CHAPTER 2

EXISTING ROUTING TECHNIQUES FOR CONGESTION AVOIDANCE IN NETWORKS ON CHIP

2.1 INTRODUCTION

On-chip communication is playing an increasingly dominant role in System on Chip (SoC) design, as the technology scales and chip integrity grows. The typical approach to present day on-chip communication is the centralized bus-based approach, by which the components on a particular chip communicate over the shared medium. This design requires a centralized arbitration mechanism to avoid congestion. Both the single medium and the centralized arbitrator are inefficient and incur much overhead to maintain and use effectively. A much better approach is to create an interconnect topology beforehand, and then specify an interface that components should use to connect to this interconnected network. This modular approach to SoC design provides far more efficient communication by leveraging computer networking principles, which have been well established for many years. In the traditional model of shared bus as a primary transmission medium to a more general interconnection network eliminates the need for a centralized traffic control mechanism, decreases contention, and allows higher throughput for on-chip communications.

To meet the performance and design productivity requirements, Network on chip has been proposed as a solution to provide better modularity, scalability, reliability and higher bandwidth compared to bus based
communication infrastructure. As one can imagine, the design choices in terms of architecture routing logic have a strong impact on the performance of such chip-based systems. The performance of Networks on Chip henceforth largely depends on the underlying routing techniques, namely output selection and input selection. However, the routing techniques for NoC have some unique design considerations besides low latency and high throughput. Due to tight constraints on memory and computing resources, the routing techniques for NoC should be reasonably simple. Between components of a chip, the data is transferred as packets through the network. The interconnecting network consists of wires, routers, processors, memories and other Intellectual Property (IP) blocks connected to routers. A routing algorithm plays a significant role on network’s operation and determines how the packets are sent from sender to receiver.

The most important consideration in NoC design is the selection of appropriate architecture based on the application constraints. The main constraints to be considered are throughput, efficient utilization of silicon chip space and energy dissipation. It is also to be taken into account, how the layout of a particular architecture can affect power consumption and performance. The increase in energy dissipation can vary inversely with respect to increase in throughput (Marconett 2010).

The operations of the Networks on Chip are distinguished between three layers, the physical layer, the switching layer, and the routing layer. The physical layer refers to link-level protocols for transferring messages and also managing the physical channels between adjacent routers. The switching layer utilizes these physical layer protocols to implement mechanisms for forwarding messages through the network. Finally, the routing layer makes routing decisions to determine probable output channels at intermediate router nodes and thereby establish the path through the network. The design of
routing protocols and their properties (e.g., deadlock and live lock freedom) are largely determined by the services provided by the switching layer. The switching techniques determine when and how internal switches are set to connect router inputs to outputs and the time at which message components may be transferred along these paths. These techniques are coupled with flow control mechanisms for the synchronized transfer of units of information between routers and through routers in forwarding messages through the network.

Flow control is tightly coupled with buffer management algorithms that determine how message buffers are requested and released, and as a result determine how messages are handled when blocked in the network. Implementations of the switching layer differ in decisions made in each of these areas and in their relative timing, that is, when one operation can be initiated relative to the occurrence of the other. The chosen decision and their relative timing interact with the architecture of the routers and traffic patterns imposed by parallel programs in determining the latency and throughput characteristics of the interconnection network.

Research activities carried out by researchers in the field of Networks on Chip are discussed. The existing works exploring different design aspects of NoC found in the literature are presented below.

2.2 **TOPOLOGIES FOR NETWORKS ON CHIP**

On-chip networks are composed of a set of shared router nodes and communication channels, and the topology of the network refers to the arrangement of these nodes and communication channels. It directly affects all key network parameters like channel length, router complexity, latency and throughput. These parameters translate into performance and power of the network. Best choice of topology for a particular application depends on
design goals and parameters. From the communication perspective, there have been various topologies for NoC architecture. These include Mesh, Torus, Ring, Butterfly, Octagon and irregular interconnection networks. There are a lot of research works reported in literature addressing NoCs topology generation. Bei Yu et al (2010) have proposed a methodology to design the suitable topology that minimizes the power consumption of interconnects and network components in NoC.

### 2.2.1 Mesh Topology

Networks with the two dimensional (2D) Mesh topology offer massive parallelism and scalability than many other approaches dealt using MultiProcessor SoC (MPSoC) interconnection networks (Ni and McKinley 1993). Besides, Meshes are suited to a variety of applications including matrix computation, image processing and problems whose task graphs can be embedded naturally into the topology (Duato 2003). 2D Mesh as shown in Figure 2.1 is the most common on-chip network topology providing high throughput point-to-point communication due to its simplicity and regularity.

![Figure 2.1 Mesh Topology of NoC](image-url)
Formally, an \( m \times n \) 2D Mesh consists of \( N = m \times n \) nodes; each node has an associated integer coordinate pair \((x, y)\), \(0 \leq x < n\) and \(0 \leq y < m\). Two nodes with coordinates \((x_i, y_i)\) and \((x_j, y_j)\) are connected by a communication channel if and only if \(|x_i - x_j| + |y_i - y_j| = 1\).

### 2.2.2 Torus Topology

A Torus network as shown in Figure 2.2 is an improved version of basic Mesh network. A simple Torus network is a Mesh in which the heads of the columns are connected to the tails of the columns and the left sides of the rows are connected to the right sides of the rows. Torus network has better path diversity than Mesh network, and it also has more minimal routes.

![Torus Topology of NoC](image)

**Figure 2.2 Torus Topology of NoC**

### 2.2.3 Ring Topology

In Ring as shown in Figure 2.3, the routing strategy selects the clockwise or counter-clockwise direction depending on the shortest path. Deadlock is resolved by adopting two virtual channels (multiple output queues) for each physical link. In Ring based NoC, the routing strategy selects
the clockwise or counterclockwise direction to transfer the packets from the source to the target node, depending on the shortest path direction. Deadlock is resolved by adopting two virtual channels (multiple output queues) for each physical link. The dimension order routing algorithms adopted in Mesh and Spidergon allows reduction of the buffer requirement, while the Ring architecture requires virtual channels doubling the architecture’s buffer needs. Crossbar has no internal buffers / virtual channels (Benini 2006). The Ring has fewer channels than the other architectures and behaves generally worse than the other NoC topologies.

![Figure 2.3 Ring Topology of NoC](image)

2.2.4 Spidergon Topology

Spidergon topology addresses the demand for a fixed and optimized topology to realize low cost multi-processor SoC implementation. In the Spidergon topology an even number of nodes are connected by unidirectional links to the neighboring nodes in clockwise and counter-clockwise directions.
plus a cross connection for each pair of nodes. In the Spidergon scheme each physical link is shared by two virtual channels in order to avoid deadlock. Figure 2.4 depicts a Spidergon topology of size 16 and its switch. The key characteristics of the Spidergon topology include good network diameter, constant node degree (equal to 3), homogeneous building blocks (the same router to compose the entire network), vertex symmetry and simple routing scheme. Moreover, the Spidergon scheme employs packet-based wormhole routing which can provide low message latency at a low cost. High node degree reduces the average path length but increases complexity.

In the Spidergon topology, nodes are connected by unidirectional links. Let the number of nodes be even \( N = 2n \). Every node in the network is indexed 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. The channels around the topology are given the same label as the nodes connected to them in clockwise direction. And the channels connecting cross network nodes are given label of the node with the lower index plus \( N \). Each

![Figure 2.4 Spidergon (4 X 4) Architecture with the Switch](image-url)
node in the network, \(x_i(0 \leq i \leq N)\), is directly connected to node \(x_{i+1 \mod N}\) by a clockwise link, to a node \(x_{i-\frac{N}{2} \mod N}\) by a counter-clockwise link and to node \(x_{i+\frac{N}{2} \mod N}\) by a cross link. Figure 2.5 clearly shows the high architectural advantage of Spidergon NoC.

![Comparison of Various Topologies of NoC](image)

**Figure 2.5 Comparison of Various Topologies of NoC**

Increasing the dimension of the architecture along X-axis and considering the longest path from some source to destination along the Y-axis, it can be seen that the Spidergon NoC clearly confirms the architectural advantage.

### 2.3 ROUTER ARCHITECTURE IN NETWORKS ON CHIP

The basic wormhole router functions can be captured in canonical router architecture (Duato et al 2003). The wormhole router shown in Figure 2.6 comprises of crossbar switch, link controller (LC), virtual channel controller (VC), routing and arbitration unit, buffers and processor interface. The functions of these components are described in brief in the following sections.
2.3.1 Crossbar Switch

This component is responsible for connecting router input buffers to router output buffers. High-speed routers will utilize crossbar networks with full connectivity, while lower-speed implementations may utilize networks that do not provide full connectivity between input buffers and output buffers. The m X n can connect any of the m inputs to each of the n outputs. The primary issue when using a crossbar as the switch in a router datapath is to provide speed up on the input of the crossbar, the output of the crossbar or on both sides. The speedup can be provided in space or time.

2.3.2 Link Controller

Flow control across the physical channel between adjacent routers is implemented by this unit. The link controllers on either side of a channel
coordinate to transfer flow control units. Sufficient buffering must be provided on the receiving side to account for delays in propagation of data and flow control signals. When a controller on the output channel signals buffer full status through an event, the controller at the input channel must have sufficient buffer to store all of the phits in transit, as well as all of the phits that will be injected during the time it takes for the flow control signal to propagate back to the output channel. If virtual channels are present, the controller is also responsible for decoding the destination channel of the received phit.

2.3.3 Virtual Channel Controller

This component is responsible for multiplexing the contents of the virtual channel onto the physical channel. With tree-based arbitration, the delay can be expected to be logarithmic in the number of channels.

2.3.4 Routing and Arbitration Unit

Routing and arbitration unit implements the routing function. For adaptive routing protocols, the message headers are processed to compute the set of candidate output channels and generate requests for them. If relative addressing is being used in the network, the new headers, one for each candidate output channel must be generated. For oblivious routing protocols, header update is a very simple operation. Alternatively, if absolute addressing is used, header processing is reduced since new headers do not need to be generated.

Routing and arbitration unit also implements the selection function of the routing algorithm: selecting the output link for an incoming message. Output channel status is combined with input channel requests. Conflicts for the same output must be arbitrated and if relative addressing is used, a header
must be selected. If the requested buffers are full, the incoming message remains in the input buffer until a requested buffer becomes free. Figure 2.6 shows a full crossbar that connects all input virtual channels to all output virtual channels. Alternatively, the router may use a design where full connectivity is only provided between physical channels, and virtual channels arbitrate for crossbar input ports. Fast arbitration policies are crucial in maintaining low flow control latency through the switch.

2.3.5 Buffers

These are FIFO (First In First Out) buffers for storing messages in transit. In the router model shown in Figure 2.6, a buffer is associated with both input and output physical channels. The buffer size is an integral number of flow control units. In alternative designs, buffers may be associated only with inputs (input buffering) or outputs (output buffering). In packet switching, however, a packet is transmitted through a channel as soon as that channel is reserved, but the next channel is not reserved (assuming that it is available) until the packet releases the channel it is currently using. Obviously, some buffer space is required to store the packet until the next channel is reserved. That buffer should be allocated before starting packet transmission. So, buffer allocation is closely related to the switching mechanism.

2.3.6 Processor Interface

This component simply implements a physical channel interface to the processor rather than to an adjacent router. It consists of one or more injection channels from the processor and one or more ejection channels to the processor. Ejection channels are also referred to as delivery channels or consumption channels.
2.4 SWITCHING TECHNIQUES IN NETWORKS ON CHIP

Network flow control also called as routing mode, determines how packets are transmitted inside a network. The mode is not directly dependent to routing algorithm. Many algorithms are designed to use some given mode, but most of them do not define which mode should be used. The switching techniques commonly used for on-chip networks are store and forward switching, virtual cut-through switching and wormhole switching whose functions are described in the following sections.

2.4.1 Store and Forward Switching

Store and forward is the simplest routing mode. Packets move in one piece, and entire packet has to be stored in the router’s memory before it can be forwarded to the next router. So the buffer memory has to be as large as the largest packet in the network. The latency is the combined time of receiving a packet and sending it ahead. Sending cannot be started before the whole packet is received and stored in the router’s memory.

2.4.2 Virtual Cut-Through Switching

Virtual cut-through is an improved version of store and forward mode. A router can begin to send packet to the next router as soon as the next router gives permission. Packet is stored in the router until the forwarding begins. Forwarding can be started before the whole packet is received and stored to router. In this switching mode, more buffer memory is needed similar to store and forward mode, but the latency is lower. Hu and Marculescu (2004a) have presented a buffer planning algorithm that plans the buffer depth at each input channel of the router based on the application characteristics. When virtual cut-through switching is used, the packets are considered as single customer.
2.4.3 Wormhole Switching

In wormhole switching, packets are divided into small and equal sized flits (flow control digit or flow control unit). A first flit of a packet is routed similarly as packets in the virtual cut-through routing. After first flit, the route is reserved to route the remaining flits of the packet. This route is called wormhole. Wormhole mode requires less memory than the two other modes because only one flit has to be stored at once. Also the latency is smaller and a risk of deadlock is larger. The risk can be reduced by multiplexing several virtual ports to one physical port, so the possibility of traffic congestion and blocking decreases. Schwiebert and Jayasimha (1995) have studied the necessary and sufficient condition for deadlock free routing for wormhole routers. The local information of the router is considered because non-local information may increase the design complexity. Hu and Kleinrock (1997) have developed an analytical model for wormhole router with finite size buffers. The model uses a deadlock free routing algorithm.

2.5 ROUTING ALGORITHMS IN NETWORKS ON CHIP

The properties of the routing algorithm are:

- Connectivity
  Ability to route packets from any source node to any destination node.

- Adaptivity
  Ability to route packets through alternative paths in the presence of contention or faulty components.

- Deadlock and livelock freedom
  Ability to guarantee that packets will not block or wander across the network forever.
Fault tolerance

Ability to route packets in the presence of faulty components.

Figure 2.7 presents taxonomy of routing algorithms (Duato et al. 2003). Routing algorithms can be classified according to several criteria. Those criteria are indicated in the left column. Each row contains the alternative approaches that can be followed for each criterion. Arrows indicate the relations between different approaches.

Figure 2.7 Taxonomy of Routing Algorithms
Routing algorithms can be primarily classified according to the number of destinations. Packets may have a single destination (unicast routing) or multiple destinations (multicast routing). Routing algorithms can also be classified according to the place where routing decisions are taken. Basically, the path can be either established by a centralized controller (centralized routing) at the source node prior to packet injection (source routing) or determined in a distributed manner while the packet travels across the network (distributed routing). Hybrid schemes called multiphase routing is also possible. In multiphase routing, the source node computes certain destination nodes. The path between them is established in a distributed manner.

Routing algorithms can be implemented in different ways. The most interesting ways proposed up to now consist of either looking at a routing table (table lookup) or executing a routing algorithm in software or hardware according to a finite-state machine. In both cases, the routing algorithm can be either deterministic or adaptive. Deterministic routing algorithms always supply the same path between a given source / destination pair. Adaptive routing algorithms use information about network traffic and/or channel status to avoid congestion or faulty regions of the network.

At a lower level, routing algorithms can be classified according to their minimality as profitable or misrouting. Profitable routing algorithms only supply channels that bring the packet closer to its destination. They are also referred to as minimal. Misrouting algorithms may also supply channels that send the packet away from its destination. They are also referred to as non-minimal. At the lowest level, routing algorithms can be classified according to the number of alternative paths as completely adaptive (also known as fully adaptive) or partially adaptive.
If a link becomes congested or fails, the unique path property can easily disrupt the communication between some input and output pairs. The congestion of packets over certain channels causes the known hotspot problem (Pfister and Norton 1985). Most popular solution is to provide multiple routing paths between any source and destination pair so as to reduce network congestion as well as to achieve fault tolerance. These methods usually require additional hardware, such as extra stages or additional channels. In Dally and Aoki (1993), three different selection functions were proposed and evaluated for 2D Meshes using wormhole switching which shows that minimum congestion can be achieved with the lowest latency and highest throughput.

The failure of routers and links produces congestion in their vicinity as packets are rerouted. These packets will experience increased blocking delays as they compete for access to smaller number of links. The presence of link congestion and blocking within the network can produce markedly different behavior, particularly if the packet destinations and lengths are non uniform. Each additional fault increases the percentage of the packet population that have to be rerouted, contributing to congestion and further degradation in performance.

The performance also depends on the buffering available at the individual nodes. The presence of virtual channels will reduce the blocking delay and increase the overall network throughput. Virtual channels can have a significant impact on latency. Hence there is a need for congestion control mechanism in the network by placing a limit on the size of the buffer on the injection channels (Boppana and Chalasani 1993), or by restricting injected messages to use some predetermined virtual channel(s) (Dally and Aoki 1993), or by waiting until the number of free output virtual channels at a node is higher than a threshold (L´opez and Duato 1993).
In unicast routing, each source sends its message to precisely one destination. Routing algorithms for unicast communication are usually implemented as system calls in parallel machines. More powerful communication primitives that are useful in the execution of parallel programs are broadcast and multicast, which allow data to be transferred from one source node to many destinations. Various algorithms mostly used in on-chip networks are given below. They are multicast routing, oblivious routing, adaptive routing and deterministic routing.

2.5.1 Multicast Routing

Multicast, or one-to-many, communication is a fundamental collective communication operation and is highly demanded in parallel computing and telecommunication applications. Examples of such applications include Fast Fourier Transform (FFT), barrier synchronization, and write update / invalidate in directory-based cache coherence protocols (Lin and Ni 1993). Also, teleconferencing and video broadcasting are typical applications in a telecommunication environment. There has been growing interest in supporting multicast in parallel computers. Multicast can be supported in either hardware or software. There has also been much work on supporting multicast in wormhole routed direct networks in software. In addition, there has been some work on supporting multicast in software in multistage networks (Duato et al 2003).

In general, the multicasting problem can be modeled by three routing schemes: unicast based, tree-based (Malumbres et al 1996) and path-based routing (Lin and Ni 1993). In unicast based, the multicast operation is performed by sending a separate copy of message from the source to every destination or, alternatively, by sending the unicast message to subset of destinations. The drawback of this scheme is on account of the fact that multiple copies of the same message are injected into the network, the traffic
of the network will be increased. Furthermore, each copy of message loses considerable startup latency at the source. In tree-based multicast approach, a spanning tree is constructed in which the source is indicated as the root and then messages are sent down the tree. In this way a message might be replicated at some of the intermediate nodes and forwarded along multiple outgoing channels toward disjoint subsets of destinations. If one branch of the tree is blocked, all are blocked. Branches must proceed forward in block step, which may cause a message to hold many channels for extended periods, resulting in increasing network contention.

Although such schemes have to be used effectively in networks employing store and forward, virtual cut-through routing and tree-based routing incur high congestion in wormhole networks (Lin and Ni 1994). A solution to overcome the disadvantage of tree-based routing is to utilize the path based multicast wormhole routing. In this method, a source node prepares a message for delivery to a set of destinations by first sorting the addresses of the destination in order in which they are to be delivered, and then placing this sorted list in the header of the message. A number of studies have shown that a path-based exhibit superior performance characteristic over their unicast-based and tree-based counterparts (Boppana et al 1998). Yuanyuan Yang (1998) has presented a class of interconnection networks for supporting multicast communications in parallel computing systems.

Wang et al (2002) have presented a Dual-Hamiltonian-Path-Based routing (DHPB) model with two virtual channels based on two Hamiltonian paths and a network partitioning strategy for wormhole routed star graph networks. They have proposed three efficient multicast routing schemes on basis of their model which are deadlock-free. The first scheme, network selection-based dual-path routing, selects sub networks that are constructed either by the first Hamiltonian path or by the second Hamiltonian path for
dual-path routing. The second scheme selects sub networks with optimum routing path for dual-path routing. The third scheme, two-phase optimum dual-path routing, includes two phases, source-to-relay and relay-to-destination. The experimental results show that their model performs better than unicast-based, the Hamiltonian-path, and the Single-Hamiltonian-Path-Based (SHPB) routing schemes significantly.

Since software approach of multicasting is not sufficient, Lu et al (2006) have proposed a hardware approach for multicast scheme for wormhole switched Networks on Chip. They have formed the multicast group by reserving virtual channels. They have concluded from the experimental results that the multicast does not affect the performance of the unicast traffic when the network is not saturated. They have also shown improved latency of the multicast with decreased throughput.

2.5.2 Oblivious Routing

Oblivious routing algorithms have no information about conditions of the network, like traffic amounts or congestions. A router makes routing decisions on the grounds of some algorithm for example randomly. The simplest oblivious routing algorithm is a minimal turn routing. It routes packets using as few turns as possible.

2.5.3 Adaptive Routing

Duato (1993) has proposed the theoretical background for the development of deadlock-free adaptive routing algorithms for wormhole networks. The routing algorithms which are also fault-tolerant were simulated; it showed higher performance. The effect of the virtual channels had been studied in their work. Chiu (2000) has presented a model that restricts packets taking certain turns without usage of virtual channels,
thereby avoiding deadlock. Their experimental results show uniform degree of adaptiveness for every pair of source-destination nodes. Though the model can be extended to non-uniform traffic, they do not support fault tolerance.

Congestion occurs in computer networks when a resource demands exceed the capacity i.e. as packets fill the queues and queue size grows until packets are lost. Even though the bandwidth of the network links is increased, the congestion does not get reduced because increase in link diversity causes severe congestion. In Networks on Chip, IP blocks that may generate arbitrary amount of traffic are connected. Depending on the path diversity and number of processing cores, congestion becomes inevitable. Common solutions to avoid congestion are congestion based routing, where an alternate path with less congestion is chosen. Increasing the buffer size will be an alternate solution that compensate for short bursts but on-chip networks are resource constrained.

Kim et al (2005) had devised a new router architecture which utilizes adaptive routing. Their low latency two-stage pipelined architecture uses neighboring congestion status to select the optimal output path with low latency. The routing algorithm shows reduced energy consumption. Ascia et al (2006) introduced the concept of Neighbors-on-Path to exploit the situations of indecision that can occur in an adaptive wormhole routing. The buffer status of the neighboring nodes is known by dedicated signals based on which the path of less congestion is computed. Turn model routing is based on prohibiting certain turns during the packet routing to prevent deadlocks.

Umit and Marculescu (2007) developed a thorough performance analysis for wormhole routing with arbitrary size messages and finite buffers under application-specific traffic patterns. This model supports arbitrary topologies and deterministic routing. Through this model, various design parameters that affect the network performance can be studied.
Daneshtalab et al (2007) proposed a router that selects minimal or non-minimal adaptive path through the circuit that detects the congestion in the neighboring nodes. Arjun Singh et al (2003) has proposed a non-minimal, adaptive routing algorithm for Torus networks that strikes a balance between the conflicting goals of locality and load balance. The algorithm uses the queue length as the congestion indicator. Yaoting et al (2007) has presented a routing scheme that uses multiple shortest paths to transfer message concurrently with the help of simple control schemes.

Tsai et al (2010) have proposed Turn-Model based Fully-Adaptive-Routing (TM-FAR) algorithm for Networks on Chip. TM-FAR retains the deadlock-free property of traditional turn model based routing algorithms (e.g., XY, Odd-Even), while relieving restrictions on turn and path selections.

### 2.5.4 Deterministic Routing

Deterministic routing algorithms establish the path as a function of the destination address, always supplying the same path between every pair of nodes. A deterministic routing algorithm will always provide the same output channel for the same destination. While deterministic algorithms are oblivious, the converse is not necessarily true. Deterministic routing became very popular when the wormhole switching was widely used. One main advantage of deterministic routing is its simplicity in terms of router design. Because of the simplified logic, the deterministic routing provides low latency when the network is not congested. However, the deterministic routers fail to respond dynamically to network congestion when the packet injected into the network increases.

In the case Hypercubes, Meshes, and Torus topologies, it is easy to compute the distance between current and destination nodes as the sum of the
offsets in all the dimensions. The simplest routing algorithm consists of reducing an offset to zero before considering the offset in the next dimension called as Dimension Order Routing (Ni and McKinley 1993). Dimension Order Routing (DOR) guarantees a minimal hop count and has a simple implementation making it a popular choice for NoC systems. While DOR performs well at low network load due to the short header latency, its performance degrades quickly as the load on the network increases due to lack of load balancing. It is called as XY routing for the 2D Mesh topology and e-cube routing for Hypercubes. A distance vector routing and a link state routing are shortest path routing algorithms. Ali et al (1997) has analyzed the need for fault tolerant algorithms. They have proposed simple form of link state routing algorithm that suits a NoC in the event of failures.

2.6  FAULT TOLERANCE IN NETWORKS ON CHIP

Performance and fault tolerance are two dominant issues facing the design of interconnection networks for large-scale multiprocessor architectures. Fault tolerance is the ability of the network to function in the presence of component failures. However, techniques used to realize fault tolerance are often at the expense of considerable performance degradation. Duato (1997) proposed a theory of fault tolerant adaptive routing which requires at least four virtual channels per physical channel.

Another fault tolerant routing algorithm that uses virtual channels has been proposed by Park et al (2006). Switches using a routing algorithm with virtual channels need additional logic circuits for buffers and multiplexers. Their approach has the undesirable characteristic of requiring additional logic circuits for buffers and multiplexers that are more liable to failures.
Zhou and Lau (2001) proposed a fault tolerant routing algorithm which uses virtual channels for the routing. Faulty links and switches are combined to f-polygons. The packets are routed around these f-polygons. The major drawback of this solution is caused by the use of virtual channels. In the case of a single faulty physical link, all virtual links belonging to that faulty physical link become faulty as well. Boppana and Chalasani (1995) have presented a similar approach that uses virtual channels for fault tolerance and combines the faults in f-chains and f-rings around which the packets are routed. Another approach for fault tolerance that uses f-chains and f-rings for routing packets around faulty regions is presented in the work by Chen and Chiu (1998).

Li et al (2006) proposed a Dynamic XY (DyXY) routing, which provides adaptive routing based on congestion conditions in the proximity, and ensures deadlock-free and livelock-free routing at the same time. The algorithm uses dedicated wires to investigate the status of the neighboring switches. The drawback is that its foresight is limited and it can show only the status of the neighbouring nodes as busy or not. Another routing algorithm taking the status of a switch into account is presented by Hu et al (2004b). The switch decides to route a packet into a specific direction based on its own status and the status of its outgoing queues.

An adaptive routing algorithm using stress values to distribute traffic uniformly across the entire network is presented by Lee et al (2005). The drawback of stress values is that they are sent to the neighboring switches when a switch detects, that it is nearly overloaded. That can cause packet loss because the neighboring switches might have sent packets to that switch right before receiving the stress value. Schonwald et al (2007) has proposed routing algorithm called Force Directed Wormhole Routing that distributes the traffic
uniformly across the entire network in the presence of faulty links or switches in the NoCs.

Error control is becoming a growing concern as technology scales toward deep submicron, because of the increased impact on signal reliability of noise sources such as crosstalk, power-supply noise, EMI and soft errors. Corrupted flits can be detected either in hardware by means of error correction or error detection / retransmission mechanisms, or can be handled at higher network layers (e.g., connection oriented transport layer). However, fast error recovery requires a hardware implementation of error control, thus increasing switch and / or Network Interface complexity. The re-use of flow control mechanisms for error handling allows to save some area and power and to avoid duplication of control lines. In Pullini et al (2005), three different buffered flow control techniques that support error control were discussed. The three different flow control schemes supported fault tolerance at different performance points like area or performance or power.

Rantala et al (2006) has proposed a re-routing mechanism and addition of return channel to each node. The routing switch node was simulated and tested in a faulty environment. It was found that the routing policy takes 10 percent longer path than the minimal paths in presence of faulty nodes. Bogdan et al (2007) has designed the on-chip interconnect as a reusable IP. By dividing the communication from computation, they provided built in tolerance with no performance penalty. They have shown that data upsets, packet losses due to buffers overflow, and severe levels of synchronization failures can be tolerated, while providing high levels of performance.

Sudeep et al (2010) have proposed the Odd-Even (OE) and Inverted Odd-Even (IOE) turn models to achieve deadlock free packet traversal. Their results show that the proposed OE+IOE scheme can provide better fault
tolerance (i.e., higher successful packet arrival rates) than traditional fault tolerant routing schemes such as N-random walk and turn model based schemes.

In spite of all the above mentioned works that have been carried out, it has been felt that there is a need for adaptive, congestion-aware and fault tolerant routing model to enhance the performance of Networks on Chip with respect to performance in terms of throughput.

2.7 CONCLUSION

In this chapter, the router architecture of the Network on chip and its components are discussed. Research works related to different types of routing algorithms are also presented. Existing works related to adaptive and fault tolerant wormhole routing are discussed. The need for congestion control mechanism which forms the important objective of this thesis is also emphasized.