and results in improved overall average latency as shown in Figure 5.7. As number of hot-spot nodes increases, proposed scheme distributes traffic evenly and clearly outperforms the two routing algorithms as shown in Figure 5.8.

### 5.4.2 Power Analysis

When we evaluated the throughput, we compared throughput achieved using our scheme to a rst throughput and o ered throughput. As shown in Figure 5.9, in case of uniform traffic when more amount of traffic is injected then only we can see some improvement in

throughput owning to runtime network adaptivity. While as shown in g 5.10, unevenness of non-uniform traffic is handled quite efficiently by proposed scheme. Proposed scheme curve is similar to offered throughput curve. Their is quite signi cant improvement in throughput. Power utilization as

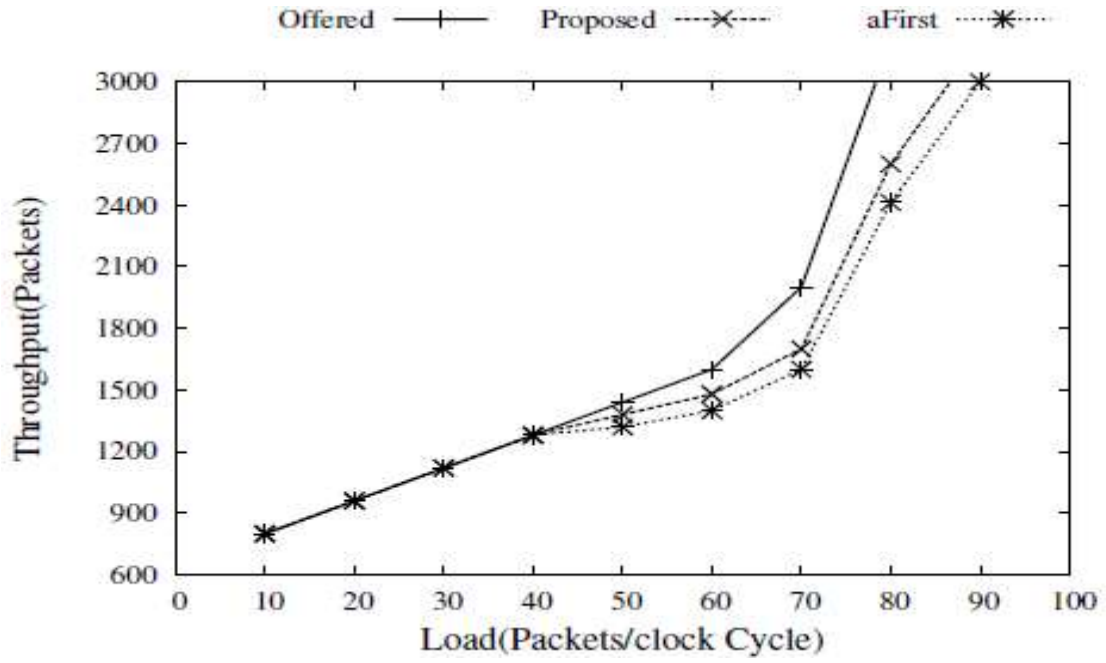


Figure 5.9: Throughput Curve for Uniform Traffic

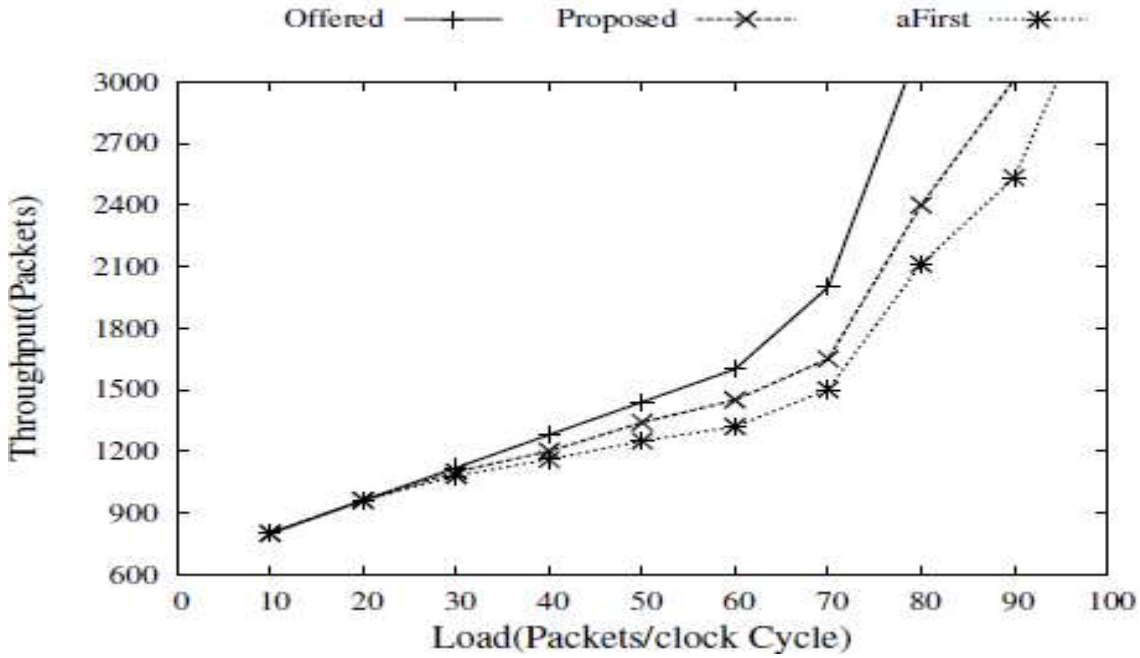


Figure 5.10: Throughput Curve for Non-Uniform Traffic

shown in Figure 5.11 is not affected by proposed routing scheme. Sice it provide minimal path, average hop count remain same and hence average switch traversed which cause maximum power consuption is same.

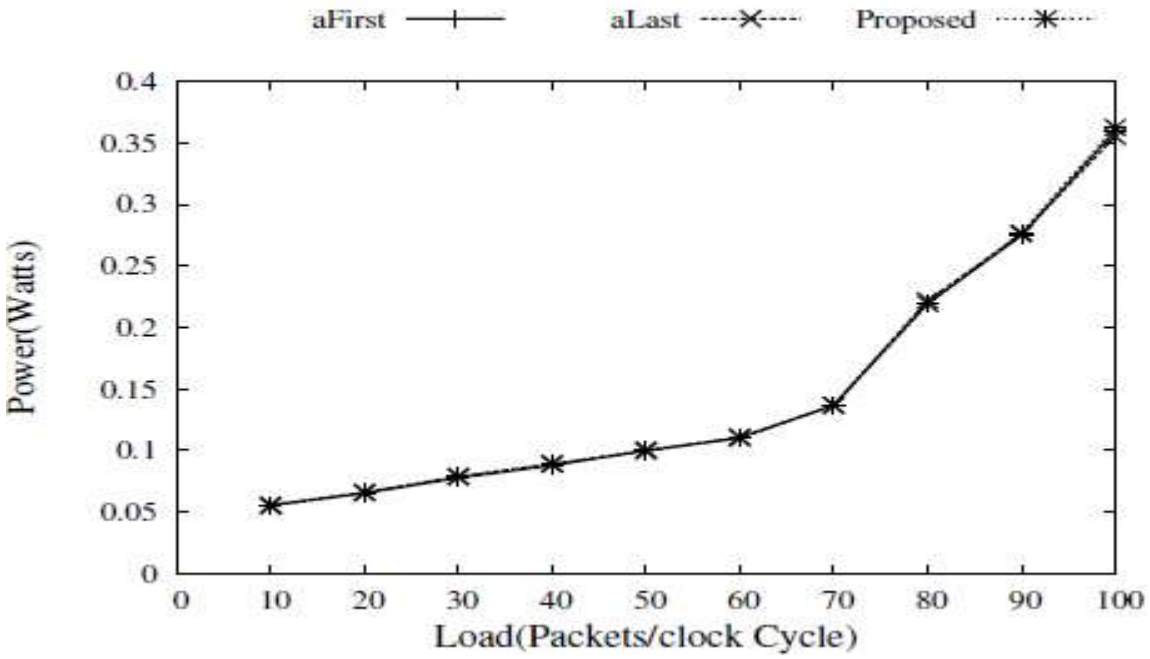


Figure 5.11: Power Curve for Non-Uniform Traffic

# Chapter 6

# Conclusions and Future Work

Table based implementation of distributed routing can be costlier in terms of silicon area and power consumption as compared to combinational logic-based implementation for large irregular NoCs. In this paper, we have proposed an implementation of cost-efficient, Reconfigurable and fault tolerant xy routing algorithm for large irregular 2D mesh without routing tables. With the help of routing and connectivity bits, we are able to omit table-based implementation. Our algorithm is designed to tolerate one or multiple single link faults within 2D mesh. Algorithm uses a few non-minimal paths (reconfigured paths) which improve reliability and fault tolerance of NoC communication. We have used xy and new routing restrictions in non-faulty and faulty regions, respectively, to achieve deadlock freedom without overhead. On the basis of simulation results, we argue that our deadlock avoidance method remains area and power efficinetbecause no additional virtual channels are required. In future work, we intend to extend the method to accommodate any number of faults within mesh topology till graph partition takes place for the given topology.

Network congestion may result into increased communication latency and power consumption, thus can severely degrade performance of on chip networks. Acyclic channel dependency graph for deadlock freedom results into lower degree of adaptiveness. In this paper, we presented a novel and improved turn model for 2D and 3D meshes. We also proposed a low cost, highly adaptive routing algorithm (CHARM) to mitigate congestion based on proposed turn model. CHARM provides higher degree of adaptiveness by allflowing cycles in channel dependency graph. It uses only one additional virtual channel along each of Y and Z dimensions. It can use minimal or non-minimal routes between source and destination nodes depending on congestion status. A packet is routed along the non-minimal paths only when minimal paths are congested at the neighboring nodes. Deadlock freedom of CHARM is ensured using Duato's theory [46]. Results, on synthetic and real traffic scenarios, show that CHARM improves network performance by routing packets through non-congested areas. Our future work is focused on incorporating global congestion awareness with additional hardware and extending proposed method for n-dimensional meshes. Further developments also include TSV modeling, area analysis and thermal-aware extension of CHARM

In this paper, we proposed a novel adaptive routing algorithm for spidergon NoC. Proposed routing scheme is minimal, adaptive and distributed in nature. It uses the current network traffic status and distributes traffic across all links evenly by taking advantage of path diversity available in spidergon. For deadlock avoidance we used 2 virtual channels and modi ed standard router of spidergon architecture by adding support for each virtual channel at each input port to fully exploit the potential of our scheme. In case of non-uniform traffic proposed scheme shows considerable improvement in throughput and latency as compared to deterministic routing. In case of uniform traffic and low congestion in network it performs at par deterministic scheme. In future,we would extended scheme to add support for fault tolerance by implementing it with logic based distributed routing(LBDR).

# Appendix A

## Performance Evaluation of Spidergon and Fattree

## A.1 Introduction

NoC architectures are defined by three parameters i.e. topology, switching logic and routing algorithm. To increase the performance of SoC, effective communication among the cores to utilize the shared resources should be increased which depends on connection pattern of nodes. This connection pattern of nodes is defined as Network topology. Optimal topology selection is the rst and foremost requirement of NoC architecture which in turn also a ects the perfor-mance of switching and routing logic. Thus topology is characterized by two main factors i. e cost and performance. Performance of a topology is measured in terms of bandwidth, latency and throughput. Choice of topology is also important because once con gured it cannot be reconfigured in non-Reconfigurable NoC [?].

A large number of toxicologies have been defined for NoC such as mesh, torus, ring, spidergon, fattree etc. Among these topologies we are comparing here three topologies that are mesh which is simple to design, spidergon which is a ring based topology with additional across links and an indirect topology i.e. fattree.

Topology de nes the pattern in which nodes are connected to each other. A topology for NoC can be of various types: Direct or indirect, Regular or irregular. In direct topologies each node is connected to at least one core(IP/PE). Example of Direct topologies are mesh, torus, spidergon. In indirect NoC topologies we have some set of nodes that are not connected to any core(IP/PE) and performs only network operation. Examples are Fattree, butter y etc. A regular topology is defined in form of some specific structure such as rings, mesh, hypercubes etc where as irregular topology does not possess any regular structure.

Some of topologies that are taken into account are mesh, spidergon and fattree. A topology is selected in order to maximize performance and minimize cost. Number of parameters that a ect selection of topology are degree of nodes present, network diameter, regularity and scalability.

A mesh is a network consisting of m rows and n columns. The routers are easily depicted as x-y coordinates in a mesh. Corner tile in mesh network has two neighbors, border tile has three neighbors and others have four neighbors tiles. Thus mesh networks are easy to

expand and multiple paths exist between a pair of nodes that makes it tolerant to failure. The limitations of mesh networks is it has a long diameter which in turn increases its communication latency. Mesh networks also possess some level of irregularity in corner and edge tiles.

Torus topology is an extension to mesh architecture obtained by adding direct connections to two end nodes in same row or column. These wrap-around channels poses advantage that latency is reduced in comparison to mesh architecture.

Spidergon architecture is similar to ring topology having unidirectional links to the neighboring nodes in both left and right direction and additionally an across link between opposite nodes. In spidergon topology every node is indefixed by a number between 0 to N-1. An arbitrary node is assigned label 0 and the label of other nodes is incremented by one as we move clockwise. And the across network nodes are given label of the node with the lower index plus N. Each node in the network, xi (0!= i! N ), is directly connected to node xj for j = (i + 1) mod N, j = (i -1) mod N and j = (i + N/2 ) mod N. The advantages of spidergon topology are lower node degree in comparison to mesh networks, vertex symmetry i. e. topology appears same from any node and edge-transitivity

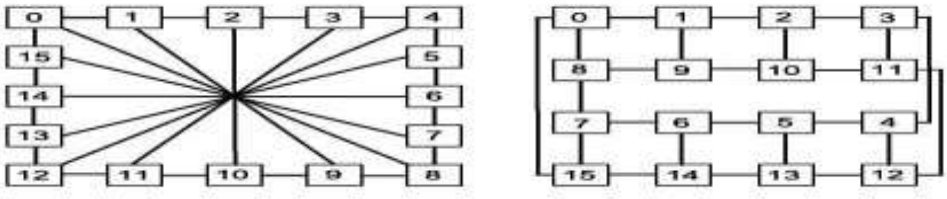


Figure A.1: Spidergon Topology and Floor-plan

Fattree is an indirect NoC topology in which processors are connected at the bottom layer. The key feature for a fat-tree is that for any switch, the number of links going down is equal to the number of links going up to the higher level. This allows all nodes connected at the bottom layer to have full bandwidth i. e., any node can communicate with any other node while having 100 percent bandwidth available for this communication. Butter y fat-tree is a type of unidirectional multistage network. A k-ary n-fly is implemented by using unidirectional switches of radix k organized in n stages. A k-ary n-y is also able to connect N = k $\wedge^{n}$ processing nodes using nk

$\wedge^{n-1}$ switches and nk $\wedge^{n}$ k unidirectional links.

## A.2 Analytical Comparison

NoC topologies can be defined by parameters like Node degree, Bisection bandwidth, Network diameter etc and performance can be evaluated by parameters like Latency, Throughput. Degree: The node degree is defined as the maximum number of physical links originating from a node.

Network diameter(ND): ND is defined as the maximum shortest path length between any pair

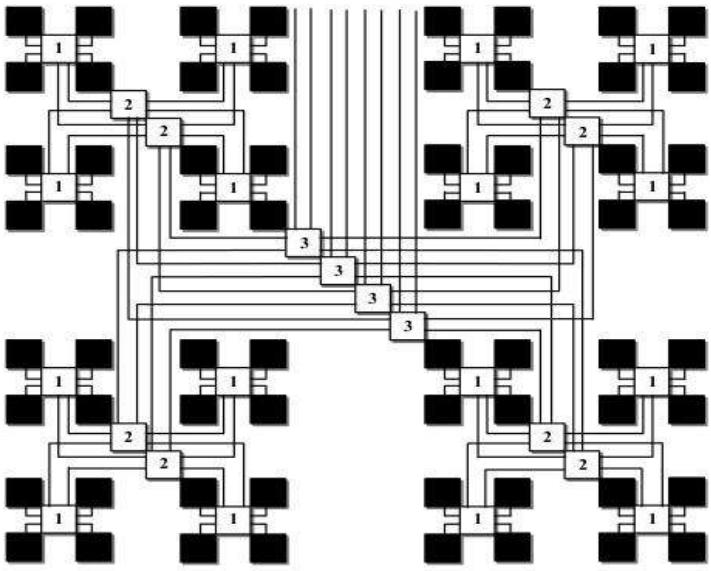

**Figure 6. The H-layout of the 4-ary butterfly Fat Tree with tree switch stages and 64 processor leaf nodes.**

Figure A.2: ButterFly Fat-Tree Floor Plan

of nodes in the topology. Network diameter of any topology should be small so as to minimize latency.

Bisection width: The Bisection Width (BW) of a network is defined as the minimum number of channels/links that must be removed to divide the network into two equal sized

networks. Throughput: Throughput of a network is the data rate in bits per second that the network accepts per input port.

Latency: Latency of a network is time required for a packet to traverse the network, from the time the head of the packet arrives at the input port to the time tail of the packet departs the output port.

Analytical comparison of the topologies considered is done for the evaluation parameters defined above. The size of all topologies is fixed i. e, N=16 where N is number of nodes.

Table A.1: Analytical Comparison Of 16 Node Topologies

| Topology | Degree | Diameter | Bisection Count |
|---|---|---|---|
| | | | |
| mesh | 2-4 | 6 | 4 |
| | | | |
| torus | 4 | 4 | 8 |
| | | | |
| tree | 3 | 8 | 1 |
| | | | |
| fat tree | 1-3 | 4 | 4 |
| | | | |
| spidergon | 3 | 4 | 4 |
| | | | |

The above analytical comparison shows that the spidergon topology has a lower network diame-ter than 2D mesh and in indirect topologies fat-tree has lower network diameter than tree. The network diameter is the deciding factor for communication latency.

## A.3 Performance Evaluation and Analysis On NIRGAM

In the follflowing section we will analyze NoC architecture performance for the defined topologies. Routing algorithm adopted by each topology is Dimension-Order Routing. In 2D -Mesh and 2D-Torus we used X-Y Dimension-Order Routing in which packet from the source node is rst transferred along the X direction and then along the Y direction to

reach to destination. In spidergon topology we adopted Across-First Routing algorithm in which if the destination for packet is at distance D>N/4 on the ring then the across link is traversed rst, to reach the opposite node and clockwise or anti-clockwise direction is taken, depending on the destination defined. In fattree and tree based topology we use common ancestor dimension-order routing.

**Simulation and Results Analysis**

For simulation we have used NIRGAM[52](NoC Interconnect Routing and Application Model-ing), a cycle accurate simulator which captures various aspects of NoC design such application mapping, topology selection, routing algorithms, switching techniques and ow control mecha-nism relevant to hardware as it is developed in systemC. For performing topology comparison we have used dimensional-order routing algorithms corresponding to topology. We have used vir-tual channel based router with credit based back pressure ow control mechanism and wormhole switching. Parameters which we have used for comparison of topologies are worst case latency and power consumption which depends upon dimension of topology, throughput required and router architecture.

Experimental Setup We have mapped CBR(Constant Bit-rate) and bursty synthetic traffic generator which generate throughput of 1GBps on 16 node and 32 node interconnect con gured according to topology selected. Routing method used are XY routing for torus and mesh, across rst routing for spidergon and common ancestor routing for tree and fattree. Latency is calculated as average time taken by packet to travel from source to destination. Power is computed as sum of router power and link power. These calculations depend upon number of links traversed, number of switch traversed, buffer read, buffer write, vc read, vc write, itsize and clock frequency.

$$latency = \frac{\Sigma(FlitRecevdTime - FlitGenTime)}{\Sigma FlitCount}$$

$$AvgPowerconsuption = \sum(RouterPower + linkPower)$$

The source and destination are set such that worst case behavior of the topology can be captured. In 4X4 Mesh, source node is 0 and destination is 15 so as to get maximum hop count. The corresponding source and destination for each topology is shown infig 3.

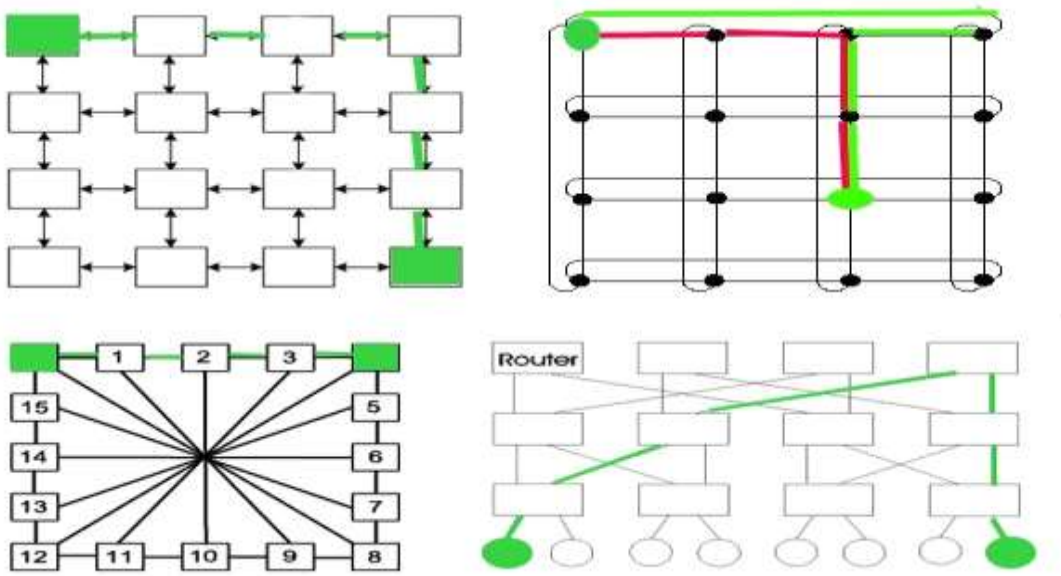


Figure A.3: Sourse Destination Path

Result Analysis Latency and power calculated for CBR traffic is shown in table II and Bursty traffic is shown in table III. Simulation results shows that spidergon and torus are best suited for smaller topologies. Spidergon has lower diameter and regular structure hence it consumes low power and has low latency. Scalability of torus is limited by wrap-around links as longer links leads to signal degradation and weak signal leads to transient faults. Despite relatively large diameter and average distance, regular lower dimension mesh is most cost effective due to its regularity, scalability and uniform wire density. However 2-D mesh is not appropriate for irregular instruction level parallelism in digital signal processing and data ow analysis. Mesh takes advantage of locality by placing core to each router. In case of indirect topologies, tree and fattree gives comparable results with dimension-order routing as it is deterministic and gives same path for a destination. Fat-tree has additional advantage of reduced links to handle fault tolerance and provide path diversity. Fat-tree is best suited for large number of cores as latency in 32 node topology for fat-tree is lesser than mesh. It is because diameter of fattree grows logarithmically as compared to mesh which grows linearly.

Table A.2: CBR Traffic Simulation Table

| | 16 Nodes | | 32 node | |
|---|---|---|---|---|
| | | | | |

| Topology | Latency (in clock cycles) | Power (in Watt) | latency (in clock cycles) | Power (in Watt) |
|---|---|---|---|---|
| Spidergon | 31 | 0.008 | 51 | 0.017 |
| Torus | 35 | 0.010 | 51 | 0.014 |
| Mesh | 51 | 0.014 | 83 | 0.022 |
| Tree | 51 | 0.010 | 67 | 0.014 |
| FatTree | 53 | 0.014 | 69 | 0.018 |

Table A.3: Bursty Traffic Simulation Table

| Topology | 16 Nodes Latency (in clock cycles) | Power (in Watt) | 32 node latency (in clock cycles) | Power (in Watt) |
|---|---|---|---|---|
| Spidergon | 30 | 0.008 | 61 | 0.017 |
| Torus | 37 | 0.009 | 52 | 0.013 |
| Mesh | 52 | 0.013 | 84 | 0.021 |
| Tree | 52 | 0.009 | 68 | 0.013 |
| FatTree | 54 | 0.013 | 70 | 0.019 |

In this section comparison of few topologies is presented using dimension-order routing algorithm. Simulation results demonstrate that despite being simple and regular, mesh

topology is not scalable for larger dimensions. Spidergon topology performs well for lower dimensions but latency increases linearly with increasing number of nodes whereas due to logarithmic increase in network diameter fattree shows better performance for larger number of nodes. Our future work includes evaluation of performance of these topologies using adaptive routing algorithm, fault tolerance capabilities and behavior under congestion.

# Appendix B

# Congestion Aware Router Architecture

## B.1 Congestion in Network On Chip

Along with various architectural parameters such as topology, routing algorithm, switching tech-niques, run time performance of the network-on-chip is affected by ow control, buffer manage-ment and congestion management policy. Congestion is state of NoC when services to a request-ing entity is denied for prolong period of time. The denial of service arise due to unavailability of resources. Since in NoC we have shared resources such as channel, buffers, switch and arbiter. A proper allocation of these resources can be means to handle congestion.Flow control is controlled by the receiving side. It ensures that the sender only sends what the receiver can handle. It is a synchronization protocol. Congestion Control ensures that everyone across a network has a "fair" amount of access to network resources, at any given time. Congestion Control also ensures efficinetUtilization of resources.

Congestion Detection Most interesting challenge in congestion handling is detecting conges-tion. NoC is communication architecture in Muti-core architecture. Application running on cores de nes the nature and pattern of traffic. Owning to large number of applications it is hard to de ne a pattern of traffic in network on chip. Secondly the resources are limited o chip owning to power consumption. Since most application's traffic generation pattern is undefined. Congestion tends to be momentary in nature. This is not a deadlock situation where resources are blocked rather resources are being used by some other agent. Hence complexity of detecting congestion is quite tedious.

NoC have multiple layer of abstraction for utilization of resources. First thing is at what level should we measure and store the congestion state. congestion state or Information means what is the current stage of congestion at a particular instant of time. We can explain it as either it is congested or not congested. But two capture nal details of congestion we can need to categorize congestion level. It is just congested , congested , totally congested or completely congested.

- Single Level
- Multi Level

Level of Congestion Detection

- Local: In this case congestion state is local to router. A router have information about its output ports. It knows which output port is congested or not. There are two sub-level in case of local congestion. We can have information specific to port or virtual channel. Once most of the port is congested we say that router is congested.
  - Virtual Channel
  - Physical Channel
  - Router

- Global: In global congestion approach we store the congestion information of each router. Each node has

  - Centralized with shared control Network
  - Centralized with Dedicated control Network
  - Distributed with shared control Network
  - Distributed with dedicated control Network Regional: In case of regional congestion

**Congestion Detection**

- Instantaneus
  - Per Cycle
  - After Inverval
  - Stop Count
  - Free Buffers
  - Free virtual channels

- Xbar demands: Number of active requesters for a given output port. Multiple con-current requests for an output port indicate a convergent traffic pattern, a likely bottleneck.

- Average Queuing delay
  - History Based
  - Averaging router delays
  - Probability based
  - Based on traffic,application

- Window Based
  - Eg. Rolling buffers

We performed experiment on these parameters . But due to simulation constraints and lack of congestion aware test applications we could not generate differentiable advantage

**Congestion Handling**

- Most work focussed on recovery after detection
- Techniques used are
  - Exploiting adaptability of routing algorithm

    * Minimal: Less Uniform Distribution of Traffic.

    *Non-Minimal: More Uniform Distribution But increases power and latency .

  - Inducing virtual channel
  - buffer optimization
  - Arbitrary links (Small World Network)

## B.2 Congestion Aware Router Architecture

### Router Architecture and its Components

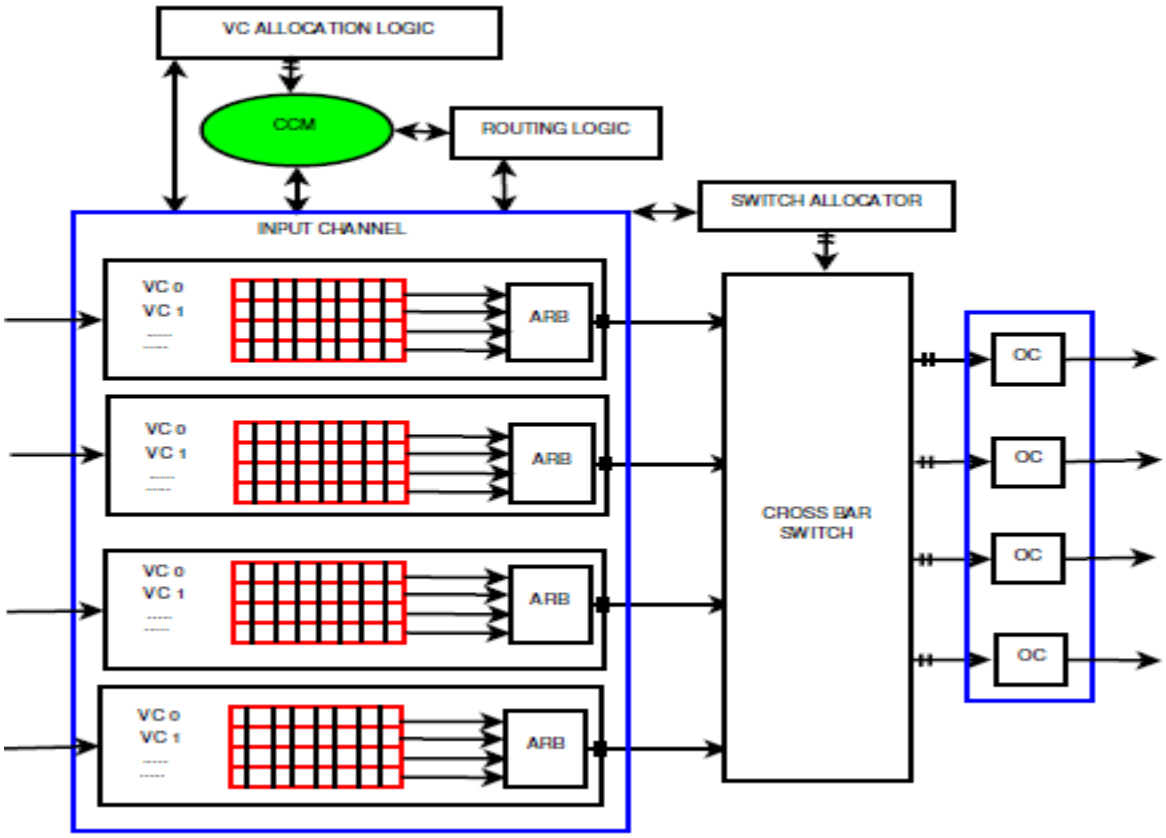


Figure B.1: Congestion Aware Router Architecture

Router is the building block of the NoC. NoC is a connection of routers which are programmable. Depending on the topology of the network, the architecture of the router may be a little different than the other topology, but the building blocks of the router are same.

There are follflowing components in any router :

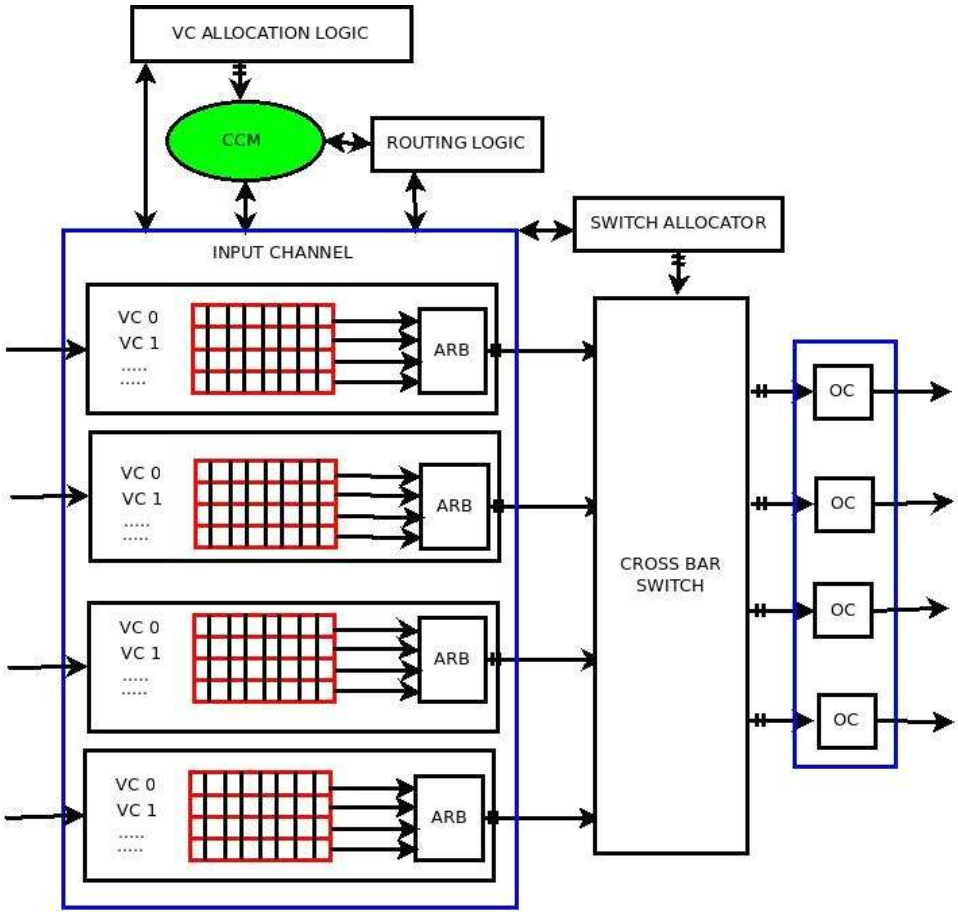


Figure B.2: Router Architecture of NoC

**Input Channel** : Input channel is a buffer FIFO which is used to store the messages that are being forwarded in the network. The communication in any NoC generally happens in the ow controllable units known as its. The input channel is a FIFO that is used to store the its while the router is busy doing processing on other its. Generally Input channel is logically split in multiple channels. This logical partitioning results in existence of virtual channels.

**Routing logic** : Routing logic helps us determine the next direction the packet should take in order to reach the destination. When any input channel receives the packet, the packet requests for the next direction which is ful lled by the routing logic.

**Virtual Channel Allocation** : Once any packet gets to know the next direction and has multiple virtual channels, it can request for a virtual channel for the next direction. This is important to ensure that the virtual channels are reserved properly because there are

two advantages of virtual channels : a. It helps us in wormhole switching. b. It does not allow traffic mixing from multiple sources.

**Switch Allocator** :Switch allocator is responsible for switching the packets from one hop to other hop depending on the next direction and the reserved virtual channel. Switch allocator is also responsible for ow control. If any next hop does not have enough credits, switch is not allotted for that direction.

**Cross bar** : It is a connecting unit which provides input to output mapping. It consists of multiplexers. Depending on the select lines , the multiplexers connect the input to output channels.

**CCM** : Congestion Control Module(CCM) monitors the traffic and available resources and enhance performance.

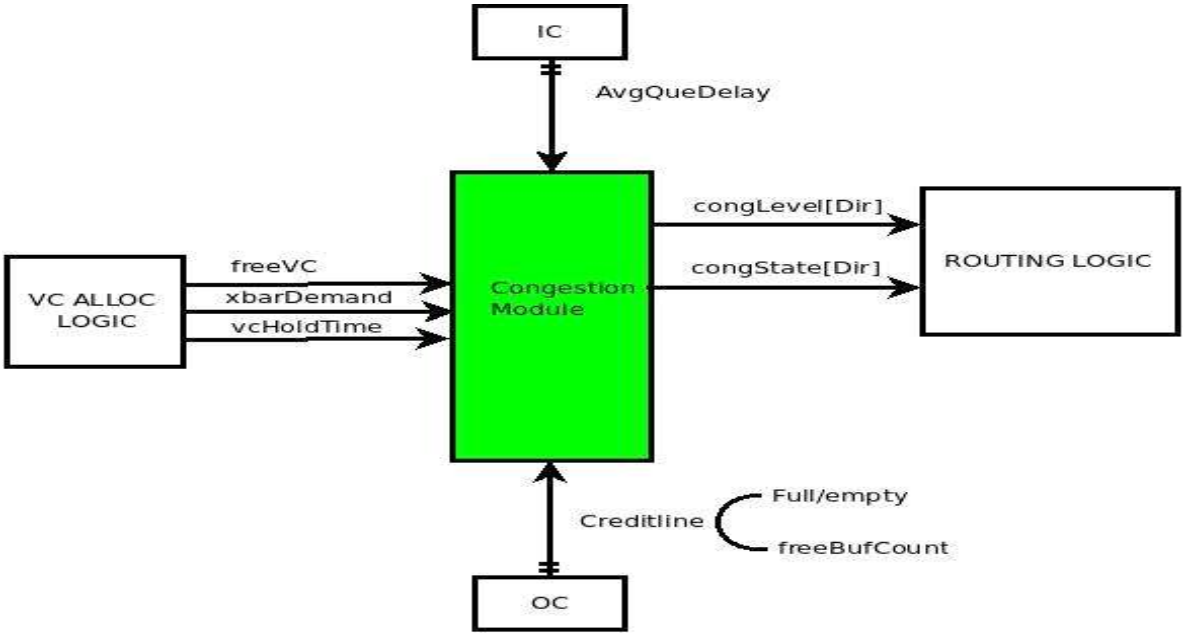


Figure B.3: Congestion Control Module

# Bibliography


- Tobias Bjerregaard and Shankar Mahadevan. A survey of research and practices of network-on-chip. ACM Comput. Surv., 38(1), June 2006.

- W. J. Dally and B. P. Towles. Principles and practices of interconnection networks. Morgan Kaufmann, 2004.

- W.J. Dally and B. Towles. Route packets, not wires: on-chip interconnection networks. In

  Design Automation Conference, 2001. Proceedings, pages 684-689, 2001.

- L Benini and G De Micheli. Networks on chips: A new soc paradigm. 2002.

- J. Duato, S. Yalamanchili, and L.M Ni. Interconnection networks - an engineering approach. IEEE, 1997.

- M. Kumar, Pankaj, V. Laxmi, M.S. Gaur, and Seok-Bum Ko. Reconfigurable distributed fault tolerant Routing Algorithm for on-chip networks. In Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT), 2013 IEEE International Symposium on, pages 290-295, Oct 2013.

- C.J. Glass and L.M. Ni. The turn model for Adaptive Routing. In Proceedings of 19th International Symposium on Computer Architecture, pages 278-287, 1992.

- Zhen Zhang, A. Greiner, and S. Taktak. A Reconfigurable routing algorithm for a fault-tolerant 2d-mesh network-on-chip. In Proceedings of 45th Design Automation Conference, pages 441-446, 2008.

- Ge-Ming Chiu. The odd-even turn model for adaptive routing. Parallel and Distributed Systems, IEEE Transactions on, 11(7):729-738, 2000.

- Binzhang Fu, Yinhe Han, Jun Ma, Huawei Li, and Xiaowei Li. An abacus turn model for time/space-efficinetReconfigurable routing. In Proceedings of 38th International Symposium on Computer Architecture, pages 259-270, 2011.

- Christopher J Glass and Lionel M Ni. Fault-tolerant wormhole routing in meshes. In

  Proceedings of 23rd International Symposium on Fault-Tolerant Computing, pages 240-249, 1993.

- Daniel H. Linder and James C. Harden. An adaptive and fault tolerant wormhole routing strategy for k-ary n-cubes. Computers, IEEE Transactions on, 40(1):2-12, 1991.

- C. R. Jesshope, P. R. Miller, and J. T. Yantchev. High performance communications in processor networks. SIGARCH Comput. Archit. News, 17(3):150-157, April 1989.

- Christopher J Glass and Lionel M Ni. Maximally fully adaptive routing in 2d meshes. In

  International Conference on Parallel Processing, volume I, pages 101-104, 1992.

- L-S Peh and William J Dally. A delay model and speculative architecture for pipelined routers. In High-Performance Computer Architecture, 2001. HPCA. The Seventh International Symposium on, pages 255-266. IEEE, 2001.

- Eung S. Shin, Vincent J. Mooney, III, and George F. Riley. Round-robin arbiter design and generation. In Proceedings of the International Symposium on System Synthesis, pages 243-248, 2002.

- Jingcao Hu and Radu Marculescu. DyAD: smart Routing for Network-on-Chip. In Proceed-ings of 41st Design Automation Conference, pages 260-263. ACM, 2004.

- A. Chien and J.H. Kim. Planar-Adaptive Routing: Low-cost Adaptive networks for multiprocessors. In Proceedings of 19th International Symposium on Computer Architecture, pages 268-277, 1992.

- Ming Li, Qing-An Zeng, and Wen-Ben Jone. DyXY - a proximity congestion-aware deadlock-free dynamic Routing method for network on chip. In Proceedings of 43rd Design Automation Conference, pages 849-852, 2006.

- P. Gratz, B. Grot, and S.W. Keckler. Regional congestion awareness for load balance in Network-on-Chip. In Proceedings of 14th International Symposium on High Performance Computer Architecture, pages 203-214, 2008.

- R.S. Ramanujam and Bill Lin. Destination-based Adaptive Routing on 2D mesh networks. In Proceedings of 6th ACM/IEEE Symposium on Architectures for Networking and Com-munications Systems, pages 1-12, 2010.

- Sheng Ma, N.E. Jerger, and Zhiying Wang. DBAR: An efficinetRouting Algorithm to support multiple concurrent applications in Network-on-Chip. In Proceedings of 38th Inter-national Symposium on Computer Architecture, pages 413-424, 2011.

- M. Ebrahimi, M. Daneshtalab, P. Liljeberg, J. Plosila, and H. Tenhunen. CATRA-conges-tion aware trapezoid-based Routing Algorithm for on-chip networks. In Proceedings of 15th Design, Automation and Test in Europe Conference Exhibition, pages 320-325, 2012.

- M. Ebrahimi, M. Daneshtalab, P. Liljeberg, J. Plosila, and H. Tenhunen. LEAR - a low-weight and highly Adaptive Routing method for distributing congestions in on-chip net-works. In Proceedings of 20th Euromicro International Conference on Parallel, Distributed and Network-Based Processing, pages 520-524, 2012.

- Masoumeh Ebrahimi, Masoud Daneshtalab, Juha Plosila, and Farhad Mehdipour. MD: Minimal path-based fault-tolerant Routing in on-chip networks. In Proceedings of 18th Asia and South Paci c Design Automation Conference, pages 35-40, 2013.

- William James Dally and Brian Patrick Towles. Principles and Practices of Interconnection Networks. Morgan Kaufmann, 2004.

- Jose Duato, Sudhakar Yalamanchili, and Lionel Ni. Interconnection Networks - An Engineering Approach. Morgan Kaufmann, 2003.

- M. Palesi, R. Holsmark, S. Kumar, and V. Catania. Application specific routing algorithms for networks on chip. Parallel and Distributed Systems, IEEE Transactions on, 20(3):316- 330, 2009.

- Jose Flich and Jose Duato. Logic-based distributed routing for nocs. IEEE Computer Architecture Letters, 7(1):13-16, 2008.

- Rajendra V. Boppana and Suresh Chalasani. Fault-tolerant wormhole Routing Algorithms for mesh networks. Computers, IEEE Transactions on, 44(7):848-864, 1995.

- David Fick, Andrew DeOrio, Gregory Chen, Valeria Bertacco, Dennis Sylvester, and David Blaauw. A highly resilient Routing Algorithm for fault-tolerant nocs. In Proceedings of 12th Design, Automation and Test in Europe, pages 21-26, 2009.

- M. Ebrahimi, Xin Chang, M. Daneshtalab, J. Plosila, P. Liljeberg, and H. Tenhunen. DyXYZ: Fully Adaptive Routing Algorithm for 3D NoCs. In Parallel, Distributed and Network-Based Processing (PDP), 2013 21st Euromicro International Conference on, pages 499-503, 2013.

- N. Dahir, T. Mak, R. Al-Dujaily, and A. Yakovlev. Highly Adaptive and deadlock-free Rout-ing for three-dimensional Network-on-Chip. Computers Digital Techniques, IET, 7(6):255- 263, 2013.

- M. Ebrahimi. Fully Adaptive Routing Algorithms and Region-based Approaches for Two-dimensional and Three-dimensional Network-on-Chip. Computers Digital Techniques, IET, 7(6):264-273, 2013.

- M. Coppola, R. Locatelli, G. Maruccia, L. Pieralisi, and A. Scandurra. Spidergon: a novel on-chip communication network. In System-on-Chip, 2004. Proceedings. 2004 International Symposium on, pages 15-, 2004.

- N. Concer, S. Iamundo, and L. Bononi. aequalized: A novel routing algorithm for the spidergon network on chip. In Design, Automation Test in Europe Conference Exhibition, 2009. DATE '09., pages 749-754, 2009.

- Nishant Satya Lakshmikanth, Krishna Kumaar N.I., and Sudha S. Dynamic stress wormhole routing for spidergon noc, 2010. http://www.hipc.org/hipc2010/HIPCSS10/.

- Chaochao Feng, Jinwen Li, Zhonghai Lu, A. Jantsch, and Minxuan Zhang. Evaluation of de ection routing on various noc topologies. In ASIC (ASICON), 2011 IEEE 9th International Conference on, pages 163-166, 2011.

- William J Dally and Charles L Seitz. Deadlock-free message routing in multiprocessor interconnection networks. Computers, IEEE Transactions on, 100(5):547-553, 1987.

- Samuel Rodrigo, Jose Flich, Antoni Roca, Simone Medardoni, Davide Bertozzi, Jesus Ca-macho, Federico Silla, and Jose Duato. Cost-efficineton-chip routing implementations for CMP and MPSoC systems. Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on, 30(4):534-547, 2011.

- V.F. Pavlidis and E.G. Friedman. 3-D Topologies for Networks-on-Chip. Very Large Scale Integration (VLSI) Systems, IEEE Transactions on, 15(10):1081-1090, Oct 2007.

- M. Ebrahimi, M. Daneshtalab, F. Farahnakian, J. Plosila, P. Liljeberg, M. Palesi, and H. Tenhunen. Haraq: Congestion-Aware Learning Model for Highly Adaptive Routing Algorithm in On-Chip Networks. In Networks on Chip (NoCS), 2012 Sixth IEEE/ACM International Symposium on, pages 19-26, May 2012.

- M. Ebrahimi, M. Daneshtalab, J. Plosila, and H. Tenhunen. Mafa: Adaptive fault-tolerant Routing Algorithm for Network-on-Chip. In Digital System Design (DSD), 2012 15th Eu-romicro Conference on, pages 201-207, Sept 2012.

- P. Lot -Kamran, M. Daneshtalab, C. Lucas, and Z. Navabi. Barp-a dynamic Routing protocol for balanced distribution of traffic in nocs. In Design, Automation and Test in Europe, 2008. DATE '08, pages 1408-1413, March 2008.

- P. Lot -Kamran, A.M. Rahmani, M. Daneshtalab, A. Afzali-Kusha, and Z. Navabi.

  fEDXYg a Low cost Congestion-aware Routing Algorithm for Network-on-Chips. Journal of Systems Architecture, 56(7):256 - 264, 2010. Special Issue on HW/SW Co-Design: Systems and Networks on Chip.

- J. Duato. A necessary and su cient condition for deadlock-free adaptive routing in wormhole networks. Parallel and Distributed Systems, IEEE Transactions on, 6(10):1055-1067, 1995.

- Nirgam:. http://wiki.mnit.ac.in/mediawiki/index.php/Nirgam/.

- A.B. Kahng, Bin Li, Li-Shiuan Peh, and K. Samadi. ORION 2.0: A fast and accurate noc power and area model for early-stage design space exploration. In Proceedings of 12th Design, Automation and Test in Europe Conference Exhibition, pages 423-428, 2009.

- T.G. Mattson, R. Van Der Wijngaart, and M. Frumkin. Programming the intel 80-core network-on-a-chip terascale processor. In High Performance Computing, Networking, Stor-age and Analysis, 2008. SC 2008. International Conference for, pages 1-11, 2008.

- J. Henkel, W. Wolf, and S. Chakradhar. On-chip networks: a scalable, communication-centric embedded system design paradigm. In VLSI Design, 2004. Proceedings. 17th Inter-national Conference on, pages 845-851, 2004.

- Marcello Coppola, Miltos D. Grammatikakis, Riccardo Locatelli, Giuseppe Maruccia, and Lorenzo Pieralisi. Design of Cost-EfficinetInterconnect Processing Units: Spidergon STNoC. CRC Press, Inc., Boca Raton, FL, USA, 1st edition, 2008.

- L. Jain, B. Al-Hashimi, Manoj Singh. Gaur, Vijay Laxmi, and A. Narayanan. Nirgam: A systemc based cycle accurate noc simulator, 2010.