Networks-on-Chips: Modeling, Analysis, and Design Methodologies

by

Haytham El Miligi

B.Sc. of Electrical Engineering, Al-Azhar University, 2000

M.Sc. of Electrical Engineering, Al-Azhar University, 2005

A Dissertation Submitted in Partial Fullfillment of the

Requirements for the Degree of

Doctor of Philosophy

in the Electrical and Computer Engineering Department

Networks-on-Chips: Modeling, Analysis, and Design Methodologies

by

Haytham El Miligi

B.Sc. of Electrical Engineering, Al-Azhar University, 2000

M.Sc. of Electrical Engineering, Al-Azhar University, 2005

Supervisory Committee

Dr. Fayez Gebali, Co-Supervisor
(Electrical and Computer Engineering Department)

Dr. M. Watheq El-Kharashi, Co-Supervisor
(Electrical and Computer Engineering Department)

Dr. Kin Fun LI, Department Member
(Electrical and Computer Engineering Department)

Dr. Sadik Dost, Outside Member
(Department of Mechanical Engineering)

Supervisory Committee

Dr. Fayez Gebali, Co-Supervisor

(Electrical and Computer Engineering Department)

Dr. M. Watheq El-Kharashi, Co-Supervisor

(Electrical and Computer Engineering Department)

Dr. Kin Fun LI, Department Member

(Electrical and Computer Engineering Department)

Dr. Sadik Dost, Outside Member

(Department of Mechanical Engineering)

Abstract

The growing complexity of System-on-Chip (SoC) designs motivates both academic and industrial researchers to find better solutions for the complexity of the chip-interconnect. For SoC designs that have hundreds of Processing Elements (PEs), a single shared bus can no longer be accepted as an efficient communication scheme. To address this problem, the Networks-on-Chip (NoC) concept is proposed as a new paradigm, which provides an integrated solution for achieving efficient interconnection scheme for complex SoC applications. NoC-based designs are composed of computational resources in the form of PE cores, and switching nodes (routers) that allow PEs to communicate with each other.

For different applications, this research work: 1) proposes new analytical models for various NoC design parameters, 2) performs comparative analyses of the commonly used network architectures, and 3) presents novel methodologies for efficiently designing the NoC-topology. The proposed methodologies are developed to help NoC-designers better achieve minimum power consumption and delay, and maximum performability for their applications.

Graph-theoretic concepts are adopted to study the topological architecture of NoCs and propose a new topology-based models for network power, performability, and delay. The proposed models take into consideration important design parameters, which significantly affect the power, performability, and delay of a NoC-based system; such as network topology architecture, traffic distribution, noise power, voltage swing, probability of edge failure, router design and number of ports, clock frequency, and target technology.

In this dissertation, we show how the proposed models could be used to optimally design the network topology so that it achieves the target design requirement for a given application. After studying each design metric individually, a joint consider-

ation of NoC power, performability, and delay is carried out simultaneously. We use Particle Swarm Optimization (PSO) to find the optimum network topology, that achieves minimum delay, maximum performability, and minimum power consumption, for a given NoC application.

Real case studies are presented to validate the proposed theoretical concepts. This validation is carried out through experimental work, targeting various real NoC applications. Experimental results show that using the proposed design methodologies, designers can improve the overall system efficiency in terms of power, delay, and performability, by choosing the design parameters (i.e., network topology architecture, PEs' mapping, etc.) efficiently at early design phases. This improvement is measured in some cases by an order of magnitude, compared to the worst case scenario of choosing wrong design parameters for the target application.

Table of Contents

Supervisory Committee **ii**

Abstract **iv**

Table of Contents **vi**

List of Tables **x**

List of Figures **xi**

List of Abbreviations **xvi**

1 Introduction **1**

1.1 Introduction 1

1.1.1 NoC Structure 2

1.1.2 Network Adapter (NA) 3

1.1.3 Networks-on-Chip Router 4

1.2 Problem Statement and Research Approaches 5

1.3 Contributions 6

1.4 Dissertation Organization 7

2 Design for Power: A Topology-Based Approach **9**

2.1 Introduction 9

2.2 Related Work 11

2.3 Power Analysis in NoC-based Systems 13

2.3.1 Networks-on-Chips: A Graph-Theoretic Approach 13

2.3.2 Power Modeling of Global Interconnection Links 15

2.3.3 Power Modeling of NoC Routers 20
2.4 Proposed Methodology 24
2.4.1 Steps 1-2: Convert TDG into a λ Matrix 24
2.4.2 Step 3: Apply Network Partitioning 24
2.4.3 Steps 4-6: Iterative Topology Selection 28
2.4.4 Steps 7-8: Long-Range Link Insertion 29
2.5 Performance Evaluation by Experimentation 30
2.5.1 Steps 1-2: Convert TDG into a λ Matrix 30
2.5.2 Step 3: Apply Network Partitioning 31
2.5.3 Steps 4-8: Topology Selection and Long-Range Link Insertion 33
2.5.4 Results 35
2.6 Chapter Summary 39

3 Design For Performability: A Topology-based Approach 45
3.1 Introduction 45
3.1.1 Sources of Errors in NoC-based Systems 46
3.1.2 Design for Performability 47
3.1.3 Chapter Organization 48
3.2 Related Work 48
3.3 A Graph-Theoretic Representation of NoC Topologies 50
3.4 Modeling NoC Performability 52
3.4.1 Network Functionality 53
3.4.2 Packets Reception 56
3.4.3 Total Network Performability 61
3.5 Topology-based Analysis of System Performability 61
3.6 Problem Formulation and Proposed Methodology 69
3.6.1 Problem Formulation 69

3.6.2 Proposed Methodology 70
3.7 Performance Evaluation by Experimentation 72
3.8 Chapter Summary 77

4 Design For Delay: A Topology-based Approach 79
4.1 Introduction 79
4.2 Related Work 80
4.3 Proposed Model for Network Delay 82
4.3.1 Link's Delay (T_l) 82
4.3.2 Router's Delay (T_r) 84
4.4 Experimental Results 97
4.4.1 Analysis of Output-Queue Model 97
4.4.2 A Case Study 101
4.5 Chapter Summary 104

5 NoC Design for Power, Performability, and Delay using Particle Swarm Optimization 105
5.1 Particle Swarm Optimization 106
5.2 Problem Formulation 108
5.3 A Case Study 111
5.4 Chapter Summary 116

6 Contributions and Future Work 118
6.1 Summary 118
6.2 Contributions 119
6.2.1 Contribution 1 120
6.2.2 Contribution 2 120
6.2.3 Contribution 3 120

6.2.4 Contribution 4 121
6.3 Future Work 121
6.3.1 Research Challenges in Wireless NoCs 123

Bibliography **126**

A List of Publications **146**
A.1 Books 146
A.2 Book Chapters 146
A.3 Journals 146
A.4 Conferences and Workshops 147
A.5 Application Notes 150

List of Tables

2.1 BSIM3 interconnect parameters. 19
2.2 Power consumption of 4-,5-,6-,7-, and 8-port NoC routers when implemented in 0.18μm technology for various operating frequencies. 21
2.3 Partitioning and refinement of the MPEG4 core shown in Figure 2.9(a) based on D entries using Algorithm 2. 33
2.4 Results of graph partitioning using four different methods. 33
2.5 Results of graph partitioning using four different methods before and after applying long-range link insertion. 34
2.6 A comparison of the architecture requirements for different network topologies when 12 PEs are to be connected and their corresponding routers' power consumption. 37
2.7 Cell leakage power consumption of 4-, 5-, 6-, 7-, 8-port (8-word queue size), and 4-port (4-word queue size) routers when implemented in 0.18μm 37

3.1 Minimum edge cut-set and other statistical analysis for NoC topologies shown in Figure 3.1, when 12 PEs are connected. 51
3.2 $\mathcal{P}_{\mathcal{H}n}$ for NoC topologies shown in Figure 3.1, when 12 PEs are connected. 54

5.1 Node descriptions of Figure 5.3. 113
5.2 PSO results. 114
5.3 Improvement ratios: A comparison between Torus and BT for different design parameters. 116

List of Figures

1.1 A sample NoC 3x3 mesh topology. 3
1.2 Network adapter (NA). 4
1.3 A block diagram of an output-queue router. 5

2.1 Eleven standard NoC topologies: (a) Mesh. (b) Torus. (c) Folded Torus. (d) Ring. (e) Octagon (Oct). (f) Spidergon (Spider). (g) Binary tree (BT). (h) Butterfly Fat Tree (BFT). (i) SPIN. (j) Hypercube (Hcube). (k) Star. (Routers are represented by white circles, whereas PEs are represented by dark squares.) 13
2.2 (a) An example of an average traffic distribution graph (TDG). (b) Corresponding traffic distribution matrix (λ). 14
2.3 (a) Ring topology: Routers are represented by white circles and PEs are represented by dark squares. The dashed rectangular represents a packet. (b) Static and dynamic sources of power consumption. 16
2.4 An $m \times m$ output-queue router; (Rx represents an input port, Tx represents an output port, m is the number of ports, and B is the maximum queue size.) 20
2.5 A flowchart of power measurement of NoC routers using Synopsys® tools. 22
2.6 Power consumption versus operating frequency for one packet per port transition of 4-word and 8-word queue size routers when implemented in 0.18 μm technology. 23
2.7 Proposed methodology to minimize the power consumption. 40
2.8 A flowchart showing the proposed graph partitioning steps. 41

2.9 MPEG4 core TDG: The dashed line shows the partitioning boundary after applying graph partitioning. The numbers written on the arrows are the average number of packets/time step transmitted and the numbers written on the circles represent PEs' numbers. Partitioning using (a) our proposed algorithm, (b) Spectral partitioning, (c) Kernighan-Lin partitioning, and (d) Linear partitioning. 42

2.10 MPEG4 application mapping to a combination of star and ring topologies. Routers are represented by white circles, PEs are represented by dark squares and the dashed line is the long-range link. The numbers written inside the circles represent PEs' numbers. 42

2.11 Comparisons between the power consumption of the global links in the generated topology (ring+star) and other standard topologies for MPEG4. (a) Power consumption versus topology type for different frequencies (for 0.18μm technology). (b) Power consumption versus topology type for different technologies (for 100 MHz operating frequency). 43

2.12 Comparisons of the power consumption between the generated topology, ring+star (R+S), and other standard topologies for MPEG4. . . 44

2.13 Comparisons between the generated topology, ring+star (R+S), and other standard and previously proposed topologies for MPEG4. (a) Average number of hops versus topology type. (b) Number of global links versus topology type. 44

3.1 Nine regular NoC topologies connecting 12 PEs: (a) Mesh. (b) Torus. (c) Folded Torus (Folded). (d) Ring. (e) Octagon (Oct). (f) Spidergon (Spider). (g) Binary tree (BT). (h) Butterfly Fat Tree (BFT). (i) SPIN. (Routers are represented by white circles, whereas Processing Elements (PEs) are represented by dark squares.) 50

3.2 An example of a 3 × 4 mesh topology. (Each router-node pairs are represented by a circle.) λ_{AB} packets are being transmitted from A to B. If an edge-failure occurred to 2 links at one of the corner-nodes, one node will be disconnected from the network; $n_f = 1$. 53

3.3 Experimental results that show the impact of the network topology on system probability of success with the change in probability of edge failure. 55

3.4 An example of a 3 × 3 mesh topology. (Routers are represented by white circles and PEs are represented by dark squares.) The gray rectangular represents a number of packets transmitted from node one to node seven (λ_{17}). P is the performability of a single on-chip interconnect and P_{17}^2 represents the performability when λ_{17} packets are transmitted from node 1 to 7. 59

3.5 The impact of the network topology on system performability with respect to traffic rate, noise standard deviation (σ) and the probability of edge failure. 63

3.6 The impact of the network topology on system performability with respect to noise standard deviation (σ), voltage swing (V_{sw}) and the probability of edge failure. 64

3.7 The average inter-node distance (Δ) of 8-, 12-, and 16-core applications for different network topologies. 68

3.8 Proposed methodology to improve NoC performability using a topology-based approach. 71
3.9 (a) TDG of Video Object Plane Decoder (VOPD). (b) VOPD Mapping to a Torus topology (an output from the proposed methodology). . . 73
3.10 Experimental results. 75

4.1 Output-queue router architecture: Packet Ready (PR), Packet Sent (PS), and Receive Ready (RR) signals are used for synchronization purpose. 85
4.2 T_1 depends on the setup time of the input buffer t_{isetup} and the time period between the packet arrival and the next clock falling edge t_w: (a) T_1 probabilities. (b) Timing diagram. 86
4.3 The arrival probability and departure rate for an M/D/1/B queue. . . 87
4.4 2-D M/D/1/B queue state transition diagram. 88
4.5 Throughput versus packet arrival rate for fixed queue size ($B = 8$). . 97
4.6 Throughput versus packet arrival rate for fixed number of states ($n = 8$). 98
4.7 Loss probability versus packet arrival rate for fixed queue size ($B = 8$). 98
4.8 Loss probability versus packet arrival rate for fixed number of states ($n = 8$). 99
4.9 Delay versus packet arrival probability for n=8. 99
4.10 Delay versus packet arrival probability for B=8. 100
4.11 MPEG4 core TDG. 102
4.12 Average network delay for different network topologies. 103
4.13 Mapping of Video Object Plane Decoder application on to Octagon topology. 104

5.1 Nine regular NoC topologies: (a) Mesh. (b) Torus. (c) Folded Torus (Folded). (d) Ring. (e) Octagon (Oct). (f) Spidergon (Spider). (g) Binary tree (BT). (h) Butterfly Fat Tree (BFT). (i) SPIN. Routers are represented by white circles, whereas Processing Elements (PEs) are represented by dark squares. 109
5.2 An example of PSO particle. 110
5.3 Traffic distribution graph of H.263 encoder MP3 decoder. 112
5.4 Comparison between network topologies optimum design values for MP3 encoder application. 115
5.5 Mapping of H.263 MP3 decoder on Torus topology. 116

6.1 An example of MBSN that employs WNoCs. Inside multi-core nodes (dotted circles): distributed routers are represented by white circles whereas central node routers are represented by dark circles and processing elements are represented by dark squares. 124

List of Abbreviations

AI	Artificial Intelligence
ALife	Artificial Life
AMBA	Advanced Microcontroller Bus Architecture
ARQ	Automatic Repeat Request
ASCII	American Standard Code for Information Interchange
AXI	Advanced eXtensible Interface
BCA	Bus Cycle Accurate
BER	Bit Error Rate
BFT	Butterfly Fat Tree
BSN	Body Sensor Network
BT	Binary tree
CBSN	Complex Body Sensor Networks
CDMA	Code-Division Multiple Access
CNN	Cellular Nonlinear Network
CMOS	Complementary Metal Oxide Semiconductor
DFP	David-Fletcher-Powell
DSP	Digital Signal Processing
FEC	Forward Error Control
FIFO	First In First Out
FSM	Finite State Machine
GA	Genetic Algorithms
HARQ	Hybrid ARQ/FEC
Hcube	Hypercube
IP	Intellectual Properity
MBSN	Multi-core body sensor network

MILP	Mixed Integer Linear Programming
NA	Network Adapter
NoC	Networks on Chip
OCP	Open Core Protocol
PE	Processing Element
PSO	Particle Swarm Optimization
PTM	Predictive Technology Model
QoS	Quality of Service
RoC	Radio-on-Chip
RTL	Register Transfer Logic
SAIF	Switching Activity Interchange Format
SER	Soft Error Rates
SF	Switch Fabric
SNFT	Simple Non-Fault-Tolerant
SoC	System-on-Chip
TDG	Traffic Distribution Graph
VCI	Virtual Component Interface
VHDL	VHSIC Hardware Description Language
VHSIC	Very High Speed Integrated Circuit
VLSI	Very Large Scale Integrated Circuit
VOPD	Video Object Plane Decoder
WNoC	Wireless Networks-on-Chips
WSN	Wireless Sensor Networks

Chapter 1

Introduction

1.1 Introduction

The continuous scaling of CMOS technology makes it possible to integrate many heterogeneous Processing Elements (PEs) on a single chip, which is known as System-on-Chip (SoC) design [1].

SoC is a major revolution taking place in the design of integrated circuits due to the ever increasing density of on-chip resources and the scaling of microchip technologies [2]. For complex designs, 3-D SoCs have been introduced to support multiple voltage and frequency islands [3].

Due to the unprecedented levels of integration possible in a single SoC, communication management becomes critical when highly diversified functions must be supported. Traditionally, shared bus architectures that are common to all processors have been the only solutions. However, when the number of components increases, the achievable bit-rate per component decreases which makes buses unsuitable for SoCs with more than 10 bus master components [4]. The growing complexity of SoC designs motivated both academic and industrial researchers to find better solutions for the on-chip communication problem [2,5]. To address this problem, the Networks-

on-Chip (NoC) is proposed as a new paradigm, which provides an integrated solution for achieving efficient interconnection between PEs [4].

1.1.1 NoC Structure

The NoC-based system is composed of four basic components: computational resources in the form of PEs, network adapters (NAs) that implement the interface by which PEs connect to the network, routing nodes (routers) that route data based on a chosen protocol, and links that connect routing nodes and provide a raw communication bandwidth [6]. An IP can be a microprocessor, a DSP, a memory unit, a microcontroller, or any other Intellectual Property (IP) [1] module that can be implemented on a chip. Network adapters are used to interface the IP core to the network and to make communication services transparently available with a minimum effort from the IP core. Routers are responsible for exchanging the data between PEs. Each router has a set of ports, which are used to connect the router to the network. From the router perspective, routing is the mechanism that chooses an output port for a message arriving at an input port [7]. The links that connect routing nodes may consist of one or more logical or physical channels.

Figure 1.1 shows a sample NoC constructed using a 3x3 mesh topology. Instead of using dedicated buses from point to point, a more general scheme is adapted by employing a grid of routing nodes spread out across the chip. These nodes are connected by communication links. In this network topology, each PE is connected to a router through an NA in a 1:1 ratio, whereas routers are connected in a mesh form.

[1] IP and PE module are used exchangeably in this dissertation.

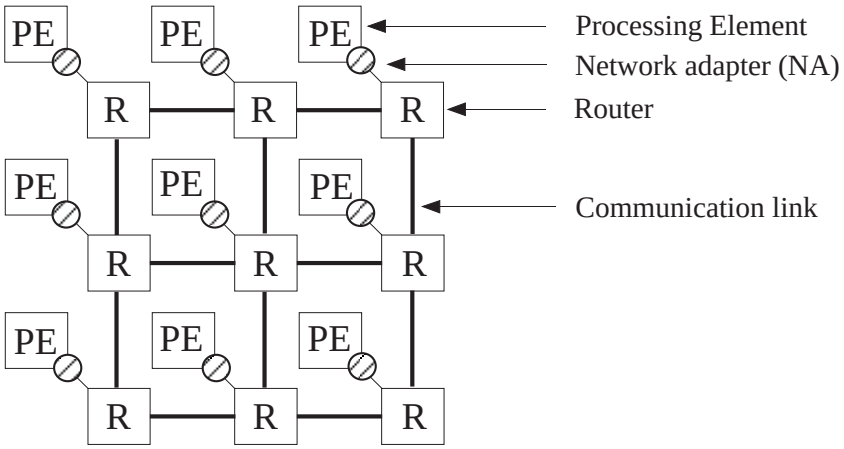


Figure 1.1: A sample NoC 3x3 mesh topology.

1.1.2 Network Adapter (NA)

The network adapter is a key component in NoC-based systems. Network adapters provide a standard interface for managing services provided by the network. This involves handling the end-to-end flow control in the network, global addressing and routing tasks, reorder buffering and data acknowledgment, and buffer management to prevent network congestion.

Each network adapter has a core interface at the core side and a network interface at the network side as shown in Figure 1.2. The core interface can be implemented as part of the PE or as a separate module. Since IP re-use is an essential component in reducing SoC development costs and timescales, many standard protocols have been developed for core interfaces such as the Open Core Protocol (OCP) [8], used in [9], the Virtual Component Interface (VCI) [10] used in the SPIN [11], and the Advanced eXtensible Interface (AXI) [12] used in [13]. AXI belongs to the Advanced Microcontroller Bus Architecture (AMBA) protocols family, which is an open standard, on-chip interconnect specification for the connection and management

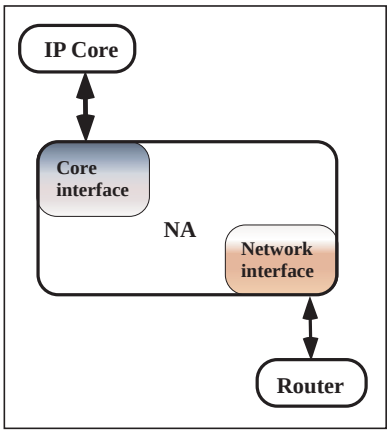


Figure 1.2: Network adapter (NA).

of functional blocks in a SoC. ARM® has recently released AMBA 4, which is the latest addition to the AMBA family adding three new interface protocols: AXI4 to maximize performance and power efficiency; AXI4-Lite and AXI4-Stream ideal for implementation in FPGA [14].

1.1.3 Networks-on-Chip Router

Routers are pivotal modules in NoC-designs [15]. Therefore, developing smart routing techniques and switching methodologies were the main objective of many researchers [16, 17]. Routers can be classified according to two different criteria [18]: the type of the switch fabric (SF) and the location of buffers and queues within the router. The second criterion classifies routers into four main types: 1. Input-queue router. 2. Output-queue router. 3. Shared-queue router. 4. Input/Output-queue router. The queue position inside the router is very important because it directly impacts the router delay, packet loss, and quality of service (QoS) [19]. Figure 1.3 shows an example of an output-queue router block diagram. Once packets arrive at the input of the router, they are stored at the input buffer, then the controller reads in

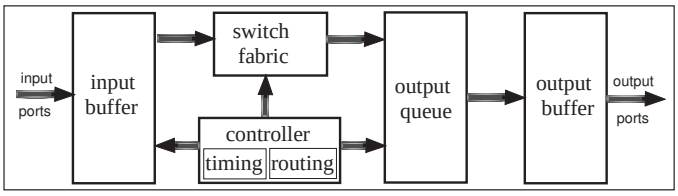


Figure 1.3: A block diagram of an output-queue router.

the packet header and configures the switch fabric to route packets to the appropriate output queue. The output queue is a group of FIFOs designed to handle multiple access requests at output ports. Then, the output buffer delivers the packet from the FIFOs to output ports. Routing tables and timing synchronization are implemented in the controller module.

1.2 Problem Statement and Research Approaches

The NoC is an emerging research area that still has several research problems needed to be addressed [20]. Unlike computer networks, NoC has shorter communication delay and limited silicon area which put more limitations on the NoC-based system design and require different approaches to overcome design challenges [21]. The design of NoCs trades off several important choices, such as network topology, router type, routing protocol. One of the most significant problems is the design of the network topology because it must be done at early design phases. This is a critical decision because at early design phases some design information, such as exact physical layout information, are not available. Also, targeting an application-specific design approach poses interesting and novel challenges to researchers in this area.

Our research addresses the following problem: Given a complex SoC application that is represented by a traffic distribution graph, it is required to find the optimum network topology/mapping that achieves minimum power consumption, minimum packet transmission delay, and maximum performability. To be able to address

this problem, we started by studying each design parameter individually. Then, we extended our study to consider all these design parameters simultaneously using a population-based stochastic/heuristic optimization technique.

Our research approach was based on developing realistic analytical models for each NoC design parameter. These models are used to evaluate the impact of changing the network topology on each performance-metric. Using these models, we proposed several design methodologies to help designers improve the system efficiency at early design phases. Following that, a joint consideration of NoC power, performability, and delay is carried out simultaneously. We used Particle Swarm Optimization (PSO) to find the optimum network topology, that achieves minimum delay and power consumption, and maximum performability for a given NoC application.

1.3 Contributions

Through this research work, several contributions have been achieved. Among these contributions are

1. Studying the impact of the network topology on system power, delay and performability using graph-theoretic concepts,

2. Modeling the power consumption of NoC routers and global interconnection links at different levels of abstractions,

3. Developing a new methodology to reduce the total power consumption of an application-specific NoC-based system by selecting the optimal network topology that matches its traffic characteristics,

4. Developing a topology-based performability model for NoC-based systems that takes into consideration the traffic figure of the target application,

5. Proposing a new methodology to improve the performability of a given NoC application at early design phases taking into consideration the possible changes in voltage swing, noise power and probability of edge failure,

6. Developing a topology-based average delay model for NoC-based systems. This contribution includes the development of a 2-D Markovian M/D/1/B queue model for NoC router, which gives an accurate representation for a queue performance when a deterministic service rate is applied, and

7. Optimizing NoC power, performability, and delay simultaneously using Particle Swarm Optimization (PSO) to find the most efficient network topology for a given application.

1.4 Dissertation Organization

This dissertation is organized as follows.

Chapter 2 analyzes the main sources of power consumption in NoC-based systems and presents analytical power models for global interconnection links and routers. It also introduces a new topology-based methodology to optimize the power consumption of complex NoC-based systems at early design phases.

Chapter 3 present a novel topology-based performability model for NoC-based systems. This model is used to perform a comparative study of nine commonly used network architectures. Based on this study, a new methodology is proposed to improve the performability of a given application at system level.

Chapter 4 presents a new analytical model for network delay using Marckov chain analysis. The proposed model is used to select the optimal topology that achieves the minimum network delay for a given application, taking into consideration the possible changes in the network traffic distribution.

In Chapter 5, a joint consideration of NoC power, performability, and delay is carried out simultaneously. We used Particle Swarm Optimization (PSO) to find the optimum network topology, that achieves minimum delay, maximum performability, and minimum power consumption, for a given NoC application.

Chapter 6 summarizes this dissertation and suggests new directions for future research.

Chapter 2

Design for Power: A Topology-Based Approach

2.1 Introduction

With SoC designs that have hundreds of PEs, implementing on-chip communication using shared buses is no longer a practical solution. To address this problem, NoC is proposed as a promising paradigm to provide a solution that achieves an efficient communication infrastructure between the PEs.

Optimizing the power consumption of NoC-based designs has become more critical with the use of high speed, complex ICs in mobile and portable applications [22,23]. Managing the power in such cases does not only target power reduction, but also ensures that all PEs receive proper and efficient amount of power to keep the application stable and reliable. Power constraints are among the major bottlenecks that limit functionality and performance of complex NoC-based designs [24, 25]. Therefore, several approaches and methodologies have been proposed to address the high power dissipation problem from both circuit and system perspectives [26–29].

At the circuit level, clock gating, voltage-islands, multiple voltage thresholds, and

transceivers' design are examples of the approaches proposed to achieve low power designs: 1) Clock gating reduces dynamic power by restricting clock distribution [30], 2) Voltage-islands use adaptive or dynamic voltage scaling to optimize supply voltage at runtime and compensate for process and temperature variations [31, 32], 3) Multiple-voltage-threshold designs use high voltage threshold cells to decrease leakage current, where performance is not critical [33], and 4) New designs of high-speed source-synchronous transceivers are introduced to speed up data rates while keeping the power consumption as low as possible [34]

At the system level, router design, first-in-first-out (FIFO) buffer resizing, and the mapping of PEs are introduced to address the high power dissipation problem; 1) Router design explores the optimum router design in terms of switching techniques, scheduling algorithms, and routing protocols [35], 2) FIFO resizing focuses on acquiring the optimum buffer size that achieves the lowest power consumption [36], and 3) Mapping of PEs depends on achieving the best matching between PEs' physical placement and their average communication traffic pattern [37].

One of the most effective approaches to address the high power dissipation problem is to select the network topology that achieves the lowest power consumption. However, at early design phases, the exact physical layout structure is not yet defined. Hence, designers do not have enough layout information to choose the most power-efficient network topology for their target applications. In this chapter, we address this problem by studying the impact of the network topology on system power consumption. We also provide designers with an efficient methodology to choose the most power-efficient architecture for a given application at early design phases.

Major contributions of this work are as follows. We analyzed the power consumption of NoC routers and global interconnection links at different levels of abstractions. The connectivity matrix concept is adopted from graph theory, with modifications to be used in the system level analysis of interconnection links. Next, 48

experiments were performed to analyze the effect of changing the number of ports per router, the operating frequency, and the router queue size on its power consumption.

We explored NoC application synthesis and mapping by studying eleven standard topologies and their irregular extensions through three mapping concepts: mapping to a standard topology [38], long-range link insertion [39], and network partitioning [40]. The target standard topologies are: Mesh [41], Torus [42], Folded Torus [43], Ring [44], Octagon (Oct) [45], Spidergon (Spider) [46, 47], Binary tree (BT), Butterfly Fat Tree (BFT) [48], SPIN [49], Hypercube (Hcube), and Star. Based on this work, we developed a new methodology to reduce the total power consumption of an application-specific NoC-based system by selecting the optimal network topology that matches its traffic characteristics. The proposed methodology was verified by experimental results and validated trough a case study.

This chapter is organized as follows. Section 2.2 highlights related work. Section 2.3 shows the power modeling and analysis of NoC-based systems at different abstraction levels. Section 2.4 discusses and explains our proposed methodology to reduce network power consumption. As a proof of concept, the proposed methodology is validated in Section 2.5 through a case study. Finally, the chapter summary is presented in Section 2.6

2.2 Related Work

The optimization of NoC power consumption has been addressed from two different perspectives. One approach is studying, analyzing, and modeling the power consumption of various NoC modules (i.e., routers, global links, etc.). Another approach is optimizing the power consumption of the network using topology-based designs [50].

In [51], a methodology was presented to automatically build the energy model of an NoC PE at the Bus Cycle Accurate (BCA) transaction level. This model allows power profiling to be performed for the entire platform at the very early stages of system design. At an intermediate abstraction level, energy models for all NoC components such as links, FIFO buffers, and routers were introduced by Bhat in [27]. In his work, Bhat showed how these models could be used to estimate the energy consumption of a complete NoC. At the circuit level, a simulation platform was implemented to trace the dynamic power consumption on switch fabrics in [52]. This trace was done with a bit-level accuracy.

On the other hand, power exploration for NoC routers was discussed in [53] for a circuit-switched router, a wormhole router, and a speculative Virtual-Channel router in a 90nm CMOS process.

The work in [37] utilized a variety of interconnect wire styles in different network topologies to achieve low-power on-chip communication. Work in [27, 54] developed analytical models for global router-to-router links and semi-global router-to-PE links to optimize power consumption.

Various tools and algorithms have been developed to choose the optimal network topology design and mapping through network partitioning [55], long-range link insertion [1], and exploring various standard topologies [38].

In this chapter, we study the effect of changing the router number of ports and queues' sizes on the total router power consumption. We also analyze the power consumption of global interconnection links from new perspectives and address the trade-off between network connectivity and power consumption for a given application. Based on this work, a new methodology is proposed to optimize the power consumption of the network using topology-based design approach. The proposed methodology introduced a new tailor-made partitioning algorithm that achieves minimum inter-partition traffic compared to other existing algorithms. The efficiency

of the proposed methodology was verified through a case study of an MPEG4 video application.

2.3 Power Analysis in NoC-based Systems

This section models and analyzes the power consumption of NoC-based systems (i.e., NoC interconnection links and routers). A brief introduction to system analysis using graph-theoretic approach is given first, then, we present a complete system-level and circuit-level analysis for global interconnection links and NoC routers.

2.3.1 Networks-on-Chips: A Graph-Theoretic Approach

The topological structure of any interconnection network can be represented by a graph. Then, using graph theory concepts, we can analyze the performance of the interconnection network from different perspectives.

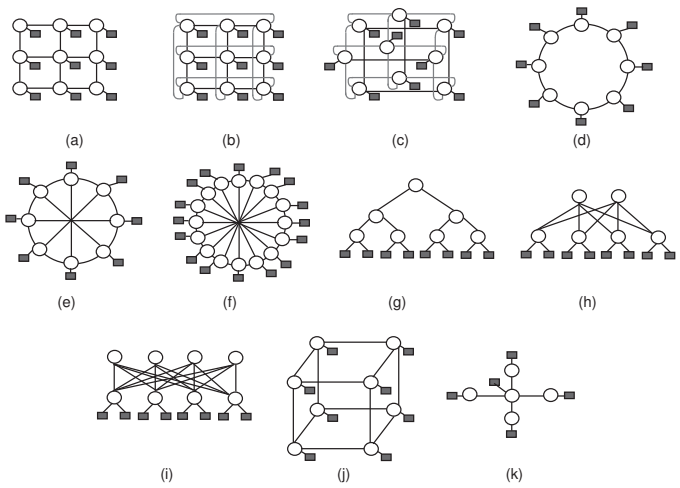


Figure 2.1: Eleven standard NoC topologies: (a) Mesh. (b) Torus. (c) Folded Torus. (d) Ring. (e) Octagon (Oct). (f) Spidergon (Spider). (g) Binary tree (BT). (h) Butterfly Fat Tree (BFT). (i) SPIN. (j) Hypercube (Hcube). (k) Star. (Routers are represented by white circles, whereas PEs are represented by dark squares.)

Figure 2.1 shows eleven standard topologies (Mesh [41], Torus [42], Folded Torus [43], Ring [44], Oct [45], Spider [46, 47], BT, BFT [48], SPIN [49], Hcube, and Star topologies). Each network topology $\mathcal{H}$ can be represented using graph theory as a graph $G = (V, E, \psi)$, where each node $v_i \in$ V represents a PE. E is a set of edges that represent the logical communication channels between PEs. ψ is the graph mapping incident function $\psi : E \rightarrow V \times V$, which maps an edge onto a pair of vertices (v_i,v_j) [56].

In the same context, the traffic distribution figure of a system (inter-module communications between all PEs in number of packets per time step) can be represented by a graph. Figure 2.2(a) shows an example of a system represented by a Traffic Distribution Graph (TDG) $G = (V, E, \psi)$. Each edge $e_{ij} \in$ E has a weight factor λ_{ij} which represents the average number of packets per time step transmitted from v_i to v_j, $1 \leq i,j \leq n$; where n is the number of PEs. This graph can also be represented in a traffic distribution matrix form (λ), as shown in Figure 2.2(b). As shown in the figure, if two vertices do not have a direct connection, a weight of zero is given to the corresponding entry in the λ matrix.

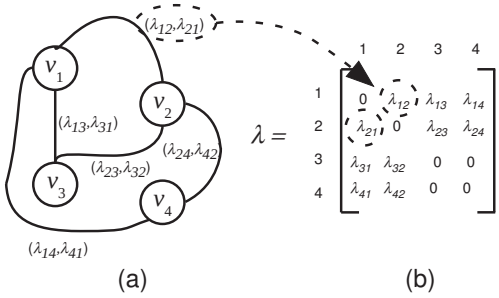


Figure 2.2: (a) An example of an average traffic distribution graph (TDG). (b) Corresponding traffic distribution matrix (λ).

In this chapter, graph-theoretic concepts are adopted with modifications to analyze and optimize the power consumption of interconnection links. The opti-

mization problem can be formulated as follows. Given a TDG $G = (V, E, \psi)$, it is required to find the optimum network topology ($\mathcal{H}$) that minimizes the network power consumption, taking into consideration the following assumptions.

1. Data communication between all PEs for a given application can be represented by a TDG.
2. Shortest path routing is used in all target topologies.
3. All global links have the same number of wires (N_{wires}) (i.e., a system with a fixed word size).
4. Network Interface (NI) units are embedded in the PEs.
5. All local (router-to-PE) interconnection links have the same length.

These assumptions can be justified as follows. 1) Data communication between PEs can be predicted based on the nature of the application and the system requirements. Therefore, it's a valid assumption to use the TDG to represent a given application, 2) Shortest path routing could be achieved if the routing protocol and QoS were designed properly. 3) The proposed model targets systems with a fixed bus width. 4) NI units has been, in most cases, embedded in the PEs to support re-usability. 5) Local links are assumed to have the same length as an initial assumption. However, the proposed methodology can handle any changes after placement and routing as shown in subsection 2.4.

2.3.2 Power Modeling of Global Interconnection Links

In this subsection, we analyze the power consumption of interconnection links at two levels of abstraction: system level and circuit level. Then, we derive a closed-form expression for the global-links' power objective function ($\mathcal{P}_{gl}$) that must be minimized to achieve an efficient network topology design.

System level modeling

Based on the target topology, and assuming shortest path routing, the power consumption of global interconnection (router-to-router) links in a target topology is measured at an abstract level by calculating the number of power units consumed to transmit all data packets given in the TDG. This measurement is done under the assumption that one unit power is consumed when a packet is transmitted over a unit length. For instance, to transmit a packet from node 3 to node 1 in Figure 2.3(a), two power units are to be consumed.

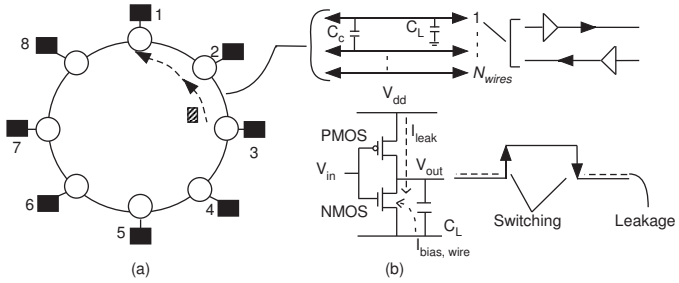


Figure 2.3: (a) Ring topology: Routers are represented by white circles and PEs are represented by dark squares. The dashed rectangular represents a packet. (b) Static and dynamic sources of power consumption.

For each network topology, a unique connectivity matrix (C) is generated [57]. This matrix represents the minimum number of links a packet goes through during its transition from the source node to the destination node. For example, the connectivity matrix $C = [c_{ij}]$ of the ring network topology shown in Figure 2.3(a) is written as follows

$$\mathrm{C} = \begin{bmatrix} 0 & 1 & 2 & 3 & 4 & 3 & 2 & 1 \\ 1 & 0 & 1 & 2 & 3 & 4 & 3 & 2 \\ 2 & 1 & 0 & 1 & 2 & 3 & 4 & 3 \\ 3 & 2 & 1 & 0 & 1 & 2 & 3 & 4 \\ 4 & 3 & 2 & 1 & 0 & 1 & 2 & 3 \\ 3 & 4 & 3 & 2 & 1 & 0 & 1 & 2 \\ 2 & 3 & 4 & 3 & 2 & 1 & 0 & 1 \\ 1 & 2 & 3 & 4 & 3 & 2 & 1 & 0 \end{bmatrix}$$

All links within standard topologies are assumed to have equal lengths. However, for topologies such as Torus, there are some connections that connect the last router in a given dimension to the first one. Those links do not have the same length as the others. To address this problem, the set of entries corresponding to those links in the connectivity matrix C is multiplied by an adjustment factor to correct the length of those links.

At the system level, the total power consumed in the global links of a network topology can be estimated from

$$\mathcal{P}_{sys} = \sum_{i=1}^{n} \sum_{j=1}^{n} \lambda_{ij} \cdot c_{ij} \cdot u_p \tag{2.1}$$

where i and j are the source and destination node indexes respectively, λ_{ij} is the average number of packets per time step associated with each logical link. c_{ij} is the minimum number of links needed to transmit these packets from their source to destination on a certain topology. u_p represents a unit power. This multiplication is performed for each entry in a given TDG. Assuming a unit power is consumed when a packet is transmitted over a unit length, the summation ($\mathcal{P}_{sys}$) represents the total power consumption of global links.

Circuit level modeling

At the circuit level, the power consumed in a global interconnect link ($\mathcal{P}_{link}$) that consists of a certain number of wires (N_{wires}) equals the summation of the dynamic and static power. As shown in Figure 2.3(b), dynamic power consumption is caused by switching power (loading and cross-coupling capacitors charging and discharging) and internal power (power consumed during switching activity at the driving gate's output due to internal short circuit). Static power consumption is mainly caused by the leakage power, irrespective of the switching activity and the state of the gate [27].

The global interconnection links power consumption can then be represented as [27]

$$\mathcal{P}_{link} = P_{switching} + P_{short} + P_{static} \tag{2.2}$$

where

$$P_{switching} = \frac{1}{2} N_{wires} V_{dd}^2 (C_L \alpha_L + C_C \alpha_C) f \tag{2.3}$$

$$P_{short} = N_{wires} \tau \alpha_L V_{dd} I_{short} f \tag{2.4}$$

$$P_{static} = N_{wires} V_{dd} (I_{bias,wire} + I_{leak}) \tag{2.5}$$

V_{dd} is the supply voltage. C_L and C_C represent self and coupling capacitance of a wire and neighbouring wires, respectively. α_L is the switching activity on a wire and α_C is the switching activity from the adjacent wires. f denotes the clock frequency, τ is the short circuit period during which I_{short} flows between source and ground. $I_{bias,wire}$ represents the current flowing from the wire to its substrate, and I_{leak} is the leakage current flowing from the source to ground regardless of the gate's state and switching activity [27]. We used the Predictive Technology Model (PTM) to obtain parameters for interconnection links and devices using the BSIM3 models in [58]. We have used the structure number 1 of the interconnect reference model, which is coupling lines above one metal ground (for top global layer), with the intermediate 0.18um technology node parameters shown in Table 2.1 [27]. From (2.1) and (2.2), and for a given TDG $G = (V, E, \psi)$, and target network topology $\mathcal{H}$, the total power consumption of the global interconnection links is given by

$$\mathcal{P}_{gl} = \frac{\mathcal{P}_{sys} \cdot \mathcal{P}_{link}}{u_p} \tag{2.6}$$

Because of the possible changes in the layout design, the proposed power modeling

Table 2.1: BSIM3 interconnect parameters.

width	0.35 um
space	0.35 um
thickness	0.65 um
$height_{ILD}$	0.65 um
k_{ILD}	3.5
length	2500 um
V_{dd}	1.8v
C_L	38.344 fF/mm
C_C	103.345 fF/mm
α_L	0.5
α_C	0.5

approach might not give precisely the amount of power dissipation after physical placement and routing. However, this could easily be addressed by using a modified connectivity matrix (C_M) that reflects the exact physical layout information after placement and routing, in case that significant layout changes occurred.

This model accurately considers the traffic switching activities and efficiently generates the dynamic power dissipation resulted from packets transfer between different hops.

For comparative analysis, we mainly care about the power dissipation of various topologies relative to each other rather than calculating the exact amount of power dissipation when a certain topology is used. Therefore, the relative power measures are used in the proposed methodology in Section 2.4 to compare the power dissipation of various topologies, for a given application, at early design phases (i.e. before physical placement and routing phases).

The choice of the target network topology $\mathcal{H}$ must be done in a way such that $\mathcal{P}_{gl}$ is a global minimum point.

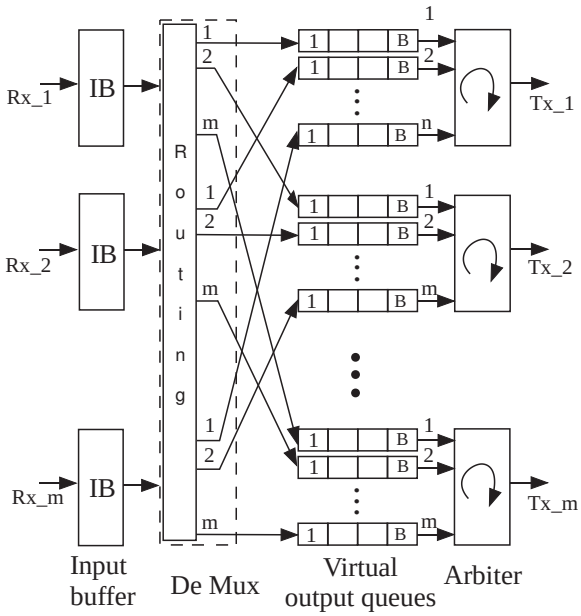


Figure 2.4: An $m \times m$ output-queue router; (Rx represents an input port, Tx represents an output port, m is the number of ports, and B is the maximum queue size.)

2.3.3 Power Modeling of NoC Routers

In this subsection, we study and analyze the power consumption of NoC routers. Figure 2.4 shows the main blocks of an m-port output-queue router, which is taken as an example through this chapter. The modeling technique adopted here could be easily applied to other router architectures. In output-queue routers, packets arrive at the input of the router asynchronously. Then, the packet header of each incoming packet, which contains the destination address, is examined by the routing module. Based on the routing table, the demultiplexers are enabled to direct the incoming packets to the corresponding output queues. As shown in Figure 2.4, there are m queues for each output port serving as FIFO buffers. Finally, the output arbiter uses a round robin scheduling algorithm to serve backlogged queues one after another at

Table 2.2: Power consumption of 4-,5-,6-,7-, and 8-port NoC routers when implemented in 0.18μm technology for various operating frequencies.

Frequency	Total Power ($\mathcal{P}_r$)				
	4-port	5-port	6-port	7-port	8-port
200 MHz	32.019 mw	48.440 mw	68.041 mw	86.709 mw	117.173 mw
100 MHz	12.793 mw	19.380 mw	27.229 mw	34.706 mw	46.901 mw
50 MHz	6.410 mw	9.705 mw	13.635 mw	17.372 mw	23.481 mw
25 MHz	3.211 mw	4.862 mw	6.832 mw	8.705 mw	11.762 mw
10 MHz	1.293 mw	1.963 mw	2.747 mw	3.505 mw	0.004726 mw
1 MHz	0.134774 mw	0.202701 mw	0.285020 mw	0.379629 mw	0.487467 mw

each output port in a fixed order.

To analyze the power consumption of the above NoC router, a set of synthesized routers (with different number of ports and queue sizes) are modeled in VHDL and synthesized using 0.18μm technology. To calculate the power consumption, 48 experiments were carried out using Synopsys® Design Compiler™, VHDLSIM™, and Power Compiler™ tools (based on switching activity) to measure the power consumption of 4-, 5-, 6-, 7-, and 8-port routers at various operating frequencies [59]. Figure 2.5 shows the methodology used to measure the power consumption of a Register Transfer Logic (RTL) design using the standard switching activity file format. The `analyze` and `elaborate` commands read the RTL design into an active memory and convert it to a technology-independent format called the GTECH design. This is done using Design Compiler™ tool, which is the core of the Synopsys® synthesis software. Then, a forward-annotated Switching Activity Interchange Format (SAIF) file is generated using the `rtl2saif` command of Synopsys®. This forward annotated file contains directives that determine which design elements to be traced during simulation. The forward-annotated SAIF file is fed into the simulator with the VHDL testbench and technology files to generate a back-annotated SAIF file. The back-annotated SAIF file contains information about the switching activity

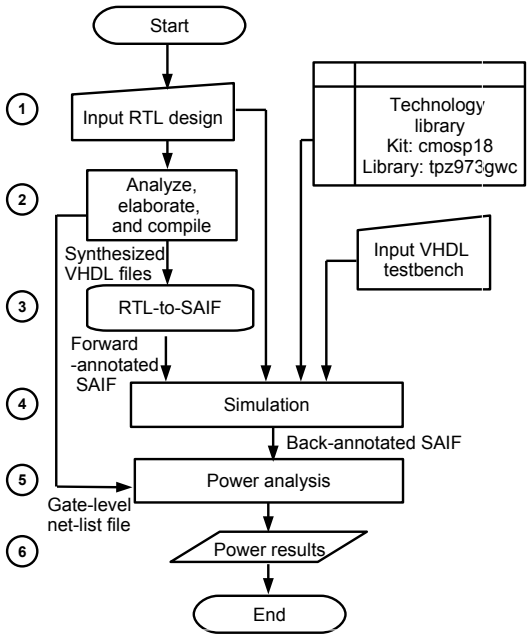


Figure 2.5: A flowchart of power measurement of NoC routers using Synopsys® tools.

of the synthesis-invariant elements in the design. Then, the back-annotated SAIF is used with the gate-level net-list file (Data-Base (DB) file) produced by the Design Compiler™ to calculate the power consumption of the router. Power Compiler™ is used to calculate the power, do power optimization, and report the power results. This experiment is repeated for various operating frequencies, as shown in Table 2.2. All testbench files used in these experiments are prepared to simulate a complete flow of one packet per port. The packet flow path inside the router starts from the input buffer and ends at the output multiplexer. Results show the significant effect of changing the number of ports on the power consumption. For instance, changing the number of ports from 4 to 8, results in an increase of 265.95% in the power consumption for 200 MHz operating frequency.

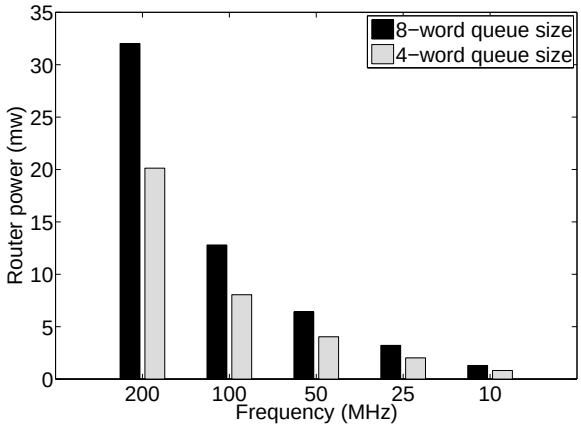


Figure 2.6: Power consumption versus operating frequency for one packet per port transition of 4-word and 8-word queue size routers when implemented in 0.18 μm technology.

We also studied the effect of changing the queue size from 8-word to 4-word length for a 4-port router. For instance, Figure 2.6 shows a reduction percentage of 37.14% of the total router power consumption (for one packet per port transition), when changing the queue size from 8-word to 4-word length for 200 MHz.

The experimental results show that the number of ports and queue size, which have direct relation with the topology used, must be optimally chosen to minimize the overall network power consumption. Results from above experiments are used to build our database library for the NoC routers' power consumption ($\mathcal{P}_r$) for various number of ports, queue sizes, and operating frequencies.

The power consumption for the whole network including global links and routers can be represened by ($\mathcal{P}_t$) as show in (2.7)

$$\mathcal{P}_t \;=\; \sum_{i=1}^{n_r} \mathcal{P}_{ri} \;+\; \mathcal{P}_{gl} \tag{2.7}$$

where n_r is the number of routers. Acquiring the minimum value of $\mathcal{P}_t$ depends on

many design factors such as traffic distribution, number of global interconnection links, length of links, number of routers, number of ports per router, and queues' sizes.

In the next section, we show our proposed methodology to reduce the power consumption of NoC-based designs.

2.4 Proposed Methodology

The proposed methodology to reduce the power consumption merges three mapping concepts (mapping to a standard topology, long-range link insertion, and network partitioning) in one procedure aiming at minimizing the total network power consumption. Figure 2.7 shows a flowchart of the proposed methodology. The following subsections explain this methodology in details.

2.4.1 Steps 1-2: Convert TDG into a λ Matrix

The first step of the proposed methodology to minimize the power consumption is to consider the TDG as the main design input. Then, from the TDG, a $\lambda = [\lambda_{ij}]$ matrix is generated as a mathematical representation of the graph, where λ_{ij} is the average number of packets per time step associated with each logical link.

2.4.2 Step 3: Apply Network Partitioning

Network partitioning is used to divide the TDG into two partitions aiming at minimizing the cost of the communication over the partition boundaries [40]. We chose to partition the graph into only two partitions because we consider a future step in our proposed methodology, which is the long-range link insertion. This step is designed to compensate the delay of packets' transmission between the two partitions with a minimum area overhead cost. If the graph is partitioned into more than

two partitions, extra long-range links will be needed to counterbalance the packet transmission delay between different partitions, which increases the area overhead significantly.

Based on the given TDG, the third step, in the proposed methodology, applies multi-constraint graph partitioning to minimize the number of cuts in the edges of the partition, reduce the imbalance in the weight of each partition, and minimize the inter-partition traffic. A tailor-made algorithm is developed to perform this partitioning step. In this subsection, we discuss and explain the theoretical concepts of this algorithm. The discussion is extended in Section 2.5 to compare its efficiency to other existing algorithms, such as spectral, linear, and Kernighan-Lin, through a case study [60].

The main idea behind this partitioning algorithm is to analyze the nodes' level of connectivity and, based on the analysis, to partition the graph into two separate partitions. Before we start our discussion, we need to define two terms.

- *Connected nodes:* nodes that have direct communication paths (one-hop communication).
- *Semi-connected nodes:* nodes that can communicate through an intermediate node (two-hop communication).

Using graph-theoretic concepts, all connected or semi-connected nodes are assigned to the same partition. Then, the graph is divided into separate partitions based on the above assignment. Following that, a redundancy cancelation and refinement processes are performed at the end to generate the final partitions. The proposed algorithm for partitioning consists of seven main steps and can be explained by the flowchart shown in Figure 2.8.

First, we consider the TDG as the main input. Second, the TDG is represented by an adjacency matrix [57], which is denoted by $A = [a_{ij}]$ and given by

$$\begin{aligned} a_{ij} &= 1, \text{ if the edge } e_{ij} \text{ exists in the TDG} \\ a_{ij} &= 0, \text{ if the edge } e_{ij} \text{ does not exist in the TDG} \end{aligned}$$

The adjacency matrix shows the direct (one-hop) paths between pairs of nodes in the network. Third, graph-theoretic concepts are adopted, with modifications, to find the nodes that have indirect communication paths through an intermediate node. Raising the adjacency matrix (A) to a power of two gives the number of two-step indirect paths between pairs of nodes in the network [44]. However, the resulting matrix contains redundant entries because of the contribution of the self-loop and multiple passes through a node entries [44]. Therefore, Algorithm 1 is developed to address the redundant entries problem and generate an $\hat{A} = [\hat{a}_{ij}]$ matrix where

$$\begin{aligned} \hat{a}_{ij} &= 1, \text{ if } v_i \text{ and } v_j \text{ are semi-connected in the TDG} \\ \hat{a}_{ij} &= 0, \text{ otherwise} \end{aligned}$$

In Algorithm 1, the $\hat{A}$ matrix is made up of the dot product of the i^{th} row and the j^{th} column of the A matrix under two constraints:

1. $\hat{a}_{ij} \in \{1, 0\}$
2. $\hat{a}_{ii} = 0$

The generated $\hat{A}$ matrix accurately represents the existence of semi-connected nodes without any redundant or self-loop entries.

Fourth, the disconnectivity matrix D is generated to represent the nodes that are not directly connected or semi-connected. Let J be the all-ones matrix and I is

Algorithm 1 Calculation of the $\hat{A}$ matrix

Require: A is an n × n square matrix

```
for i = 1 to n do
  for j = 1 to n do
    if A(ij) = 0 and i ≠ j then
      if Σ_{k=1}^{n} A(i,k) · A(k,j) ≥ 1 then
        Â(ij) = 1
      else
        Â(ij) = 0
      end if
    else
      Â(ij) = 0
    end if
  end for
end for
```

the identity matrix, the $D = [d_{ij}]$ matrix is given by

$$D \;=\; J - I - A - \hat{A} \tag{2.8}$$

For each $d_{ij} \in D$,

$$d_{ij} \;=\; 1, \text{ if } v_i \text{ and } v_j \text{ are not connected or semi-connected}$$

$$d_{ij} \;=\; 0, \text{ otherwise}$$

Fifth, the graph is divided into two partitions based on the entries of the D matrix. Starting from the first row in the D matrix, nodes v_i and v_j are assigned to different partitions if $d_{ij} = 1$. The assignment process is performed in a row by row basis until all nodes are assigned to one of the two partitions. Let X be the main set of all nodes and X_1 and X_2 be partitions of X. Algorithm 2 explains this partitioning step.

Algorithm 2 Partitioning a graph based on the D matrix

Require: $X = \{v_1, v_2, v_3, ..., v_n\}$
Require: $D = [d_{ij}]$ is an n × n square matrix
1: **for** $i = 1$ to n **do**
2: **for** $j = 1$ to n **do**
3: **if** $d_{ij} = 1$ and $X \neq \phi$ **then**
4: $X_1 = X_1 \cup v_i$
5: $X_2 = X_2 \cup v_j$
6: $X = (X \cap (X_1 \cup X_2))^c$
7: **end if**
8: **end for**
9: **end for**

Sixth, a redundancy cancelation process is performed to remove the redundant nodes from one of the partitions. This is done based on the entries in the traffic distribution matrix (λ). A node is assigned to one of the two partitions if the traffic density between this node and other nodes in this partition (summation of λ_{ij}) is greater than the other one.

Finally, a refinement process is performed to check if a node v_i is an orphan node, i.e. it has no connections with any node in its current partition whereas it is connected to other nodes in another partition. In such a case, the node v_i is moved to the other partition.

The efficiency of the proposed partitioning algorithm is verified through a case study in Section 2.5.

2.4.3 Steps 4-6: Iterative Topology Selection

The fourth step of the proposed methodology to reduce the network power consumption is to select an initial topology, perform an initial mapping, and generate an initial traffic distribution matrix (λ) for each partition. Then, using the connectivity matrix concept discussed in Subsection 2.3.2, a unique connectivity matrix (C) is

generated for each one of the eleven topologies mentioned in Section 2.1. These matrices are used to represent all possible mappings of each partition. The total power consumption ($\mathcal{P}_t$) for global links and routers is calculated for each one of those topologies using (2.7), as shown in the fifth step in Figure 2.7.

The sixth step minimizes the power consumption of each partition by re-mapping all PEs such that those who have the highest traffic rates are associated with the shortest distances. An exhaustive search algorithm is used to re-allocate PEs to different positions within each one of the eleven topologies. The re-mapping process is done by analyzing all numbers in the λ matrix. Then, all possible changes in rows and columns are done to re-arrange the matrix, and hence, re-map the PEs such that the highest traffic rates (numbers) are associated with shortest distances (neighboring nodes). Following that, a new $\mathcal{P}_t$ (which is the objective function) is calculated using (2.7), to get the new power consumption, and compared to previously obtained values. This process is repeated until the minimum power is obtained. Based on the results obtained, the topology $\mathcal{H}$ that has minimum power $\mathcal{P}_t$ is selected for each partition. Then, one node is selected from each topology to act as a connecting node. The choice of these two nodes ($v_i \in X1,\ v_j \in X2$) is done such that the average number of packets (λ_{ij}) is a maximum value for all node pairs.

2.4.4 Steps 7-8: Long-Range Link Insertion

The seventh step is to add a long-range link between two selected nodes in each topology. The choice of target nodes (v_i, v_j) is done such that the communication cost between these two nodes (k) is the maximum value for all $G(V, E, \psi)$, $1 \leq i,j \leq n$. k is given by

$$k = \max(\lambda_{ij} \cdot C_{ij}),\ v_i, v_j \text{ are non-neighboring nodes.} \tag{2.9}$$

Finally, the total power is evaluated after applying the long-range link insertion technique and compared to its original value. The modified topology is selected only if the power improvement ratio exceeds the ratio of router-area overhead. Area overhead analysis of long-range link insertion is discussed in [1], where Ogras *et al.* defined the maximum amount of permissible overhead due to the addition of long-range links.

2.5 Performance Evaluation by Experimentation

We validate the proposed methodology through an experimental case study. The MPEG4 core discussed in [61] is taken as an example to evaluate the performance of the proposed methodology. The same methodology can be applied to any application, that could be represented by a TDG, to select the optimum topology. The power calculations and matrices are generated using Matlab®. This section is organized as follows. Subsection 2.5.1 applies the first two steps in the proposed methodology on the MPEG4 core. Subsection 2.5.2 explains the partitioning of the MPEG4 core based on the proposed algorithm. Following that, Subsection 2.5.3 shows the results of applying steps 4-8 of the proposed methodology on the MPEG4 core. Finally, the efficiency of the proposed methodology is evaluated in Subsection 2.5.4 through discussing the experimental results.

2.5.1 Steps 1-2: Convert TDG into a λ Matrix

Figure 2.9(a) shows a TDG typical for video applications (MPEG4 core) [61]. The numbers written on the arrows are the average number of packets/time step transmitted and the numbers written on the circles represent PEs' numbers. From this TDG, the traffic distribution matrix (λ), which represents the initial mapping of

the MPEG4, shown in Figure 2.9, is given by

$$\lambda = \begin{bmatrix} 0 & 0 & 0 & 0 & 190 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 60 & 40 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 600 & 40 & 0 & 0 & 0 & 0 & 0 & 0 \\ 190 & 0.5 & 60 & 600 & 0 & 0 & 0 & 0 & 0.5 & 910 & 32 & 0 \\ 0 & 0 & 40 & 40 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 250 & 0 & 670 & 173 & 500 \\ 0 & 0 & 0 & 0 & 0 & 0 & 250 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 910 & 0 & 670 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 32 & 0 & 173 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 500 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

2.5.2 Step 3: Apply Network Partitioning

In this subsection, we follow the steps of the proposed partitioning algorithm in Subsection 2.4.2. In Figure 2.9, we compare four different partitioning solutions for the MPEG4 core. Our partitioning algorithm divides the graph into two partitions, as shown in Figure 2.9(a). The algorithm is developed using Matlab® to acquire the minimum number of cuts in the edges needed to divide the graph into two partitions. This is done according to three constraints: 1) achieve balanced partition weights (in terms of number of nodes ± 2), 2) minimize the inter-partition traffic, and 3) accept only one inter-partition edge's weight to be above-average. The third condition is accepted because the above-average edge will be recovered in the seventh step of the proposed methodology, which is applying long-range link insertion, as shown in Figure 2.7.

The adjacency matrix (A) of the MPEG4 shown in Figure 2.9 is given by

$$\mathrm{A} = \begin{bmatrix} 0&0&0&0&1&0&0&0&0&0&0&0 \\ 0&0&0&0&1&0&0&0&0&0&0&0 \\ 0&0&0&0&1&1&0&0&0&0&0&0 \\ 0&0&0&0&1&1&0&0&0&0&0&0 \\ 1&1&1&1&0&0&0&0&1&1&1&0 \\ 0&0&1&1&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&1&0&1&1&1 \\ 0&0&0&0&0&0&1&0&0&0&0&0 \\ 0&0&0&0&1&0&0&0&0&0&0&0 \\ 0&0&0&0&1&0&1&0&0&0&0&0 \\ 0&0&0&0&1&0&1&0&0&0&0&0 \\ 0&0&0&0&0&0&1&0&0&0&0&0 \end{bmatrix}$$

Then, for semi-connected nodes, the generated $\hat{A}$ matrix for the MPEG4 TDG is

$$\hat{A} = \begin{bmatrix} 0&1&1&1&0&0&0&0&1&1&1&0 \\ 1&0&1&1&0&0&0&0&1&1&1&0 \\ 1&1&0&1&0&0&0&0&1&1&1&0 \\ 1&1&1&0&0&0&0&0&1&1&1&0 \\ 0&0&0&0&0&1&1&0&0&0&0&0 \\ 0&0&0&0&1&0&0&0&0&0&0&0 \\ 0&0&0&0&1&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&1&1&1 \\ 1&1&1&1&0&0&0&0&0&1&1&0 \\ 1&1&1&1&0&0&0&1&1&0&1&1 \\ 1&1&1&1&0&0&0&1&1&1&0&1 \\ 0&0&0&0&0&0&0&1&0&1&1&0 \end{bmatrix}$$

The next step is to generate a matrix D which represents all nodes that are not directly connected or semi-connected. For the MPEG4 graph, D is written as

$$\mathrm{D} = \begin{bmatrix} 0&0&0&0&0&1&1&1&0&0&0&1 \\ 0&0&0&0&0&1&1&1&0&0&0&1 \\ 0&0&0&0&0&0&1&1&0&0&0&1 \\ 0&0&0&0&0&0&1&1&0&0&0&1 \\ 0&0&0&0&0&0&0&1&0&0&0&1 \\ 1&1&0&0&0&0&1&1&1&1&1&1 \\ 1&1&1&1&0&1&0&0&1&0&0&0 \\ 1&1&1&1&1&1&0&0&1&0&0&0 \\ 0&0&0&0&0&1&1&1&0&0&0&1 \\ 0&0&0&0&0&1&0&0&0&0&0&0 \\ 0&0&0&0&0&1&0&0&0&0&0&0 \\ 1&1&1&1&1&1&0&0&1&0&0&0 \end{bmatrix}$$

Based on the entries of the D matrix, the graph is divided into two partitions using Algorithm 2. First, the partitioning process is performed until all nodes are assigned to one of the two partitions. (Six steps are needed in the MPEG4 case, as shown in Table 2.3.), Then, redundancy cancelation and refinement processes are performed

to remove repeated nodes from one of the two partitions and acquire the optimal partitioning, as shown in Table 2.3.

Table 2.3: Partitioning and refinement of the MPEG4 core shown in Figure 2.9(a) based on D entries using Algorithm 2.

Partitioning steps	Nodes in partition 1	Nodes in partition 2
First row entries	1	6,7,8,12
Second row entries	1,2	6,7,8,12
Third row entries	1,2,3	6,7,8,12
Fourth row entries	1,2,3,4	6,7,8,12
Fifth row entries	1,2,3,4,5	6,7,8,12
Sixth row entries	1,2,3,4,5,6	6,7,8,9,10,11,12
Redundancy cancelation	1,2,3,4,5,6	7,8,9,10,11,12
Refinement	1,2,3,4,5,6,9	7,8,10,11,12

In order to verify the efficiency of our partitioning algorithm, we used Chaco 2.0 [60], a software package designed to partition graphs, to perform other partitioning algorithms, as shown in Figure 2.9(b,c,d). Table 2.4 shows a comparison between the proposed algorithm and three other partitioning methods with respect to minimum number of cuts in edges [62]. The proposed algorithm achieves a minimum number of cuts in the edges for the MPEG4 application.

Table 2.4: Results of graph partitioning using four different methods.

Partitioning	Proposed algorithm	Spectral	Kernighan-Lin	Linear
Number of cuts in edges	2	5	7	3

2.5.3 Steps 4-8: Topology Selection and Long-Range Link Insertion

After applying the graph partitioning algorithm, the λ matrix is divided into two sub-matrices with connection nodes at the PEs numbered 5 and 10. The dashed

Table 2.5: Results of graph partitioning using four different methods before and after applying long-range link insertion.

Partitioning algorithm	Proposed algorithm		Spectral		Kernighan-Lin		Linear	
	before	after	before	after	before	after	before	after
Inter-partition traffic	942	32	1100.5	500.5	596	406	942.5	32.5
Number of cuts in edges	2	1	5	4	7	6	3	2

line in Figure 2.9(a) shows the boundary between the two partitions generated after this step.

An exhaustive search is done to explore all possible mappings for each one of the eleven topologies. Then, the topology/mapping pair that has the minimum $\mathcal{P}_t$ is selected for each partition. Figure 2.10 shows the mapping of MPEG4 application to a combination of star and ring topologies. In this case study, a ring topology is selected for the first PE-set (1,2,3,4,5,6,9) and a star topology is chosen for the second PE-set (7,8,10,11,12). The connection nodes (nodes that connect the two partitions) are found to be nodes 5 and 10. Finally, the long-range link insertion step is applied to connect nodes (3,5) and (5,11), as shown in Figure 2.10.

In order to show the compatibility of our partitioning algorithm with the proposed methodology to reduce the power consumption, Chaco 2.0 software program [60] is used to obtain the inter-partition traffic and the number of cuts in edges for various partitioning algorithms. Table 2.5 shows a comparison between various algorithms before and after applying the long-range link insertion [62].

From this table we can notice that, although Kernighan-Lin algorithm is known to be one of the most powerful partitioning algorithms, its goal is to minimize the total weight of all edge cuts but does not minimize the number of edges cut. That can be clearly observed from the results in Table 2.5 as Kernighan-Lin gives the minimum inter-partition traffic before applying the long-range link insertion step.

However, the proposed partitioning algorithm takes into consideration minimiz-

ing the number of cuts in edges, which fits the proposed methodology. Therefore, it gives the minimum inter-partition traffic and number of cuts in edges after applying the long-range link insertion, which verifies its compatibility with the proposed methodology.

2.5.4 Results

To evaluate the efficiency of the proposed methodology, experiments were carried out to analyze the power consumption of the global links and routers individually. Then, the power consumption of the whole network was measured and comparisons were made between the selected network topology (ring+star) and other network topology architectures for the MPEG4 application. We discuss the results of these experiments in this subsection.

Global-Links

We started with analyzing the global-links' power consumption. As shown in Figure 2.11(a), the generated topology (ring+star) achieved the minimum power consumption over standard topologies for different operating frequencies. However, increasing the frequency directly increases the dynamic power consumption. This can be explained from (2.3) and (2.4). Also, the maximum power consumption was obtained when the MPEG4 core is mapped to a BT topology. The reason behind this result is that the BT architecture forces packets to go through longer paths as the number of levels in the tree increases. This is an example of the effectiveness of the proposed methodology.

In Figure 2.11(b), the generated topology achieved minimum global links' power consumption over standard topologies for different technology implementations (7.32% less than Octagon (Oct), which has the lowest power consumption) for 100

MHz operating frequency. The main reason behind this improvement is the optimum choice of 1) the connection nodes from both star and ring topologies and 2) the long-range link insertion location. Scaling down the target technology from 0.18μm to 90nm reduces the power consumption by 20.93% in the case of (star+ring) because the target V_{dd} for 90nm is 1.2V, whereas it is 1.8V for 0.18μm. At the same time, the power consumption is proportional to the supply voltage, as given by (2.2).

Routers

We explained in Table 2.2 and Figure 2.6 the significant effect of changing the number of ports and queue sizes on the router power consumption. Here, we first highlight the routers' architecture in different topologies, then extend the previous analysis to measure the router leakage power consumption when the number of ports or the queue size is changed.

Table 2.6 shows a comparison between different network topologies with respect to their architecture requirements. The comparison is done when 12 PEs are to be connected, which matches our case study, and shows their corresponding routers' power consumption for 500MHz operating frequency. The routers' power consumption is measured using Synopsys® tools as shown in Subsection 2.3.3. As we can see, BFT needs only 5 routers to connect 12 PEs. However, three of these routers are 7-port type, which increases the overall area and power significantly. On the other hand, the selected topology (ring+star) requires 12 routers, but nine of them are 2- and 3-port type. Although the routers' power consumption of the generated topology (star+ring) is not the minimum, the total network power turns to be the minimum when the global-links' power is added as explained in the next subsection.

To further study the impact of changing the number of ports and the queue size, other experiments are done to highlight their effect on the cell leakage power of the

Table 2.6: A comparison of the architecture requirements for different network topologies when 12 PEs are to be connected and their corresponding routers' power consumption.

Topology	No. of routers	2-port	3-port	4-port	5-port	6-port	7-port	12-port	Router power consumption (w)
Mesh	12	0	4	6	2	0	0	0	0.835
Torus	12	0	0	10	2	0	0	0	0.835
Folded	12	0	0	10	2	0	0	0	0.835
Ring	12	0	12	0	0	0	0	0	0.769
Octagon	12	0	0	4	8	0	0	0	1.0315
Spider	12	0	0	12	0	0	0	0	0.769
BT	11	1	10	0	0	0	0	0	0.705
BFT	5	2	0	0	0	0	3	0	0.648
SPIN	6	0	3	0	0	0	3	0	0.713
Hcube	12	0	0	8	4	0	0	0	0.900
Star	13	12	0	0	0	0	0	1	1.284
Ring+star	12	2	7	1	0	1	0	0	0.777

router when implemented on 0.18μm technology.

Table 2.7: Cell leakage power consumption of 4-, 5-, 6-, 7-, 8-port (8-word queue size), and 4-port (4-word queue size) routers when implemented in 0.18μm

4-port	5-port	6-port	7-port	8-port
8.1104 μw	12.704 μw	18.233 μw	24.654 μw	31.970 μw

The results are shown in Table 2.7. As we can see from the table, changing the number of ports from 4 to 8 for 8-word queue size router increases the cell leakage power significantly (by 294.2%). The main reason behind this is adding memory elements. For each extra port, more queues are to be added, as shown in Figure 2.4. The same effect can be noticed when minimizing the queue size by half (from 8 to 4 word-length in this experiment). This minimization reduces the overall power consumption of the router by 39.2%.

Network Power Analysis

Here, we evaluate by experiments the effectiveness of the proposed methodology. Figure 2.12 shows a comparison between the power consumption of the generated topology (ring+star) and other standard topologies for MPEG4 implementation. Although the routers' power is not the minimum for the generated network (ring+star), it achieves the minimum total network power consumption over others (8.55% less than Spider, which has the lowest power consumption). The reason behind this is the significant contribution of the global-links' power, which is aimed to be the minimum, to the total network power.

Another evaluation metric is the average number of hops. Figure 2.13(a) shows a comparison between the generated topology (ring+star) and other standard topologies for MPEG4 implementation with respect to average number of hops. The results of the comparison show the efficiency of the proposed methodology in choosing an architecture that has a minimum average number of hops compared to standard topologies and other implementations. The average number of hops of the ring+star topology is 10.8% less than Oct, which has the lowest number of hops, and 50.8% less than the one proposed in [61]. The average number of hops ($\mathcal{C}_{avg}$) is a reference metric to measure the delay and power consumption and is given by

$$\mathcal{C}_{avg} \;=\; \frac{C_a}{N_a} \tag{2.10}$$

where

$$C_a \;=\; \sum_{i=1}^{n} \sum_{j=1}^{n} C_{ij} \;\forall\; \lambda_{ij} \;\neq\; 0 \tag{2.11}$$

and N_a is the number of elements in the traffic distribution matrix (λ) that do not equal zero.

Figure 2.13(b) shows a comparison between the generated topology, ring+star

(R+S), and other standard and previously proposed topologies for MPEG4 with respect to number of global links. It can be noted that the number of global links for the generated topology is 56.25% less than the one proposed in [37] for the same application. This is due to the fact that we considered a wide number of standard topologies (eleven) and their extensions to solve the optimization problem.

2.6 Chapter Summary

A new methodology to acquire the optimum architecture that achieves the lowest power consumption for a given application, in comparison with standard topologies, is presented in this chapter. The proposed topology-based methodology explores eleven standard topologies plus their possible extensions through network partitioning and long-range link insertion. A new partitioning algorithm is presented and its performance is evaluated by experimentation. Circuit and system analysis for global interconnection links and routers are performed to model the power consumption of NoC-based designs. The efficiency of the proposed methodology is validated through a case study. Experimental results for an MPEG4 video application show 8.55% improvement in the power consumption and 10.80% reduction in the average number of hops compared to the best known standard topology mapping.

This work has been published in part in the proceedings of the International Conference of Circuits and Systems (ISCAS'08) [63] and in full at Elsevier Journal of Microprocessors and Microsystems [64].

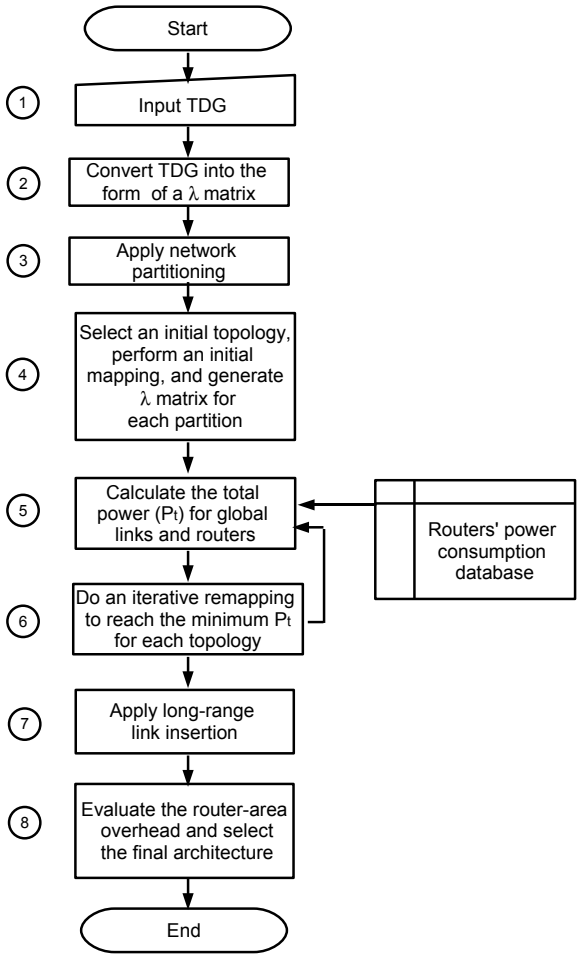


Figure 2.7: Proposed methodology to minimize the power consumption.

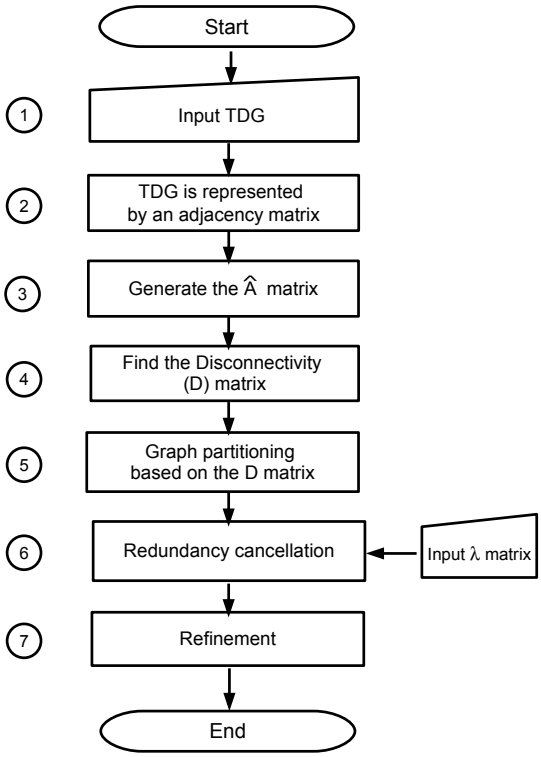


Figure 2.8: A flowchart showing the proposed graph partitioning steps.

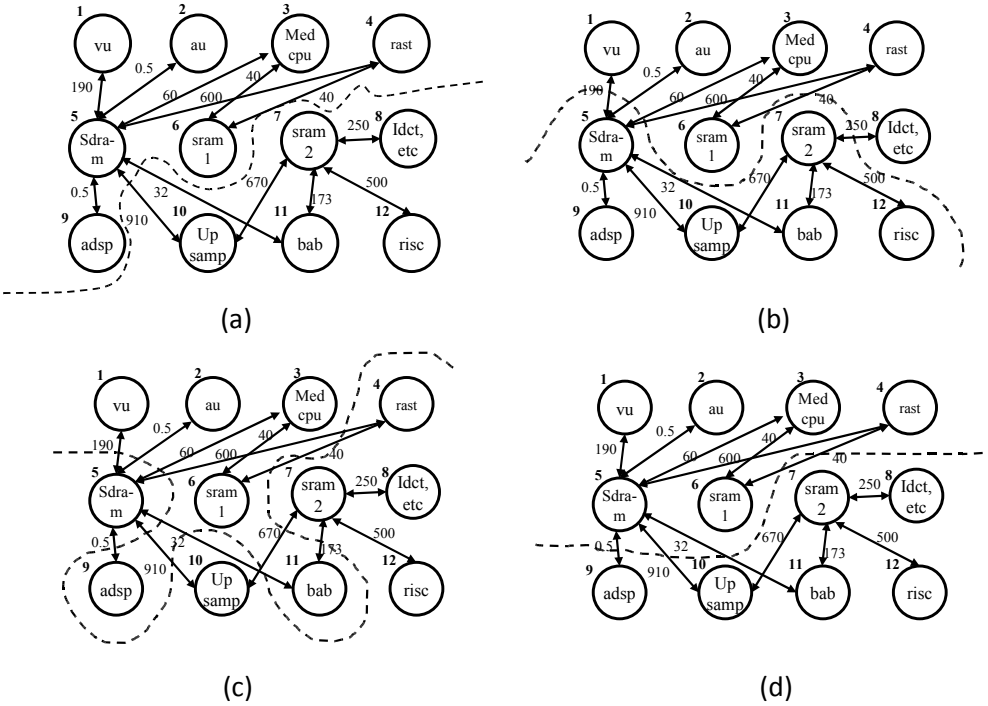


Figure 2.9: MPEG4 core TDG: The dashed line shows the partitioning boundary after applying graph partitioning. The numbers written on the arrows are the average number of packets/time step transmitted and the numbers written on the circles represent PEs' numbers. Partitioning using (a) our proposed algorithm, (b) Spectral partitioning, (c) Kernighan-Lin partitioning, and (d) Linear partitioning.

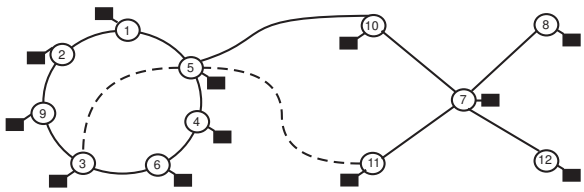


Figure 2.10: MPEG4 application mapping to a combination of star and ring topologies. Routers are represented by white circles, PEs are represented by dark squares and the dashed line is the long-range link. The numbers written inside the circles represent PEs' numbers.

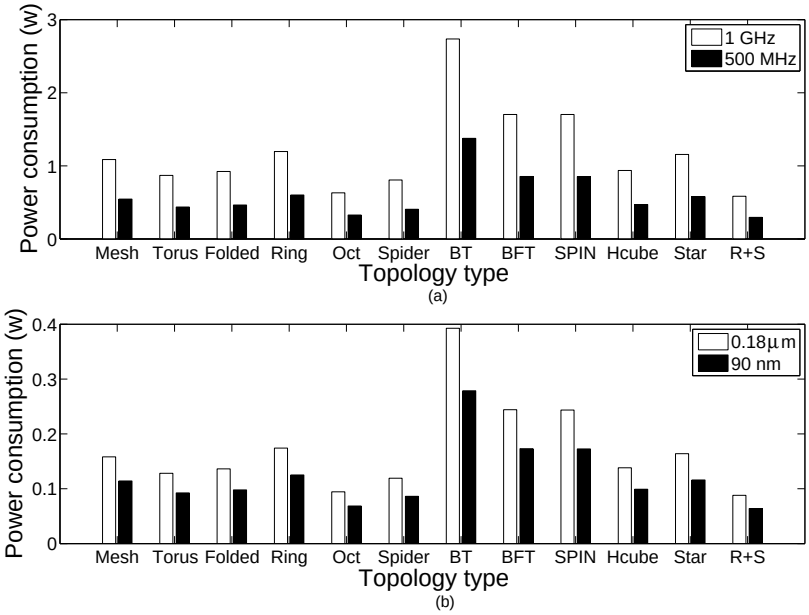


Figure 2.11: Comparisons between the power consumption of the global links in the generated topology (ring+star) and other standard topologies for MPEG4. (a) Power consumption versus topology type for different frequencies (for 0.18μm technology). (b) Power consumption versus topology type for different technologies (for 100 MHz operating frequency).

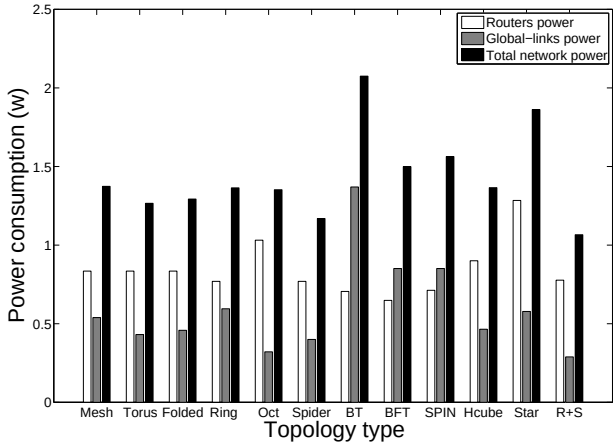


Figure 2.12: Comparisons of the power consumption between the generated topology, ring+star (R+S), and other standard topologies for MPEG4.

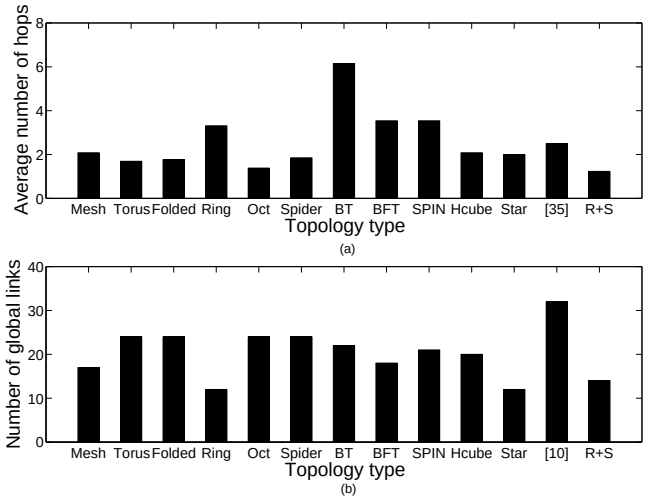


Figure 2.13: Comparisons between the generated topology, ring+star (R+S), and other standard and previously proposed topologies for MPEG4. (a) Average number of hops versus topology type. (b) Number of global links versus topology type.

Chapter 3

Design For Performability: A Topology-based Approach

3.1 Introduction

In the last few years, NoC paradigm appeared as a promising solution for the inter-communication of SoCs [65] and complex designs such as cellular nonlinear network (CNN) applications [66]. The migration from a bus-based design to a network-based one opens the door for developing new techniques to improve systems' performance and reliability, and hence, improve their performability. The concept of system performability was primarily introduced by J. F. Meyer in 1978 in his evaluation of an aircraft control system [67]. He defined performability as a composite measure of performance and dependability. Dependability is a marginal term which includes various metrics such as reliability, availability, and security. Since that time, performability analysis is used, in most cases, to measure the performance and reliability in a composite way. This measurement is proven to be essential to properly assess the effectiveness of complex multiprocessor systems [68]. This is done by evaluating the ability of a system to successfully complete a specific function in

a finite time interval [68]. For NoC-based systems, performability could be modeled at different levels of abstraction. In the same context, improving systems performability could be achieved at different design phases.

3.1.1 Sources of Errors in NoC-based Systems

The continuous scaling of technology, shrinking of transistor dimensions, and lowering supply voltage result in higher sensitivity to neutron and alpha particles, leading to significantly higher soft error rates (SER) [69]. In deep sub-micron circuits, since the capacitance associated with circuit nodes is very small, non-negligible disturbance can be originated when energized particles strike a circuit. For instance, for 3.3 V technology, the disturbance noise could reach a level 21% larger than a normal swing. Thus, to restore the correct value of the struck node, the transistor will take more time to suppress the charge-drift process [70]. This problem becomes more critical with high speed designs, such as advanced SoC-based designs, since the noise voltage pulse may become comparable to the gate propagation delay, which might cause an erroneous transition on the output of combinational logic circuits.

Several techniques have been proposed to mitigate the impact of errors in complex VLSI designs. However, with the recent migration to NoC as an emerging communication infrastructure for complex SoC deigns, new sources of errors are considered. The impact of permanent, transient, and intermittent faults is becoming more significant due to many factors, such as crosstalk, electromagnetic interference, alpha particle hits, and cosmic radiation. These phenomena can alter the synchronization and functionality of NoC-based systems and thus degrade their QoS features and, in some cases, lead to failures for the whole system [71].

In NoC-based designs, other problems might occur such as the break-down of links and/or routers. In such cases, fault-tolerant techniques are used to provide

substitute routing paths/services to preserve the network QoS [72].

3.1.2 Design for Performability

The NoC design approach depends mainly on the target application requirements and design constraints, such as low power consumption, limited silicon area, etc. Shrinking the die size for complex chip designs leads to a complex integration process, which could result in unreliable data transmission over wires. The integration complexity increases the probability of communication errors between PEs due to various noise sources, such as cross-talk, coupling noise, soft errors, etc [73]. Therefore, error-control schemes are used to address the reliability issue. However, using error-control schemes in NoC degrades the overall system performance due to the added hardware (silicon area) and logic operations (extra delay) [74]. Hence, a joint measure of reliability and performance must be considered through a unified measure, *performability*, to optimally choose the most efficient control scheme for a target application [74]. Among many design factors, the performability criterion of a system could be studied at different levels of abstraction. At early design phases, the network architecture must be designed efficiently to achieve a maximum system performability. However, at early design phases, the exact physical layout structure is not yet defined. Hence, designers do not have enough layout information to choose the most efficient network topology for their target application. In this chapter, we address this problem by: 1) studying the impact of the network topology on system performability using graph-theoretic concepts, 2) developing a topology-based performability model for NoC-based systems, 3) proposing a new methodology to improve the performability of a given NoC application at early design phases. The proposed methodology aims at selecting the optimal topology that achieves the maximum system performability for a given application, taking into consideration the possible changes in noise power

and voltage swing. Finally, the efficiency of the proposed methodology is verified through a case study of a video application.

3.1.3 Chapter Organization

This chapter is organized as follows. Section 3.2 highlights related wok. Section 3.3 discusses the architecture of various NoC topologies through graph-theoretic concepts. Section 3.4 presents a performability model for NoC-based systems. Section 3.5 discusses the impact of the network topology on system performability. Section 3.6 presents a mathematical problem formulation and proposes a methodology to improve the performability of a given application using topology-based design. As a proof of concept, the proposed methodology is verified through a case study in Section 3.7. Finally, this chapter is summarized in Section 3.8.

3.2 Related Work

The work related to performability in NoC literature could be classified into two main categories. The former is reliability and performance issues for NoC applications [73], whereas the latter is topological optimization for NoCs [55].

For the first category, research is focused on modeling and improving NoC-based system reliability [75], or evaluating its performance [76]. Reliability is usually achieved via fault-tolerant designs, which enables the network to survive the failure of one or more components without a disruption of its operation [77]. In NoC-based designs, recovering transient errors that occur in the communication subsystems is one of the main key challenges. As technology scales down, supply voltage decreases and interconnects become more sensitive to noise [78]. Therefore, analysis of error control schemes for NoCs received increased attention recently [79]. For instance, Ejlali et al.

analyzed the impact of various error-control schemes on the mutual trade-off between reliability, performance, and energy consumption when voltage swing varies [80].

As for the second category, NoC topology optimization has been addressed from different perspectives such as reducing the communication cost, power consumption, and performance. In [55], Ahonen et al. developed a tool to handle the NoC optimization based on exploiting target application domain specific features. Their optimization methodology applied network partitioning to minimize the cost of the communication over partition boundaries [55]. Power-aware topology optimization was the focus of many researchers [50, 63]. Chang et al. proposed a power-aware topology construction method and compared customized irregular network topologies to regular mesh and torus topologies with respect to packet latency and average energy consumption. A mixed integer linear programming (MILP) formulations for synthesis of custom NoC architectures is proposed in [81]. The objective function in [81] was to minimize the power consumption subject to given performance constraints. Also, a multi-objective algorithms is used in [82] for the optimization of performance and power dissipation. Another approach proposed in [39] synthesized an architecture which is neither regular nor fully customized. Instead, the proposed communication architecture is a superposition of a few long-range links and a standard mesh network [39].

Although some of the above work has addressed the performance and reliability of NoCs, only one group addressed the performability metric in [80]. Even though, this group did not discuss the impact of the network topology design on system performability. In this chapter, we fill the gap between the two mentioned research directions by analyzing the impact of the network topology architecture on system performability. We tackle this problem from a totally new, topology-based, perspective aiming at improving the NoC-based system performability at early design phases. We optimize the topological architecture of interconnection network

subject to a joint consideration of performance and reliability, or in another word, *performability*, for a given application.

3.3 A Graph-Theoretic Representation of NoC Topologies

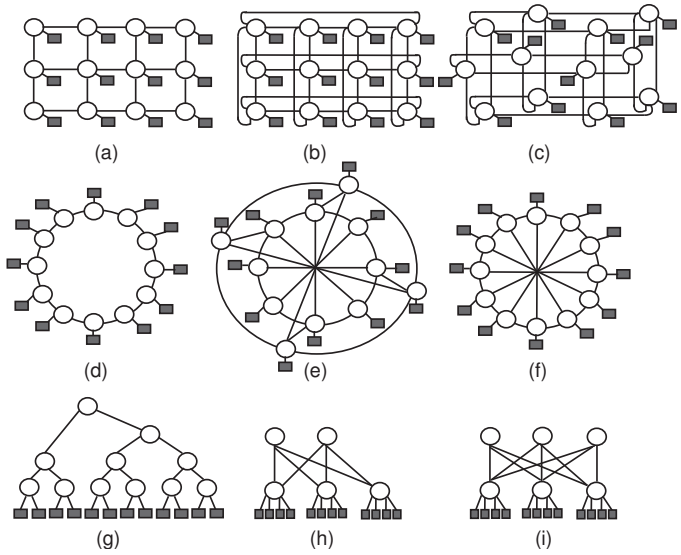


Figure 3.1: Nine regular NoC topologies connecting 12 PEs: (a) Mesh. (b) Torus. (c) Folded Torus (Folded). (d) Ring. (e) Octagon (Oct). (f) Spidergon (Spider). (g) Binary tree (BT). (h) Butterfly Fat Tree (BFT). (i) SPIN. (Routers are represented by white circles, whereas Processing Elements (PEs) are represented by dark squares.)

In this chapter, we use the same graph-theoretic concepts that have been mentioned in Subsection 2.3.1. Because this chapter focuses only on 2-D topologies, Hypercube has been excluded. Also, from a reliability point of view, having a centralized communication architecture with one single routing module is not recommended because of the high possibility of network failure if this module failed to function for any reason. Therefore, star topology has been excluded. In this chapter, we study three types of regular network topologies: Mesh-based, Ring-based, and

Tree-based architectures, which are represented by nine different topologies, as shown in Figure 3.1 [41–49].

Studying Networks' Architectural Differences using Graph Theory

In this subsection, we employ graph-theoretic concepts to study the differences between different network architectures. One of the graph-theoretic definitions is the edge cut-set. An edge cut-set is a set of edges whose removal from the graph will disconnect the vertices and partition the network into two or more subnetworks. Table 3.1 shows the minimum edge cut-set for the topologies shown in Figure 3.1, when 12 PEs are connected. As shown in Table 3.1, one broken edge is enough for a topology like Binary tree to disconnect 8 PEs, which is 66.7% of the network, whereas 4 edges must be broken in a topology like Torus to disconnect only one PE, which is 8.3% of the network in this example.

Table 3.1: Minimum edge cut-set and other statistical analysis for NoC topologies shown in Figure 3.1, when 12 PEs are connected.

Network topology	No. of edges (m)	Minimum edge cut-set (s)	Percentage s/m (γ)	Total no. of PEs (n)	Max no. of PEs to fail (n_f)	Percentage n_f/n (ϑ)
Mesh	17	2	11.8%	12	1	8.3%
Torus	24	4	16.7%	12	1	8.3%
Folded torus	24	4	16.7%	12	1	8.3%
Ring	12	2	16.7%	12	6	50%
Octagon	22	3	13.6%	12	1	8.3%
Spidergon	24	3	12.5%	12	1	8.3%
Binary Tree	10	1	10%	12	8	66.7%
BFT	6	2	33.3%	12	4	33.3%
SPIN	9	3	33.3%	12	4	33.3%

These statistics reflect several important features, which are unique for each topology. These features could be used as indicators to evaluate the performability of a certain topology relative to others. For instance, maximizing the ratio γ

is an essential requirement because it reflects the network ability to preserve its functionality at edge failure. Consequently, minimizing the ratio ϑ is another important objective because it indicates the level of partial system failure at worst case scenarios. Therefore, we used graph-theoretic concepts to present a complete model of the total network performability in the following section.

3.4 Modeling NoC Performability

The performability of a network topology $\mathcal{H}$ could be modeled through considering two main criteria:

1. **Network functionality:** The ability of a network to preserve its functionality at edge-failure.

2. **Packets reception:** The probability of all packets to be received correctly under the presence of noise.

For the first criterion, we developed an analytical model that represents the network capability of delivering packets from any PE to any PE considering the probability of edge failure. This model reflects the architectural differences of various network topologies through considering the probability of one or more edge failure in each network topology.

For the second criterion, we developed an analytical model that represents the probability of successful packets' reception under the presence of noise. This model reflects the architectural differences of various network topologies through considering the performability of all packets being received successfully for a given application. In the following subsections, we explain how we developed these models. Then, we present a complete performability model for the whole network.

3.4.1 Network Functionality

Figure 3.2 shows an example of 12 PEs connected through a 3 × 4 mesh topology, where λ_{AB} packets are being transmitted from A to B. In this example, the minimum edge cut-set (s) equals 2 and if an edge-failure occurred to these 2 links at the corners, the network will be broken by disconnecting one PE. Assuming the probability of an

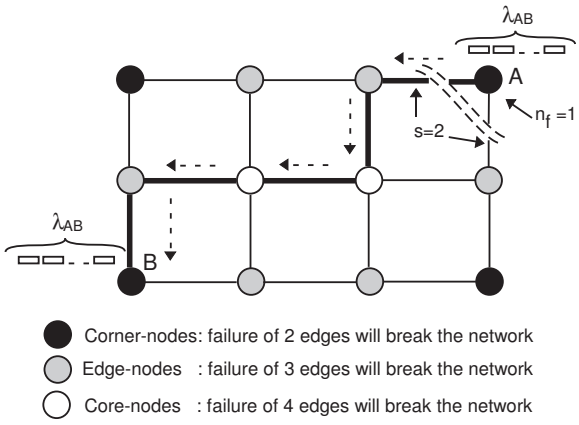


Figure 3.2: An example of a 3 × 4 mesh topology. (Each router-node pairs are represented by a circle.) λ_{AB} packets are being transmitted from A to B. If an edge-failure occurred to 2 links at one of the corner-nodes, one node will be disconnected from the network; $n_f = 1$.

edge failure is P_e, a new metric to represent the network probability to preserve its functionality due to edge failure ($\mathcal{P}_{\mathcal{H}n}$), when an application is mapped to a network topology $\mathcal{H}$, could be modeled for the network topology shown in Figure 3.2 as follows

$$\mathcal{P}_{nmesh} = (1 - P_e^2)^4 \cdot (1 - P_e^3)^6 \cdot (1 - P_e^4)^2 \tag{3.1}$$

Equation (3.1) consists of three components. The first component represents the probability of success, which equals 1-probability of failure, for the edges around the corner-nodes. As shown in the figure, P_e is raised to the power 2 in the first

component because the corner- nodes will be disconnected from the network if 2 edges failed. Because we have 4 corner-nodes, the first component of the equation $(1 - P_e^2)$ is raised to the power 4. Similarly, the probability of success for the edges around the edge-nodes and core-nodes were presented in the second and the third components, respectively. Using the same approach, $\mathcal{P}_{\mathcal{H}n}$ for all NoC topologies shown in Figure 3.1, when 12 PEs are connected, could be written as shown in Table 3.2. To elaborate more on how these models could help us understand the impact of

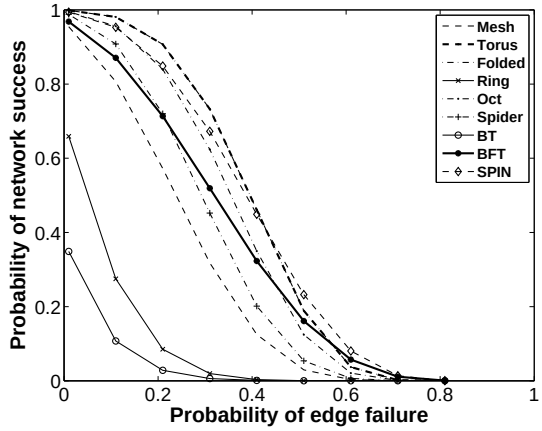
Table 3.2: $\mathcal{P}_{\mathcal{H}n}$ for NoC topologies shown in Figure 3.1, when 12 PEs are connected.

Network topology $\mathcal{H}$	$\mathcal{P}_{\mathcal{H}n}$
Mesh	$(1 - P_e^2)^4 \cdot (1 - P_e^3)^6 \cdot (1 - P_e^4)^2$
Torus	$(1 - P_e^4)^{12}$
Folded torus	$(1 - P_e^4)^{12}$
Ring	$(1 - P_e)^{12} + 12 \cdot P_e \cdot (1 - P_e)^{11}$
Octagon	$(1 - P_e^3)^4 \cdot (1 - P_e^4)^8$
Spidergon	$(1 - P_e^3)^{12}$
Binary Tree	$(1 - P_e)^{10}$
BFT	$(1 - P_e^2)^3 \cdot (1 - P_e^3)^2$
SPIN	$(1 - P_e^3)^6$

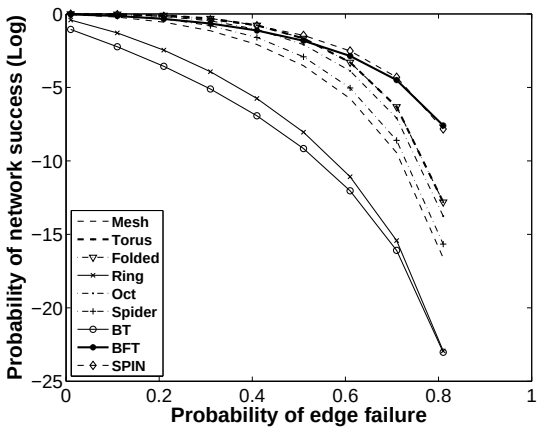
the architectural differences on the probability of network failure, we carried out an experiment using MATLAB® to evaluate the values of $\mathcal{P}_{\mathcal{H}n}$ for all topologies shown in Figure 3.1. Figure 3.3 shows the probability of network success versus probability of edge failure for the nine regular topologies in Figure 3.1 in normal and log scale for different scales of P_e.

Case 1: High probability of edge failure [0-1]

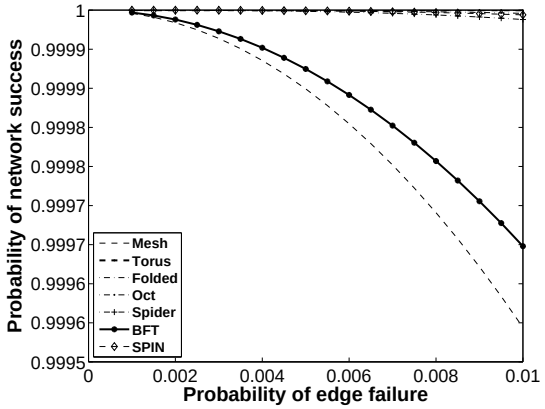
Figure 3.3(a) and Figure 3.3(b) show clearly that at high probabilities of edge failure, mesh-based (mesh, tours, folded) topologies outperform other types such as ring-based (ring, octagon, spidergon) or tree-based (BT, BFT, SPIN) at the lowest values of P_e. As the probability of edge failure increases, tours and folded tours outperform



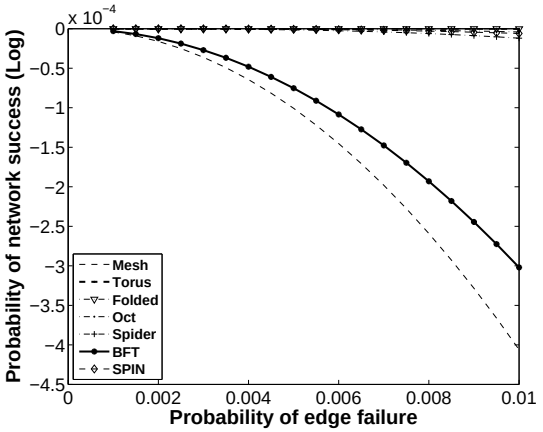
(a) Probability of network success versus probability of edge failure (high probability) for the nine regular topologies in Figure 3.1.



(b) Probability of network success (in log scale) versus probability of edge failure (high probability) for the nine regular topologies in Figure 3.1.



(c) Probability of network success versus probability of edge failure (low probability) for the nine regular topologies in Figure 3.1.



(d) Probability of network success (in log scale) versus probability of edge failure (low probability) for the nine regular topologies in Figure 3.1.

Figure 3.3: Experimental results that show the impact of the network topology on system probability of success with the change in probability of edge failure.

the mesh. At the same time, Octagon and Spidergon values start to be relatively close to the mesh-based topologies. Also BFT slowly becomes relatively better than other topologies. When the probability of edge failure reaches very high values, BT and ring could not preserve their functionalities and clearly became the worst choice, whereas mesh is improving again and BFT reaches the top, slightly higher than mesh topology.

Case 2: Low probability of edge failure [0.01-0.1]

In Figure 3.3(c) and Figure 3.3(d), we have not consider the ring and BT topologies because they proved to have the lowest probability of network success and result in unclear plotting due to the big difference in values. At low probabilities of edge failure, there is only very small difference between all topologies, expect Ring, BT, SPIN and mesh. The results in Case 1 and Case 2 could be justified through the statistical analysis shown in Table 3.1 and by observing the architectures of all topologies shown in Figure 3.1. BT is the only topology that only one edge failure could disconnect 50% or more of its nodes, which indicates how unreliable structure it is. Consequently, ring topology is very vulnerable to edge failure since two simultaneous edge failures will break the network. On the other hand, topologies such as Torus requires at least 16% or more of its communication resources to fail before it is disconnected. Other topologies such as BFT and SPIN have a considerable immunity to the probability of edge failure because of their unique balance feature in terms of edge cut-set and number of PEs to fail as shown in Table 3.1.

3.4.2 Packets Reception

In this subsection, we developed an analytical model that represents the probability of successful packets' reception under the presence of noise. First, we present the

performability definition of a single on-chip interconnect (link), that was proposed in the literature. Then, we show how to extend this performability definition, from a topology-based perspective, to represent the whole network.

Performability of a Single On-chip Interconnect

The performability of a single on-chip interconnect (link) is defined as the probability to transmit a certain number of useful bits (L), which are put into λ packets through a single link during a specific time interval (T) in the presence of noise [80]. Ejlali et al. in [80] developed performability models of a single on-chip interconnect for four communication schemes: simple non-fault-tolerant (SNFT) communication and three error-control schemes; Automatic Repeat Request (ARQ), Forward Error Control (FEC), and Hybrid ARQ/FEC (HARQ). In these models, the sum of several uncorrelated noise sources affecting a single on-chip interconnect is modeled as a single gaussian noise source [80, 83]. Although other noise models are proposed in the literature, the gaussian noise model is chosen by many researchers based on the assumption that all the noise sources collectively induce a noise voltage on the channel which follows a Gaussian distribution with zero mean and variance σ^2 [78, 80, 84]. Using Gaussian noise model, the Bit Error Rate (BER) for a link is given by [78, 80]

$$BER = \frac{1}{\sqrt{2\pi}} \int_{\frac{V_{sw}}{2\sigma}}^{\infty} e^{-\frac{\mu^2}{2}} d\mu \tag{3.2}$$

where V_{sw} is the swing voltage used to reduce the energy consumption and σ is the noise standard deviation. Assuming SNFT, where the whole packet is considered corrupted if only one bit is not received correctly, the packet error rate (PER), defined as the probability of having a corrupted packet is given by [78, 80]

$$PER = 1 - (1 - BER)^L \tag{3.3}$$

where L is the number of bits per packet, assume fixed packet size. From (3.3), the probability that a transmission of λ_{ij} packets will be successful over a single link is the probability that each packet of the λ_{ij} packets will be transmitted successfully, which is given by [78, 80]

$$\begin{aligned} P_{ij} &= (1 - PER)^{\lambda_{ij}} \\ &= \left(1 - \frac{1}{\sqrt{2\pi}} \int_{\frac{V_{sw}}{2\sigma}}^{\infty} e^{-\frac{\mu^2}{2}} d\mu\right)^{L\lambda_{ij}} \end{aligned} \tag{3.4}$$

We use the probability definition in (3.4) to represent the performability of a global (router-to-router) on-chip link for any NoC topology.

Extending the Performability Definition

Based on the above definition for a single link performability, and assuming the shortest path and static routing, the network performability of a given application depends on the target topology mapping. Therefore, a unique connectivity matrix (C) is generated for each one of the nine regular network topologies mentioned in Section 3.3 [57]. This matrix represents the minimum number of links a packet goes through during its transition from the source node to the destination node. Hence, it could be used to represent the architecture and the mapping of PEs for any network topology. For example, the connectivity matrix C of the mesh network topology shown in Figure 3.4 is written as follows

$$C = \begin{bmatrix} 0 & 1 & 2 & 1 & 2 & 3 & 2 & 3 & 4 \\ 1 & 0 & 1 & 2 & 1 & 2 & 3 & 2 & 3 \\ 2 & 1 & 0 & 3 & 2 & 1 & 4 & 3 & 2 \\ 1 & 2 & 3 & 0 & 1 & 2 & 1 & 2 & 3 \\ 2 & 1 & 2 & 1 & 0 & 1 & 2 & 1 & 2 \\ 3 & 2 & 1 & 2 & 1 & 0 & 3 & 2 & 1 \\ 2 & 3 & 4 & 1 & 2 & 3 & 0 & 1 & 2 \\ 3 & 2 & 3 & 2 & 1 & 2 & 1 & 0 & 1 \\ 4 & 3 & 2 & 3 & 2 & 1 & 2 & 1 & 0 \end{bmatrix}$$

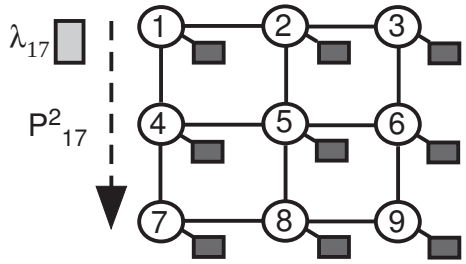


Figure 3.4: An example of a 3×3 mesh topology. (Routers are represented by white circles and PEs are represented by dark squares.) The gray rectangular represents a number of packets transmitted from node one to node seven (λ_{17}). P is the performability of a single on-chip interconnect and P_{17}^2 represents the performability when λ_{17} packets are transmitted from node 1 to 7.

For a given TDG, λ_{ij} represents the number of packets being transmitted from node i to node j. The corresponding performability measure over a single link (P) is denoted by P_{ij}. Now, if λ_{ij} packets need to go through c_{ij} hops in a certain topology to reach their destination, and assuming independent packet transmission, the total performability measure is denoted by ($P_{ij}^{c_{ij}}$).

For example, if a number of packets (λ_{17}) is to be transmitted from node one to seven, as shown in Figure 3.4, these packets have to go through two hops to reach their destination and, hence, the performability in this example could be represented by P_{17}^{2}. $P_{ij}^{c_{ij}}$ in this case is a function of the number of transmitted packets (λ_{ij}). From the above discussion, we can use Gaussian noise model, which represents the interconnect Bit Error Rate (BER) [78], to develop an analytical model of the network performability ($\mathcal{P}_{\mathcal{H}r}$) considering the probability of successful packets' reception. The network performability for a given application, normalized to the total number of packets transmitted, when it is mapped to a specific network topology ($\mathcal{H}$) is given by

$$\mathcal{P}_{\mathcal{H}r} = \prod_{i=1}^{n} \prod_{j=1}^{n} P_{ij}^{c_{ij}} \tag{3.5}$$

where P_{ij} is given by [80]

$$P_{ij} = \left(1 - \frac{1}{\sqrt{2\pi}} \int_{\frac{V_{sw}}{2\sigma}}^{\infty} e^{-\frac{\mu^2}{2}} d\mu\right)^{\frac{L\lambda_{ij}}{\sum_{i=1}^{n}\sum_{j=1}^{n}\lambda_{ij}}} \tag{3.6}$$

The normalization to the total number of packets transmitted is important to have a performability representation between 0 and 1.

Because the exact physical layout structure is not usually defined at early design phases, the connectivity matrix (C) is used in (3.5) based on the assumption that all links have equal BER. However, a modified connectivity matrix (C_M) could be used after placement and routing to reflect the exact physical layout information. The modified connectivity matrix (C_M) could be generated using the Hadamard product of C and $\mathcal{D}$ matrices as follows

$$C_M = C \times \mathcal{D} \tag{3.7}$$

where $\mathcal{D}$=$[d_{ij}]$ is a matrix that represents the relative physical lengths of the links between routers after placement and routing. The entries of this matrix could be extracted from the ASIC design tool after placement and routing. Then, C could be replaced by C_M in (3.5), in a back-annotation step, to acquire the exact network performability $\mathcal{P}_{\mathcal{H}r}$. Because we target early design phases in this chapter, this step is left for future work.

3.4.3 Total Network Performability

From (3.5) and Table 3.2, and based on the above discussion, the total network performability which considers the two criteria mentioned before (at the beginning of Section 3.4) could be defined as

$$\mathcal{R}_{\mathcal{H}} = \mathcal{P}_{\mathcal{H}n} \cdot \mathcal{P}_{\mathcal{H}r} \tag{3.8}$$

3.5 Topology-based Analysis of System Performability

In this section, we study and analyze the impact of the network topology architecture on system performability using the models proposed in Section 3.4. Taking into consideration the possible changes in noise variance and voltage swing, several experiments are performed to evaluate the performability of the network and explore its characteristics. A framework is developed using MATLAB® to generate connectivity matrices for the nine topologies mentioned in Section 3.3 and calculate their performability using (3.8). All experiments discussed in this section use SNFT as an example of a network communication scheme. However, other error control schemes could be easily applied using the equations developed in [80].

To accurately evaluate the performability of different network topology architectures, the following experimental setup were used.

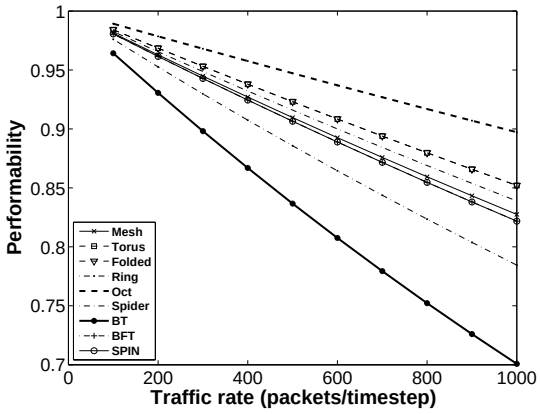
1. Nine connectivity matrices are generated to represent the connectivity of 12 PEs when mapped to the nine network topologies discussed in Section 3.3. The number of PEs is chosen to be in line with our case study, which is presented in Section 3.7.

2. A uniform traffic distribution is used to allow omitting the mapping factor (the location of each PE within the network topology). This means, each PE sends the same number of packets to the other 11 PEs. We designed the network to perform that way in order to study only the impact of the architecture without considering the mapping effect.

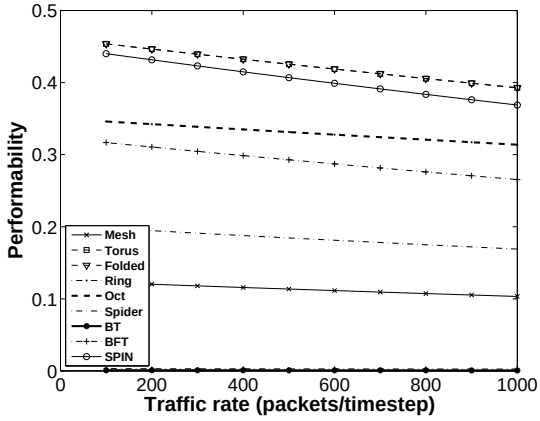
3. Router architecture is assumed to be based on an output-queue router, as an example, but other architectures could also be used.

4. Wormhole routing is used as a routing protocol.

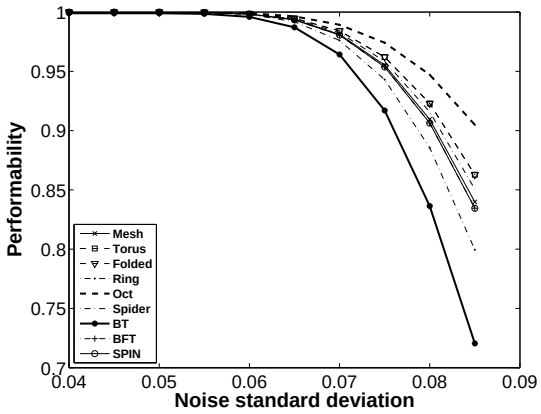
Experimental results in Figure 3.5 show the impact of the network topology on system performability with respect to traffic rate, noise standard deviation (σ) and probability of edge failure. The first set of experiments is performed to explore the effect of the traffic rate changes on the network performability. Figure 3.5(a) shows network performability versus traffic rate in packets per timestep for 12 PEs mapped on the nine regular topologies in Figure 3.1. The voltage swing is set to 0.5, white gaussian noise standard deviation (σ) is set to 0.07, and the probability of edge failure= 10^{-4}. With the increase in the traffic rate, Octagon achieves the maximum performability compared to other topologies, whereas BT gives the minimum performability. The main observation that could be observed from Figure 3.5(a) is the significant effect of the topology architecture on the performability especially with the increase in the traffic rate. For instance, performability is degraded by 21.90% if BT is chosen instead of Octagon to connect 12 PEs transferring



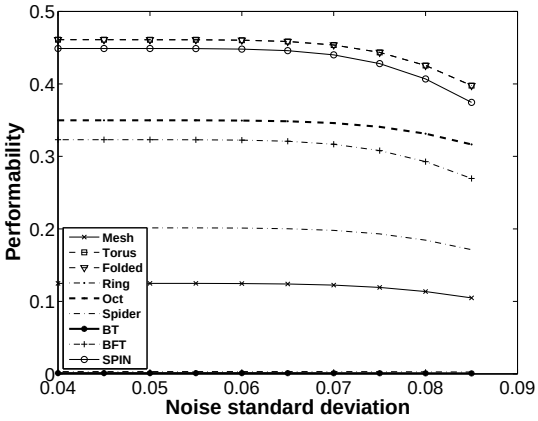
(a) Network performability versus traffic rate for 12 PEs mapped on the nine regular topologies in Figure 3.1, when SNFT scheme is used. $V_{sw} = 0.5$, $\sigma = 0.07$, and probability of edge failure= 10^{-4}.



(b) Network performability versus traffic rate for 12 PEs mapped on the nine regular topologies in Figure 3.1,when SNFT scheme is used. $V_{sw} = 0.5$, $\sigma = 0.07$, and probability of edge failure= 0.5.

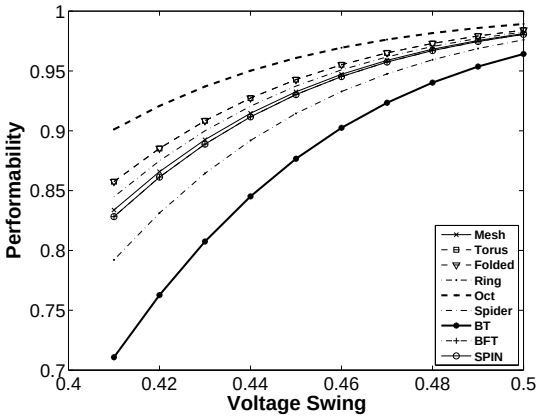


(c) Network performability versus noise standard deviation (σ) for the nine regular topologies in Figure 3.1, when SNFT scheme is used. $V_{sw} = 0.5$ and probability of edge failure= 10^{-4}.

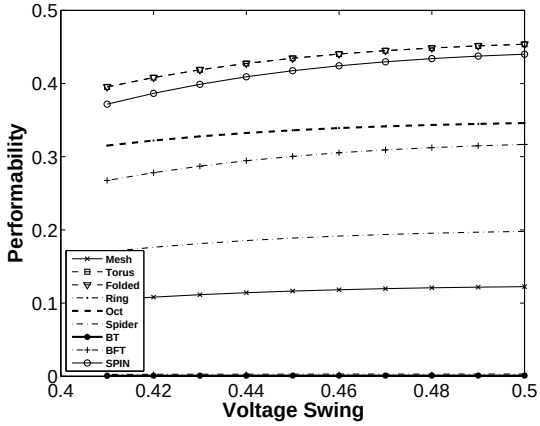


(d) Network performability versus noise standard deviation (σ) for the nine regular topologies in Figure 3.1, when SNFT scheme is used. $V_{sw} = 0.5$ and probability of edge failure= 0.5.

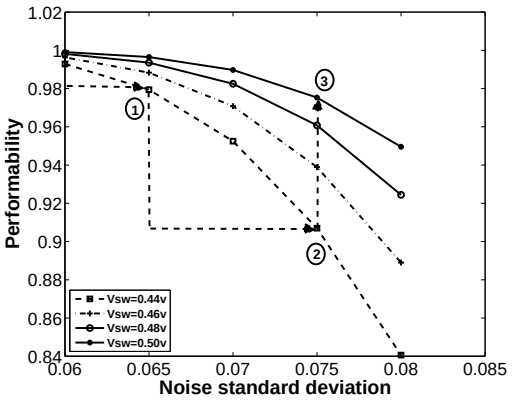
Figure 3.5: The impact of the network topology on system performability with respect to traffic rate, noise standard deviation (σ) and the probability of edge failure.



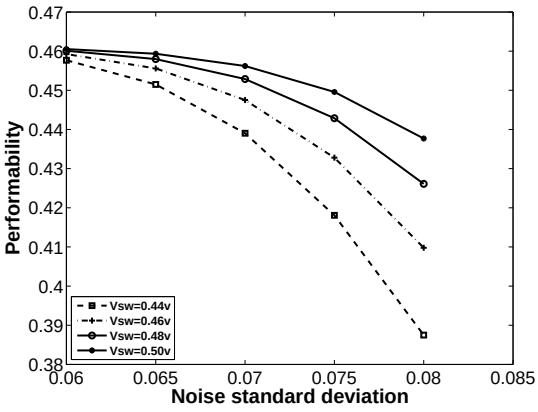
(a) Network performability versus voltage swing (V_{sw}) for the nine regular topologies in Figure 3.1 when SNFT scheme is used. $\sigma = 0.07$ and probability of edge failure= 10^{-4}.



(b) Network performability versus voltage swing (V_{sw}) for the nine regular topologies in Figure 3.1 when SNFT scheme is used. $\sigma = 0.07$ and probability of edge failure= 0.5.



(c) Network performability versus standard deviation for different voltage swings when SNFT scheme is used (for Torus topology) and probability of edge failure= 10^{-4}.



(d) Network performability versus standard deviation for different voltage swings when SNFT scheme is used (for Torus topology) and probability of edge failure= 0.5.

Figure 3.6: The impact of the network topology on system performability with respect to noise standard deviation (σ), voltage swing (V_{sw}) and the probability of edge failure.

1000 packets/timestep per node on a uniform traffic distribution pattern. These results completely change with the increase in the probability of edge failure. As shown in Figure 3.5(b), mesh and torus outperform other topologies because of their architectural features, which allows many alternative routes for packets transmission.

Knowing this result is very important for designers at early design phases because it allows them to choose the appropriate architecture that achieves the maximum performability at the system level design phase. Nevertheless, these results are just examples and cannot be generalized for all applications. In fact, the TDG and PE mapping, which are represented by the λ matrix, are vital factors and will definitely impact system performability.

The second set of experiments is carried out to measure the influence of noise power on system performability when different network architectures are used. Figure 3.5(c) shows network performability versus noise standard deviation (σ) for the nine regular topologies discussed in Section 3.3. As shown in the figure, Octagon achieves the maximum performability, whereas BT has the minimum value, which is in line with the results of the previous experiment. However, the key observation in Figure 3.5(c) is that some network architectures are less immune to noise than others. In other words, performability degradation ratio, due to the increase in noise power, is not the same for all network topology architectures. For instance, when the value of σ increases from 0.07 to 0.08, the performability degrades by 4.2% for Octagon topology, whereas it degrades by 13.2% for a BT topology. Consequently, the difference between Octagon and BT with respect to performability increases with the increase in noise power. For instance, at $\sigma = 0.07$, BT topology achieves a performability of 2.5% less than Octagon topology. This percentage increases to 11.6% at $\sigma = 0.08$. Figure 3.5(d) showes how the different topologies react to the increase of the probability of edge failure. Tours and folded topologies again proved to be more immune to noise than other topologies.

The third set of experiments is conducted to measure the impact of changing the voltage swing on the network performability for different network topologies. Voltage swing reduction is often used to address the energy consumption issues in NoC applications [85]. However, reducing the voltage swing increases the BER due to the increase in the Signal-to-Noise Ratio (SNR). Figure 3.6(a) shows network performability versus voltage swing for the same nine regular topologies. As shown in the figure, Octagon achieves the maximum performability, whereas BT has the minimum value, which is in line with the results of the previous experiments. With the increase in the voltage swing, performability improvement ratio changes according to the used network topology. For instance, increasing the voltage swing from 0.41 to 0.50 leads to an improvement in the performability by 9.7% for Octagon topology and by 35% for BT topology. Also, the difference between Octagon and BT with respect to performability decrease with the increase of voltage swing. For instance, at $V_{sw} = 0.41$, BT topology achieves a performability of 21.2% less than Octagon topology. This percentage decreases to 2.4% at $V_{sw} = 0.50$. Once the probability of edge failure increases to 0.5 as shown in Figure 3.6(b), tours and folded became the best topologies followed by SPIN. Therefore, a joint consideration of voltage swing and noise power is needed to study the impact of both parameters simultaneously on the network performability.

The fourth set of experiments is performed to address the joint consideration of voltage swing and noise power on the network performability. Figure 3.6(c) shows network performability versus noise standard deviation (σ) for different swing voltages. Torus topology is taken as an example in this experiment to show how we can improve the performability of a certain topology by choosing the optimum value of V_{sw} for a system noise power level (σ). The main observation which could be made from Figure 3.6(c) is that increasing the voltage swing could compromise the increase in the noise power to a certain limit. Increasing the voltage swing leads to a

corresponding increase in the noise margin, and hence, improve the SNR. To explain this, assuming a scenario in which the noise standard deviation increases by 15.3% (by moving from point one to point two in Figure 3.6(c)). This increase leads to a corresponding degradation of 7.4% in the network performability. However, increasing the voltage swing by 13.6% (by moving from point two to point three in Figure 3.6(c)) compensates the noise increase effect and returns the network performability to its original value. Therefore, for applications that require a certain performability level, such as safety-critical applications, there is an optimum value of V_{sw} that could compensate the expected noise power level (σ) to achieve the design requirements. The same concepts applies when the probability of edge failure increases as shown in Figure 3.6(d).

To conclude from the above findings, there are six important parameters which significantly affect the performability of a NoC-based system: network topology architecture, traffic distribution, PEs' mapping, noise power, voltage swing, and probability of edge failure. A joint consideration of two or three of these six parameters could lead to a considerable improvement in the performability. However, all of them must be considered simultaneously to acquire an optimum performability, at early system level design phases, for a given NoC-based application. Also, the main reason behind the results found in the previous four-set of experiments is that BT have an architecture that forces packets to go through longer paths to reach their destination as the number of levels in the tree increases. Also, it is very sensitive to the increase of edge failure probability. On the other hand, mesh-based topologies, especially Torus, have an architecture that allows shorter packet transmission paths. In other words, the average inter-node distance (Δ) for a certain topology ($\mathcal{H}$) has a significant effect on its total performability, where Δ is given by

$$\Delta \;=\; \frac{\sum_{i=1}^{n}\sum_{j=1}^{n}\lambda_{ij} \;\times\; c_{ij}}{\sum_{i=1}^{n}\sum_{j=1}^{n}\lambda_{ij}},\; i \;\neq\; j \tag{3.9}$$

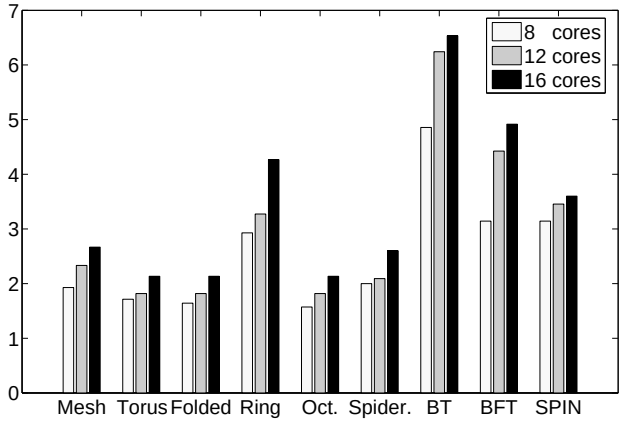


Figure 3.7: The average inter-node distance (Δ) of 8-, 12-, and 16-core applications for different network topologies.

An experiment is performed to show the significance of the parameter Δ as an early system design indicator when various number of cores are considered. Since the number of cores in almost all the case studies found in the literature ranges from 8 to 16, we examined three cases: 8-, 12-, and 16-core applications. We use the same experimental setup mentioned at the beginning of this section. Figure 3.7 shows Δ of 8-, 12-, and 16-core applications for different network topologies. As the average inter-node distance (Δ) goes low for a given topology ($\mathcal{H}$), its corresponding performability gets high. Therefore, for this case study, Octagon topology is the optimum choice, whereas BT remains the worst network architecture even with the change in the number of cores.

However, we must emphasis on that Δ is not the only factor that affects the NoC-based system performability. These results will change if a different traffic pattern (different application) is considered. In this experiment we assumed that

each PE sends the same number of packets to all other PEs. We designed the network to perform in that way in order to study only the impact of the architecture without considering the mapping effect. For a different traffic distribution, BT might become the optimum choice and Octagon might be the worse. In other words, the traffic distribution matrix (λ), which is a unique matrix for each application, is a major design parameter that significantly affects the choice of the target network topology. Also, the noise level (σ) and voltage swing (V_{sw}) must be considered based on the design constraint and requirements. We considered all these parameters in our proposed methodology in Section 3.6.2.

3.6 Problem Formulation and Proposed Methodology

To acquire a maximum network performability, certain design parameters must optimally be chosen. In this section, we present a mathematical formulation of this optimization problem, then introduce our proposed methodology to solve the optimization problem and, hence, find the network topology architecture that achieves a maximum performability.

3.6.1 Problem Formulation

For a given application represented by a TDG and a traffic distribution matrix (λ)

$$\text{Maximize } \mathcal{R}_{total} \;=\; f(\mathcal{H}, C, V_{sw}, \sigma, P_e) \tag{3.10}$$

Subject to:

$$
\begin{aligned}
c_{ij} &= c_{ji} \\
V_{swa} &\leq V_{sw} \leq V_{swb} \\
\sigma_a &\leq \sigma \leq \sigma_b \\
P_e &\geq P_{emax}
\end{aligned}
$$

where $C = [c_{ij}]$ represents the connectivity matrix as discussed in Subsection 3.4.2. V_{swa} and V_{swb} are the minimum and maximum permissible swing voltage according to design requirements of a given application. σ_a and σ_b are the margins of the expected noise standard deviation. P_{emax} is the maximum expected edge failure probability.

3.6.2 Proposed Methodology

The proposed methodology targets the nine regular topologies, discussed in Section 3.3, aiming at maximizing network performability. Other regular or irregular topologies could be easily considered by adding their C matrices to increase the granularity of the topologies' library. Figure 3.8 shows a flowchart of the proposed methodology to improve the performability based on topology optimization. This methodology could be explained as follows.

The first step of the proposed methodology considers the TDG as the main design input. Then, from the TDG, a $\lambda = [\lambda_{ij}]$ matrix is generated, with an initial mapping, as a mathematical representation of the graph, where λ_{ij} is the number of packets per time step associated with each logical link. The third step is to generate C matrices for all nine topologies based on the number of PEs given in the TDG. Following that, we maximize the performability in two consecutive steps (the fourth and fifth steps). The fourth step finds the optimum topology/mapping pair that achieves minimum

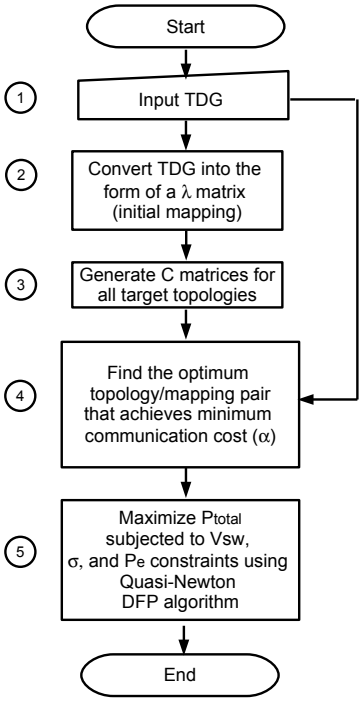


Figure 3.8: Proposed methodology to improve NoC performability using a topology-based approach.

communication cost (α) assuming shortest path routing, where α is given by

$$\alpha \;=\; \min\left(\sum_{i=1}^{n}\sum_{j=1}^{n}\lambda_{ij} \;\times\; c_{ij}\right),\; i \;\neq\; j \tag{3.11}$$

From (3.6) and (3.8), reducing the minimum communication cost (α) increases the corresponding performability. This is done by re-mapping all PEs such that those who have the highest traffic rates are associated with the shortest distances. A search algorithm is developed to re-allocate PEs to different positions within each one of the nine topologies. The re-mapping process is done by analyzing all numbers in the λ matrix. Then, all possible changes in rows and columns, on a one-by-one base,

are done to re-arrange the matrix, and hence, re-map the PEs such that the highest traffic rates (numbers) are associated with shortest distances (neighboring nodes). Finally, we use the Quasi-Newton David-Fletcher-Powell (DFP) algorithm [86] to find the optimum V_{sw} that could compensate the expected noise level σ and, at the same time, achieves a maximum performability ($\mathcal{R}_{total}$) at the maximum expected probability of edge failure P_e. Quasi-Newton DFP is used because it is very efficient for more than one parameter varying at the same time, which is the case in our optimization problem. The DFP algorithm is modified to generate a set of optimum points, for V_{sw}, that achieve maximum performability values with a variation of 10^{-5}. P_e is set to be greater than or equal the maximum expected probability of edge failure. Finally, the V_{sw} value that achieves a maximum performability, under given noise level constraints, is selected.

One of the advantages of the proposed methodology is its scalability. The proposed methodology could be used for any application, regardless the number of PEs, as long as its TDG is provided. The generation of λ and C matrices, topology/mapping search, and the mathematical operations of the DFP algorithms are executed using MATLAB®.

3.7 Performance Evaluation by Experimentation

The proposed methodology is validated through an experimental case study. The Video Object Plane Decoder (VOPD) core discussed in [61] is taken as an example to evaluate the performance of the proposed methodology.

Figure 3.9(a) shows a typical TDG for the video application (VOPD) [61]. The numbers written on the arrows are the number of packets/time step transmitted and the numbers written on the circles represent PEs' numbers. From this TDG, the

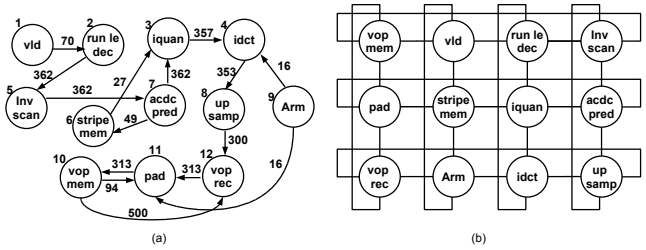


Figure 3.9: (a) TDG of Video Object Plane Decoder (VOPD). (b) VOPD Mapping to a Torus topology (an output from the proposed methodology).

traffic distribution matrix (λ), which represents the initial mapping of the VOPD, shown in Figure 3.9(a), is given by

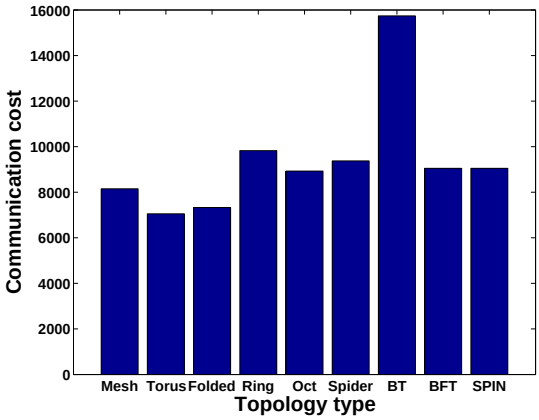
$$\lambda = \begin{bmatrix} 0 & 70 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 362 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 357 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 353 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 362 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 27 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 362 & 0 & 0 & 49 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 300 \\ 0 & 0 & 0 & 16 & 0 & 0 & 0 & 0 & 0 & 0 & 16 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 94 & 500 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 313 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 313 & 0 \end{bmatrix}$$

Following that, connectivity matrices are generated for all nine topologies discussed in Subsection 3.4.2. A search algorithm developed in MATLAB® is used to find the optimum topology/mapping pair that achieves minimum communication cost (α). Figure 3.9(b) shows the mapping of VOPD to a Torus topology after applying our search algorithm. The traffic distribution matrix (λ), after mapping, is given by

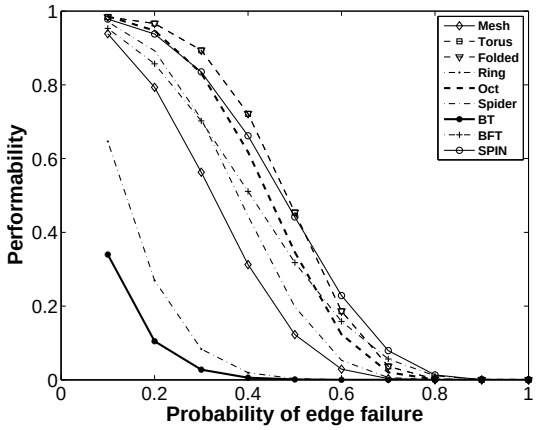
$$\lambda = \begin{bmatrix} 0 & 94 & 500 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 313 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 313 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 70 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 27 & 0 & 0 & 0 & 0 \\ 0 & 16 & 0 & 0 & 0 & 0 & 0 & 0 & 16 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 362 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 357 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 353 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 362 & 0 \\ 0 & 0 & 0 & 0 & 49 & 0 & 0 & 362 & 0 & 0 & 0 & 0 \\ 0 & 0 & 300 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

The minimum communication cost for all nine topologies are calculated using (3.11) and compared to show the significance impact of the topology/mapping selection on the network communication cost. As shown in (3.11), the communication cost is calculated by counting how many hops are needed, for each packet, to reach its destination. Figure 3.10(a) shows the minimum communication cost versus topology type after mapping the VOPD core. Results show that for VOPD, a Torus topology achieves the minimum α, whereas BT has the maximum α. The reason behind this result is that the BT architecture forces packets to go through longer paths as the number of levels in the tree increases. The communication cost (α) could have been increased by 122.6% if a BT topology is chosen instead of the generated Torus. This example highlights the importance of choosing the most appropriate topology/mapping pair for a given application.

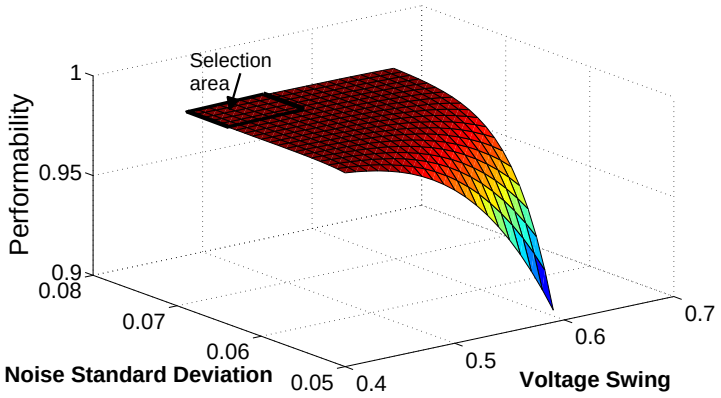
Finally, Quasi-Newton DFP algorithm explained in [86] is used to find the set of optimum V_{sw} values that compensates the noise level given in the design constraints. Then, the V_{sw} value that achieves a maximum performability ($\mathcal{R}_{total}$), under given noise level and probability of edge failure constraints, is selected. For this case study, we used the following design constraints



(a) Minimum communication cost (α) versus topology type for the VOPD application.



(b) Performability of various probability of edge failure for the VOPD application, when SNFT scheme is used. $V_{sw} = 0.5$, $\sigma = 0.07$.



(c) Network performability versus standard noise deviation (σ) and voltage swing (V_{sw}) for an VOPD application mapping to a Torus topology.

Figure 3.10: Experimental results.

$$\begin{aligned} 0.40 &\leq V_{sw} \leq 0.65 \\ 0.05 &\leq \sigma \leq 0.08 \\ P_e &\geq 0.1 \end{aligned}$$

and the initial point (x_0), which is required to start the search in the Quasi-Newton DFP algorithm, is set, randomly, to

$$x_0 = [V_{sw0}\ \sigma_0\ P_e] = [0.40\ 0.05\ 0.3]$$

The resulting output has a performability of 0.998 at $V_{sw} = 0.47$, $\sigma = 0.066$, and $P_e = 0.105$. To have a better understanding of the behavior of different network topologies with the change of the probability of edge failure for this specific application, we carried out an experiment to evaluate the system performability at different probabilities of edge failure. Figure 3.10(b) shows that at torus topology has outperformed other topologies till the probability of edge failure reaches 50%. Following that SPIN topology has become the optimum choice under the given experimental setups. Because the constrains in our case study requires that $P_e \geq 0.1$, torus topology was the recommended candidate for the target application.

To elaborate more on the impact of noise level and voltage swing changes on the selection of a network topology for this specific application, a 3-D curve is plotted. Figure 3.10(c) shows performability versus standard noise deviation and voltage swing for an VOPD application mapping to a Torus topology. The ranges used for both V_{sw} and σ are the one used as constraints in the optimization problem. Results show that we can find multiple set of points to choose from to achieve the maximum performability. The selection area shown in the figure contains multiple set of points

that achieves a maximum performability of 0.99xx. The differences between these points, in terms of performability values, does not exceed 10^{-5}, which is not a vital difference compared to the original value. Results in Figure 3.10(c) also highlight the significant effect of the proposed methodology. Choosing inappropriate design value such as $V_{sw} = 0.59$ for a noise level of $\sigma = 0.005$ results in a degradation of 9.2% in the network performability compared to the value achieved by the proposed methodology.

3.8 Chapter Summary

A new methodology to acquire the optimum topology architecture to achieve maximum performability was presented in this chapter. The proposed methodology targets nine regular topologies aiming at maximizing NoC performability at early design phases. Graph-theoretic concepts were adopted to study the topological architecture of NoCs and propose a new topology-based model for network performability. Using the proposed model, we studied important design parameters, which significantly affect the performability of a NoC-based system; such as network topology architecture, traffic distribution, noise power, voltage swing, and probability of edge failure. This study shows the significant impact of these parameters on the system performability.

As a proof of concept, we validated the proposed methodology through a case study of a VOPD application. Results show that performability could have been degraded by 9.2% for the VOPD application, compared to the value achieved by the proposed methodology, if incorrect parameters were chosen.

This work has been published in part in the the proceedings of the IEEE International Conference on Design and Technology of Integrated Systems in Nanoscale

Era (DTIS'09) [87] and in full at the International Journal of Circuit Theory and Applications, Wiley [88].

Chapter 4

Design For Delay: A Topology-based Approach

4.1 Introduction

In the past few years, various methodologies have been proposed to design NoC-based applications [89, 90]. Based on the target application requirements and constraints, researchers have developed design methodologies that focus on optimizing specific design parameters such as power consumption [22, 26], routing algorithms [91], network throughput [92], and delay [93]. Network delay is one of the most important design parameters because optimizing the network delay could eliminate the network congestion and significantly improve the overall system performance. Among many design factors, the delay criterion of a system could be studied at different levels of abstraction.

At early design phases, the network architecture must be designed efficiently to achieve a minimum network delay. However, at early design phases, the exact physical layout structure is not yet defined. Hence, designers do not have enough layout information to choose the most efficient network topology for their target

application. In this chapter, we address this problem by: 1) studying the impact of the network topology on NoC system delay using graph-theoretic concepts, 2) developing a topology-based average delay model for NoC-based systems, and 3) using the proposed model to improve the delay of a given NoC application at early design phases. This chapter presents a new 2-D Markovian M/D/1/B queue model, which is developed for NoC router. The 2-D model gives an accurate representation for a queue performance when a deterministic service rate is applied. This queue can serve in a wide range of applications such as video and telecommunication application where a deterministic access rate between internal PE cores is used.

The proposed model is used to select the optimal topology that achieves the minimum network delay for a given application, taking into consideration the possible changes in the network traffic distribution. Finally, experimental work is carried out to validate the proposed model.

This chapter is organized as follows. Section 4.2 highlights related work. Section 4.3 presents our proposed network delay model. Section 4.4 starts by studying the queue model proposed in Subsection 4.3.2, then we present a case study to show how the proposed network delay model could be used to select the optimal topology ($\mathcal{H}$) that achieves the minimum network delay. Finally, the chapter summary is presented in Section 4.5.

4.2 Related Work

Researchers addressed NoC delay from different perspectives [6]. The research in this area can be categorized under two titles: router-delay modeling and network-delay modeling. Few researchers focused only on the router such as Chien et al. [94], who proposed a delay model for wormhole and virtual channel routers. However, Chien's model was designed for 0.8-micron CMOS and cannot be applied for pipelined

architectures. Lopez et al. [95] proposed an extension to Chien's model for pipelined routers. But Lopez's model assumes that the time duration of the clock cycle depends on the router latency, which is not a practical assumption. Peh et al. [96] proposed a delay model for a variable pipelined wormhole router with fixed time cycle to address the Lopez's model problem. However, these models did not consider the impact of changing router design parameters (such as queue size) on its delay. Optimizing buffers is an essential task since buffers are a major source of cost and power consumption [97]. Therefore, the model proposed in this chapter is developed to address this problem.

In the second category, researchers tried to improve the network delay by analyzing analytical models and proposing new techniques to improve the network delay. In [98], authors tried to improve the delay by transporting data payloads and control information on separate planes, optimized for bandwidth and latency respectively. In [99], authors use code-division multiple access (CDMA) in NoC to eliminate the data transfer latency variations by sharing the data communication media among multiple users concurrently. In [100], authors used dynamic task mapping to improve packet latency.

Other techniques have been introduced to minimize network delay such as combining networks and bus architectures to construct a hybrid interconnect [101], and employing double-data-rate, wave-pipelined interconnect [102].

For delay modeling, an example is the work in [103], which proposed a delay computing-model for a 2D-mesh wormhole based NoC architecture. Using genetic algorithm, authors of [103] provide a mapping technique for large scale NoCs to achieve a minimum NoC average delay. However, the work in [103] was applied only to mesh networks and did not study or consider any other network topology, which limits the impact of the network architecture design factor. In this chapter, we address this limitaion by exploring the impact of changing the NoC topology on the network

delay for a given application.

For 3-D topologies, a design tool, SunFloor 3D, is introduced in [104, 105] to synthesize application-specific 3-D NoCs and an analytic model is proposed for the zero-load latency in [106]. The proposed model in [106] considers the effects of the topology on the performance of a 3-D NoC. However, zero-load latency is not an accurate assumption for practical applications. Also, the proposed model in [106] requires detailed physical layout information which might not be available at early design phases. Our work tackles this issue by proposing a model that could be used at high level of abstraction.

4.3 Proposed Model for Network Delay

The sources of delay in the network could be classified into two groups: delay from the links (T_l) and delay from the routers (T_r). The average network delay ($\mathcal{T}_{n-av}$) could be defined as follows

$$\mathcal{T}_{n-av} = \frac{T_l + T_r}{\sum_{i=1}^{n} \sum_{j=1}^{n} \lambda_{i,j}} \tag{4.1}$$

The derivation of the mathematical formulas that describe these two types of delay is presented in the following subsections.

4.3.1 Link's Delay (T_l)

Based on the target topology, and assuming shortest path routing, the delay of global interconnection links in a target topology is measured at an abstract level by calculating how many time units are needed to transmit all data packets given in the TDG. This measurement is done under the assumption that one time unit is needed to transmit a single packet over a unit length. In this chapter, we use

the same graph-theoretic concepts that have been mentioned in Subsection 2.3.1 and Section 3.3.

At system level, the time consumed in the global links of a network topology can be measured by

$$T_l \;=\; \sum_{i=1}^{n} \sum_{j=1}^{n} \lambda_{i,j} \;\cdot\; c_{i,j} \;\cdot\; u_t \tag{4.2}$$

where i and j are the destination and source node indexes, respectively, $\lambda_{i,j}$ is the average number of packets per time step associated with each logical link. $c_{i,j}$ is the minimum number of links needed to transmit these packets from their source to destination when routed on a certain topology, and u_t represents a unit time. This multiplication is performed for each entry in a given TDG. Assuming a unit time is consumed when a packet is transmitted over a unit length, the summation (T_l) represents the total global links delay consumption.

Because the exact physical layout structure is not usually defined at early design phases, the connectivity matrix (C) is used in (4.2) based on the assumption that links between routers have equal lengths. However, a modified connectivity matrix (C_M) could be used after synthesis, Static Timing Analysis (STA), or placement and routing to reflect the exact physical layout information. The modified connectivity matrix (C_M) could be generated using the Hadamard product of C and $\mathcal{D}$ matrices as shown in (3.7), which is repeated below

$$C_M = C \times \mathcal{D} \tag{4.3}$$

where $\mathcal{D}$=$[d_{ij}]$ is a matrix that represents the relative physical lengths of the links between routers after placement and routing. The entries of this matrix could be extracted from the ASIC design tool after placement and routing. Then, C could be replaced by C_M in (3.5), in a back-annotation step, to acquire the exact network

delay.

4.3.2 Router's Delay (T_r)

Figure 4.1 shows the architecture of an n-port output-queue router. Router datapath consists of four main modules: input buffer, Switch Fabric (SF), output queue, and output buffer, whereas the controller part is implemented as a finite state machine (FSM). Packets arrive at the input of the router asynchronously with a packet ready signal indicator (PR). The packet arrival rate depends on the target application [107–109]. The controller reads the packet header then directs the packet to the corresponding output queue. The configuration commands are generated based on routing tables in the controller. For n input ports, n output ports, there are n queues for each output port serving as FIFO buffers. Finally, a round robin scheduler algorithm serves backlogged queues one after another in a fixed order in the output. Packets are sent with a Packet Sent (PS) signal indicator to the next hop.

In output-queue routers, packet loss may occur if burst transmission is not handeled correctly. Burst transmission occurs when two or more packets, that arrive in adjacent clock cycles from one source, target the same output port. Based on the NA interface protocol and router QoS, a proper queue size should be chosen to reduce the packet loss probability. For output-queue routers, the delays for one packet can be modeled as follows.

Estimation of input buffer delay (T1)

The delay T_1 depends on the setup time of the input buffer t_{isetup} and the time period between the packet arrival and the next clock falling edge t_w as shown in Figure 4.2. Once packets are synchronized with the clock falling edge, the setup time in the

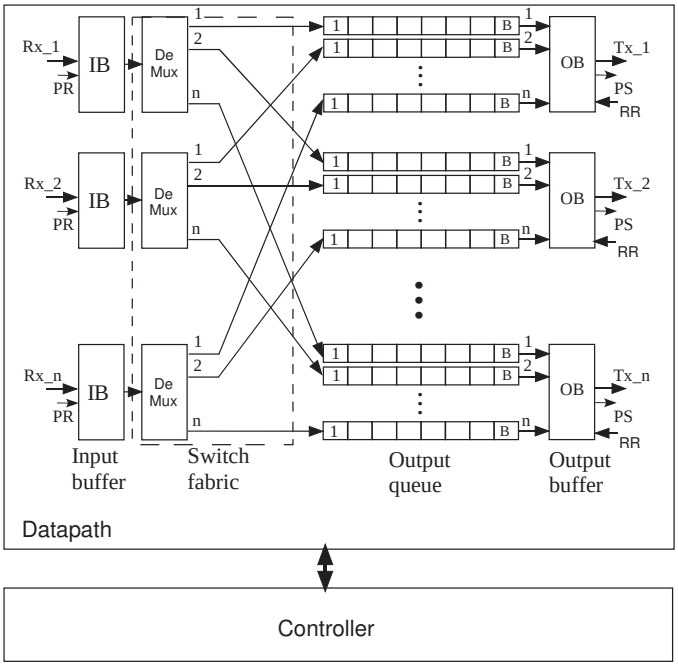


Figure 4.1: Output-queue router architecture: Packet Ready (PR), Packet Sent (PS), and Receive Ready (RR) signals are used for synchronization purpose.

following stages is negligible. Thus,

$$\begin{aligned} T_1 &= t_w + t_{clk},\ t_w < t_{isetup} \\ &= t_w,\ t_w > t_{isetup} \end{aligned} \tag{4.4}$$

Assuming that the maximum values of the time period between the packet arrival and the next clock falling edge is

$$t_w = \frac{t_{clk}}{2} \tag{4.5}$$

The maximum value of T_1 can be written as

$$T_1 = \frac{3}{2}\, t_{clk} \tag{4.6}$$

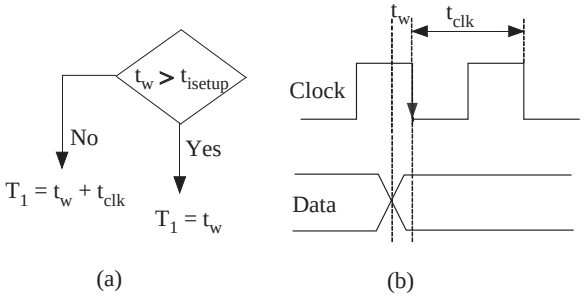


Figure 4.2: T_1 depends on the setup time of the input buffer t_{isetup} and the time period between the packet arrival and the next clock falling edge t_w: (a) T_1 probabilities. (b) Timing diagram.

Estimation of switch fabric delay (T2)

The delay T_2 is due to the switch fabric (SF) propagation delay. SF needs only one clock cycle to direct the received packet to the appropriate output queue based on updated routing tables. This assumption matches the moore state machine implementation of the switching process.

$$T_2 \;=\; t_{clk} \tag{4.7}$$

Estimation of the queue delay (T3)

The delay T_3 is due to the queue propagation delay. In this section, we propose our 2-D *M/D/1/B* queue model to acquire the delay.

An M/D/1/B queue model is used in NoC routers for different reasons. For example, if a dynamic memory refresh routine is activated periodically, a queue with deterministic service rate would be an appropriate model that is suitable for traffic characteristics between the memory and microprocessor. In this section, we derive an *M/D/1/B* queue model which matches routers that have random packet arrival rate and deterministic service rate. The model is driven for a size B queue, as shown in Figure 4.3, as a function of packet arrival probability a, service rate *1/n*, and packet

loss probability L.

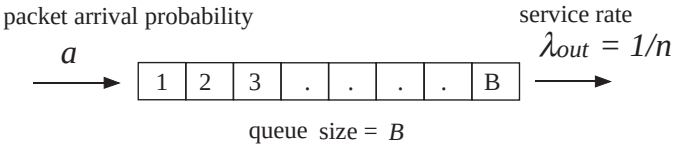


Figure 4.3: The arrival probability and departure rate for an M/D/1/B queue.

In this model we assume single packet arrival, single packet departure per time step provided that time step T is given by

$$T < min\{\tau_a, \tau_s\} \tag{4.8}$$

where τ_a is the average inter-arrival time τ_s is the service time.

Using Markov chain analysis, the state transition diagram is shown in Figure 4.4. In this figure, states in each column represent the queue occupancy starting from state 0 ($s_{00} : s_{n0}$) which represents an empty queue to state B ($s_{0B} : s_{nB}$) which represents a fully occupied queue. Rows in the transition diagram depend on the packet service rate λ_{out}. The service rate in this model is based on a round-robin scheduler and equals a deterministic value of *1/n*. Hence, the state transition diagram extends to *n-1* rows.

The state vector can be written as a set of sub-vectors:

$$\overrightarrow{S} = [\ \overrightarrow{S_0}\ \ \overrightarrow{S_1}\ \ \overrightarrow{S_2}\ \ \cdots\ \ \overrightarrow{S_B}\ ]^t \tag{4.9}$$

where $\overrightarrow{S_0}$ is a set represents an empty queue state and $\overrightarrow{S_B}$ is a set represents a full queue stat, and each $\overrightarrow{S_i}$ represents a vector as follows

$$\overrightarrow{S} = [\ s_{0,0}\ s_{1,0}\ ...\ s_{n-1,0} \qquad s_{0,1}\ s_{1,1}\ ...\ s_{n-1,1} \qquad \cdots \qquad s_{0,B}\ s_{1,B}\ ...\ s_{n-1,B}\ ]^t \tag{4.10}$$

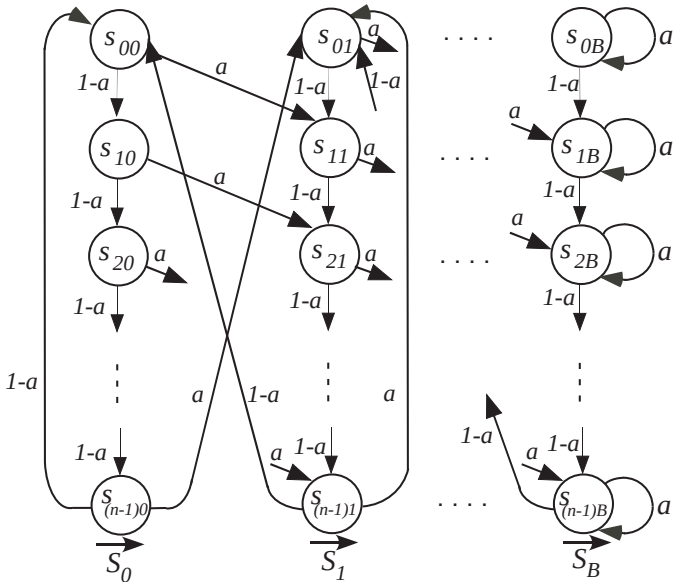


Figure 4.4: 2-D M/D/1/B queue state transition diagram.

Since the packet arrival rate is independent from the packet departure rate, and based on the assumption that packets can be served at the same time step with a packet service rate of λ_{out}, the transition matrix $\mathbf{P}$ can be written as:

$$\mathbf{P} = \begin{bmatrix} P_{0,0} & P_{0,1} & \cdots & P_{0,B-1} & P_{0,B} \\ P_{1,0} & P_{1,1} & \cdots & P_{1,B-1} & P_{1,B} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ P_{B-1,0} & P_{B-1,1} & \cdots & P_{B-1,B-1} & P_{B-1,B} \\ P_{B,0} & P_{B,1} & \cdots & P_{B,B-1} & P_{B,B} \end{bmatrix} \tag{4.11}$$

For 2-D model, each $P_{i,j}$ element is an $n\times$ n matrix. A quick look at Figure 4.4 shows that states in each column (j) in our model correlates only with (*j-1 , j , j+1*) columns, the transition matrix turns to be a diagonal matrix

$$\mathbf{P} = \begin{bmatrix} P_{0,0} & P_{0,1} & 0 & \cdots & 0 & 0 & 0 \\ P_{1,0} & P_{1,1} & P_{1,2} & \cdots & 0 & 0 & 0 \\ 0 & P_{2,1} & P_{2,2} & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & P_{B-2,B-2} & P_{B-2,B-1} & 0 \\ 0 & 0 & 0 & \cdots & P_{B-1,B-2} & P_{B-1,B-1} & P_{B-1,B} \\ 0 & 0 & 0 & \cdots & 0 & P_{B,B-1} & P_{B,B} \end{bmatrix} \tag{4.12}$$

Because of state transition similarities shown in Figure 4.4, the transition matrix **P** consists of six sub-matrixes (A,B,C,D,E, and F) in the following order

$$\mathbf{P} = \begin{bmatrix} A & C & 0 & \cdots & 0 & 0 & 0 \\ B & D & C & \cdots & 0 & 0 & 0 \\ 0 & E & D & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & D & C & 0 \\ 0 & 0 & 0 & \cdots & E & D & C \\ 0 & 0 & 0 & \cdots & 0 & E & F \end{bmatrix} \tag{4.13}$$

Where:

$$A = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 & 0 & 1-a \\ 1-a & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 1-a & 0 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & 0 & \cdots & 1-a & 0 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 1-a & 0 \end{bmatrix} \tag{4.14}$$

$$B = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 & 0 & a \\ a & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & a & 0 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & 0 & \cdots & a & 0 & 0 \\ 0 & 0 & 0 & \cdots & 0 & a & 0 \end{bmatrix} \tag{4.15}$$

$$C = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 & 0 & 1-a \\ 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 & 0 \end{bmatrix} \tag{4.16}$$

$$D = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 & 0 & a \\ 1-a & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 1-a & 0 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & 0 & \cdots & 1-a & 0 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 1-a & 0 \end{bmatrix} \tag{4.17}$$

$$E = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ a & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & a & 0 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & 0 & \cdots & a & 0 & 0 \\ 0 & 0 & 0 & \cdots & 0 & a & 0 \end{bmatrix} \tag{4.18}$$

and

$$F = \begin{bmatrix} a & 0 & 0 & \cdots & 0 & 0 & 0 \\ 1-a & a & 0 & \cdots & 0 & 0 & 0 \\ 0 & 1-a & a & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & a & 0 & 0 \\ 0 & 0 & 0 & \cdots & 1-a & a & 0 \\ 0 & 0 & 0 & \cdots & 0 & 1-a & a \end{bmatrix} \tag{4.19}$$

Using the transition matrix $\boldsymbol{P}$, we can drive an expression for the equilibrium distribution vector **S**

$$\overrightarrow{S} = \begin{bmatrix} \overrightarrow{S_0} & \overrightarrow{S_1} & \overrightarrow{S_2} & \cdots & \overrightarrow{S_B} \end{bmatrix}^t, \text{ where } \sum_{i=0}^{n-1} \sum_{j=0}^{B} s_{ij} = 1 \tag{4.20}$$

At steady state, the distribution vector $\overrightarrow{S}$ satisfies the equation:

$$\mathbf{P}\ \overrightarrow{S}\ =\ \overrightarrow{S} \tag{4.21}$$

We can write (4.21) as

$$\begin{bmatrix} A & C & 0 & \cdots & 0 & 0 & 0 \\ B & D & C & \cdots & 0 & 0 & 0 \\ 0 & E & D & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & D & C & 0 \\ 0 & 0 & 0 & \cdots & E & D & C \\ 0 & 0 & 0 & \cdots & 0 & E & F \end{bmatrix} \cdot \begin{bmatrix} \overrightarrow{S_0} \\ \vdots \\ \overrightarrow{S_B} \end{bmatrix} = \begin{bmatrix} \overrightarrow{S_0} \\ \vdots \\ \overrightarrow{S_B} \end{bmatrix} \tag{4.22}$$

From (4.22):

$$A \cdot \overrightarrow{S_0} \; + \; C \cdot \overrightarrow{S_1} \; = \; \overrightarrow{S_0} \tag{4.23}$$

$$B \cdot \overrightarrow{S_0} \; + \; D \cdot \overrightarrow{S_1} \; + \; C \cdot \overrightarrow{S_2} \; = \; \overrightarrow{S_1} \tag{4.24}$$

$$E \cdot \overrightarrow{S_1} \; + \; D \cdot \overrightarrow{S_2} \; + \; C \cdot \overrightarrow{S_3} \; = \; \overrightarrow{S_2} \tag{4.25}$$

$$E \cdot \overrightarrow{S_{i-1}} \; + \; D \cdot \overrightarrow{S_i} \; + \; C \cdot \overrightarrow{S_{i+1}} \; = \; \overrightarrow{S_i}, \; 1 \; \leq \; i \; \leq \; B-1 \tag{4.26}$$

$$E \cdot \overrightarrow{S_{B-1}} \; + \; F \cdot \overrightarrow{S_B} \; = \; \overrightarrow{S_B} \tag{4.27}$$

From (4.23), we can write

$$\overrightarrow{S_1} \; = \; C^{-1}[I - A] \; \overrightarrow{S_0} \tag{4.28}$$

where

$$\overrightarrow{S_0} = [\; s_{0,0} \; s_{1,0} \; s_{2,0} \ldots \; s_{n-1,0} \;]^t \tag{4.29}$$

and similarly

$$\overrightarrow{S_{i+1}} \; = \; C^{-1} \; [(I - D) \; \overrightarrow{S_i} - \; E \; \overrightarrow{S_{i-1}}], \; 1 \; \leq \; i \; \leq \; B-1 \tag{4.30}$$

$$\overrightarrow{S_B} \;=\; [I - F]^{-1}\; E\; \overrightarrow{S_{B-1}} \tag{4.31}$$

From (4.28), the difference equations for the state probability vector are given by

$$s_{1,0} \;=\; s_{0,0}\,(1-a) \tag{4.32}$$

$$s_{2,0} \;=\; s_{1,0}\,(1-a) \tag{4.33}$$

$$s_{n,0} \;=\; s_{n-1,0}\,(1-a) \tag{4.34}$$

The solution to these difference equations can be written in its general form as

$$s_{i+1,0} = s_{i,0}\,(1-a), \quad 0 \le i \le n-1 \tag{4.35}$$

Using (4.35), we can obtain the state distribution vector. We wrote a Matlab program to solve (4.22) for $\boldsymbol{S}$ using eigenvalues and eigenvectors of matrix $\boldsymbol{P}$, and using (4.28), (4.30), and (4.31). The simulation results are discussed in details in Section 4.4

The average output traffic from the *M/D/1/B* queue N_0:

$$\begin{aligned} N_0 \;&=\; \lambda_{out} \sum_{i=0}^{n-1} \sum_{j=1}^{B} s_{ij} \\ &=\; \frac{(1 - \sum_{i=0}^{n-1} s_{i0})}{n} \le \frac{1}{n} \end{aligned} \tag{4.36}$$

and the average input traffic to the queue is given by

$$N_i \;=\; a \tag{4.37}$$

The efficiency of the *M/D/1/B* queue equals:

$$\eta \;=\; \frac{N_o}{N_i} \;=\; \frac{1 - \sum_{i=0}^{n-1} s_{i0}}{a\; n} \le \frac{1}{a\; n} \tag{4.38}$$

and an expression of the packet loss probability can be written as

$$\begin{aligned} L &= 1 - \eta \\ &= 1 - \frac{1 - \sum_{i=0}^{n-1} s_{i0}}{a\ n} \end{aligned} \tag{4.39}$$

For *M/D/1/B* model, the average number of time steps a packet spends inside the queue is given by little's law

$$T_{av} = \frac{B_{av}}{N_o} \tag{4.40}$$

Where

$$B_{av} = \sum_{i=0}^{n-1} \sum_{j=1}^{B} j\ s_{ij} \tag{4.41}$$

From (4.36), we can get the maximum throughput of a queue when packets arrive with various arrival rates to obtain the maximum burstiness ratio that will not result in packet loss. From (4.39), we can calculate the packet loss probability when packets arrive with arrival probability of a and served with an *(1/n)* service rate. Also, we can use (4.40) to calculate the queue delay.

If technology and physical layout allows using both clock edges to speed up the packet shift inside the queue, the packet processing speed is doubled inside the queue. Hence, T_3 can be expressed in terms of clock cycles as

$$T_3 = \frac{T_{av}}{2}\ \cdot\ t_{clk} \tag{4.42}$$

Estimation of output buffer delay (T4)

The delay T_4 is due to the output buffer which is synchronized with the master clock falling edge. Because the last memory element in the queue is also activated at the falling edge, assuming an even queue size, packet will spend one clock cycle to be

delivered from the queue to the output port through the output buffer.

$$T_4 \;=\; t_{clk} \tag{4.43}$$

Assuming that $u_t = t_{clk}$, the total router delay T_{router} can be written as

$$T_{router} \;=\; \frac{7 \;+\; T_{av}}{2}\; u_t \tag{4.44}$$

Total routers' delay for the whole network

To calculate the total routers' delay for the whole network, we need to create two new variables: a router-count matrix (R) and a port-count vector (M). Using the connectivity matrix, discussed in Subsection 4.3.1, and the the application traffic distribution matrix, shown in Subsection 2.3.1 and Section 3.3, a new unique router-count matrix ($R \;=\; [r_{ij}]$) is generated for each topology ($\mathcal{H}$). Each entry in this matrix represents the minimum number of routers a packet goes through during its transition from the source node to the destination node. The matrix (R) could be obtain by

$$r_{ij} \;=\; c_{ij} \;+\; 1,\; \forall\; \lambda_{ij} \;\neq\; 0 \tag{4.45}$$

For each one of the network topologies shown in Figure 3.1, an n $\times$ n adjacency matrix $A = [a_{i,j}]$ is another representation of the graph and shows all the direct connections between its routers [110]. For any A matrix, $a_{i,j} \;\in\; \{0, 1\}$, where

$$\begin{aligned} a_{i,j} \;&=\; 1,\; \text{if there is a direct link from node } v_i \text{ to node } v_j \\ a_{i,j} \;&=\; 0,\; \text{otherwise} \end{aligned}$$

For example, the adjacency matrix of the mesh topology shown in Figure 3.1(a) could be written as

$$\mathrm{A} = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \end{bmatrix}$$

A port-count vector ($M = [m_i]$), which represents the number of ports per router could be obtained by

$$m_i = \sum_{j=1}^{n} a_{i,j} + 1 \tag{4.46}$$

For example, the port-count vector of the mesh topology shown in Figure 3.1(a) could be written as

$$\mathrm{M} = \begin{bmatrix} 3 & 4 & 3 & 4 & 5 & 4 & 3 & 4 & 3 \end{bmatrix}$$

The first element m_1 shows that the first router in Figure 3.1(a) has 3 ports; two router-to-router ports and one PE-to-router port. To calculate the time consumed inside the routers for all packets in a given application.

From (4.44), (4.45), and (4.46), the total router-delay in the network for a give application could be written as

$$T_r = \sum_{i=1}^{n} \sum_{j=1}^{n} \left(\lambda_{ij} \cdot \sum_{x=1}^{r_{ij}} T_{router}(m) \right) \tag{4.47}$$

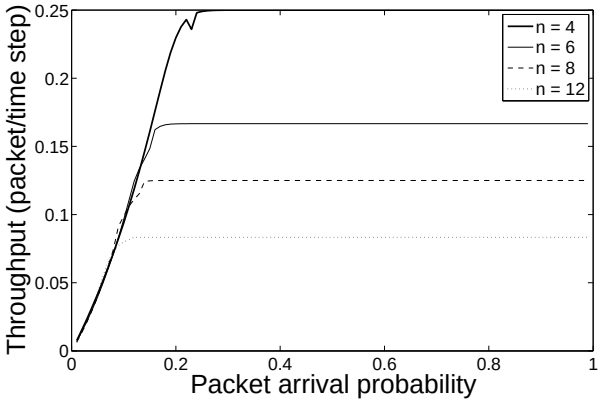


Figure 4.5: Throughput versus packet arrival rate for fixed queue size ($B = 8$).

4.4 Experimental Results

In this section, we start by studying the queue model proposed in Subsection 4.3.2, then we present a case study to show how the network average delay model ($\mathcal{T}_{n-av}$) proposed in (4.1) could be used to select the optimal topology ($\mathcal{H}$) that achieves the minimum network delay.

4.4.1 Analysis of Output-Queue Model

We used (4.36), (4.39), and (4.22) to study the relation between the packet loss probability (L), packet inter-arrival probability (a), number of ports (n), throughput (N_o), and queue size (B). The queue size and service rate are chosen to match the requirements of various NoC topologies [6].

Figure 4.5 presents the throughput versus packet arrival probability for different number of states *(n)* when a fixed queue size *B=8* is chosen. As shown in Figure 4.5, changing the service rate, which is represented in the model by number of states *(n)*, impacts the throughput significantly. For instance, for *a=0.5*, changing the number

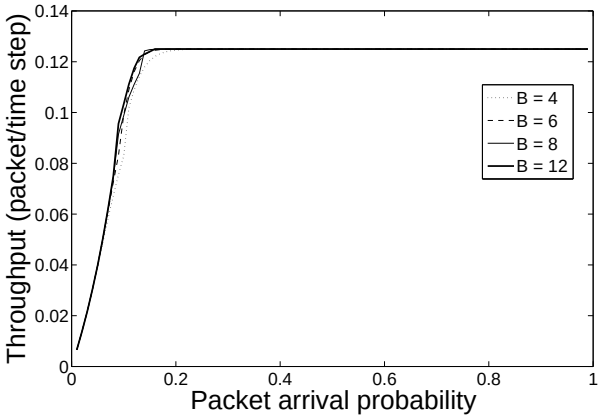


Figure 4.6: Throughput versus packet arrival rate for fixed number of states ($n = 8$).

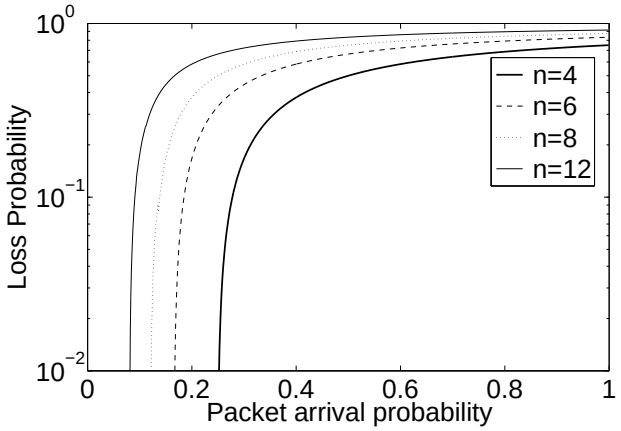


Figure 4.7: Loss probability versus packet arrival rate for fixed queue size ($B = 8$).

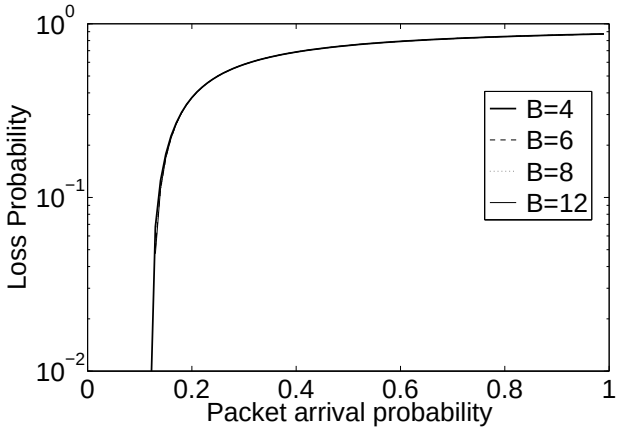


Figure 4.8: Loss probability versus packet arrival rate for fixed number of states ($n = 8$).

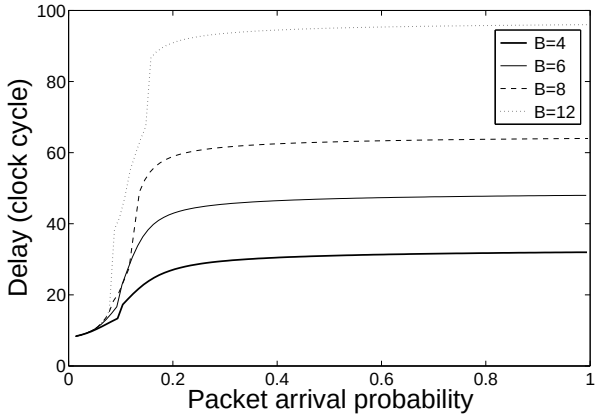


Figure 4.9: Delay versus packet arrival probability for n=8.

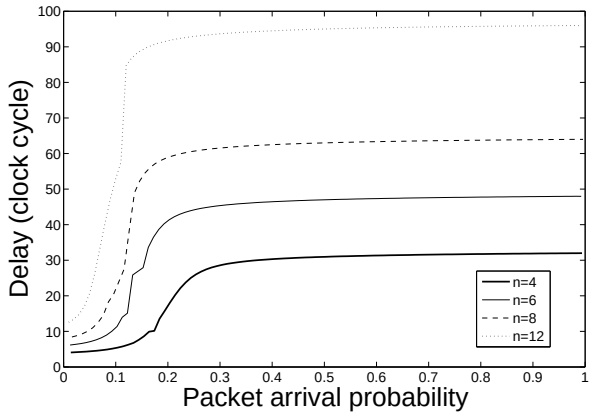


Figure 4.10: Delay versus packet arrival probability for B=8.

of states from 4 to 12 reduces the throughput by 66.7%. From this figure, we can see that the throughput maximum value equals service rate *(1/n)* for M/D/1/B model.

On the other hand, Figure 4.6 shows that changing the queue size does not have noticeable effect on the throughput. This can be justified by dividing the curve into two sections. The first section starts on Y axis from 0 to less than 0.125 (just before the throughput curve reaches its maximum level). We can barely notice a change in the throughput curve in this area and that is because, in this experiment, we change the queue size from 4 to 12 which is a very short range. However, we did this little change intentionally to represent a realistic implementation for NoC application. We can easily run our program for higher queue sizes but it will not be a realistic assumption. Unlike computer networks, we have limited silicon area in NoC applications and increasing the queue size means adding more memory elements. In subsection 4.4.2, we proved that an implementation of an 8-port output-queue router that has an 8-bit queue and 4-queues per port costs more than 44K equivalent gates. The second part of the figure is when the throughput reaches its maximum value. All curves are overlapped in this area and that is because the maximum throughput value cannot exceed the service rate, which is *1/8* in this experiment.

Figure 4.7 presents the packet loss probability versus packet arrival rate for fixed queue size B=*8* and different number of states *(n)*. The simulation results show that the change of the number of states *(n)*, which is corresponding to packet service rate, impacts the loss probability. For instance, for a=*0.5*, changing the number of states from 4 to 12 increases the packet loss probability by 39.9%. To analyze the effect of changing queue size on the packet loss probability, Figure 4.8 presents the packet loss probability versus packet arrival rate for different queue sizes *(B)* when fixed number of states n=*8* is chosen. As shown in the figure, the change in the queue size does not impact the loss probability.

Figure 4.10 shows the queue delay versus packet arrival probability for B=8. We used (4.40) to study the effect of changing the service rate on the router delay for a fixed queue size B. In this experiment, B is chosen to be 8 while n = 4, 5, and 8 to match various NoC topologies [76]. For 80% packet arrival probability, changing the number of ports from 4 to 8 increases the router delay from 20 to 36 clock cycles.

4.4.2 A Case Study

In this subsection, we used the proposed network average delay model to select the optimal topology ($\mathcal{H}$) that achieves the minimum network delay for a given application. Figure 4.11 shows a TDG typical for video applications (MPEG4 core) discussed in [61]. The numbers written on the arrows are the average number of packets transmitted and the numbers written on the circles represent the PEs' numbers.

The traffic distribution matrix (λ) which represents the initial mapping of the MPEG4 shown in Figure 4.11 is given by

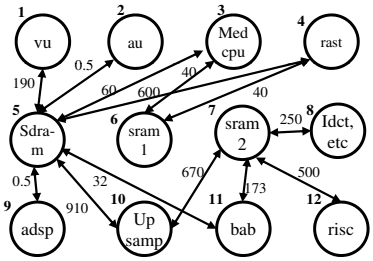


Figure 4.11: MPEG4 core TDG.

$$\lambda = \begin{bmatrix}
0 & 0 & 0 & 0 & 190 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 60 & 40 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 600 & 40 & 0 & 0 & 0 & 0 & 0 & 0 \\
190 & 0.5 & 60 & 600 & 0 & 0 & 0 & 0 & 0.5 & 910 & 32 & 0 \\
0 & 0 & 40 & 40 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 250 & 0 & 670 & 173 & 500 \\
0 & 0 & 0 & 0 & 0 & 0 & 250 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 910 & 0 & 670 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 32 & 0 & 173 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 500 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}$$

For a given application represented by a TDG and a traffic distribution matrix (λ), we aim to

$$\text{Minimize } \mathcal{T}_{n-av} \;=\; f(\mathcal{H}, C, R, \lambda) \tag{4.48}$$

subject to:

$$c_{ij} \;=\; c_{ji}$$

where $C \;=\; [c_{ij}]$ represents the connectivity matrix as discussed in Subsection 4.3.1.

To solve this problem, we generated connectivity matrices for all nine topologies shown in Figure 3.1. The network topologies in Figure 3.1 are extended as explained in [65] to connect 12 PEs. Then, a search algorithm developed in MATLAB® is used to calculate the average network delay using (4.1) to find the optimum topology/mapping pair that achieves minimum network delay ($\mathcal{T}_{n-av}$). Figure 4.12

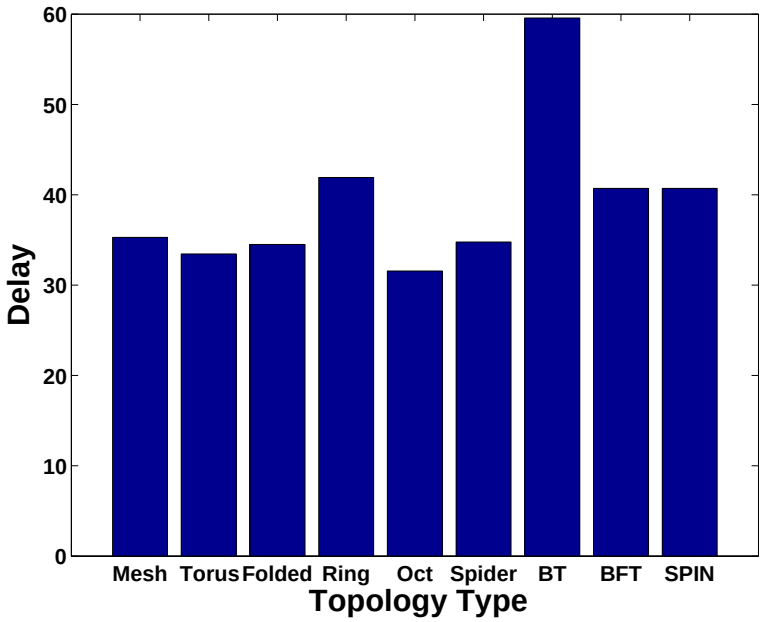


Figure 4.12: Average network delay for different network topologies.

shows the calculated average network delay for different network topologies. Octagon topology is found to be the one that could achieve minimum average network delay, whereas BT is the one that has maximum average network delay. The PE mapping for Octagon is generated as shown in Figure 4.13. Routers are represented by the dark squares while PEs are represented by the white squares. The reason behind this result is that Octagon has an architecture that allows shorter packet transmission paths, whereas BT architecture forces packets to go through longer paths as the number of levels in the tree increases. Figure 4.12 shows that the average network delay could have been increased by 90.8% if a wrong topology, such as BT, was chosen instead of Octagon for the MPEG4 application, which highlights the importance and efficiency of the presented work.

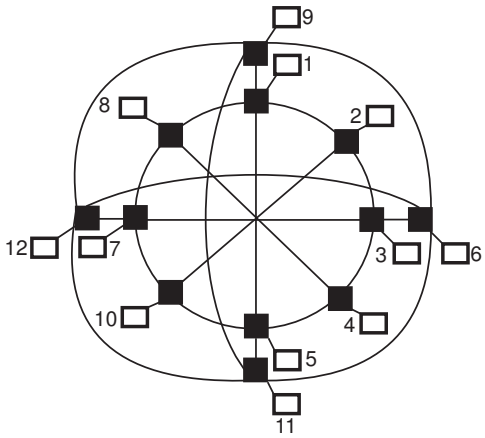


Figure 4.13: Mapping of Video Object Plane Decoder application on to Octagon topology.

4.5 Chapter Summary

A topology-based model is developed to calculate the network average delay for NoC-based applications. Using graph-theoretic concepts, the proposed model uses the application distribution graph, connectivity and adjacency matrices to measure links' and routers' average delays. We used Markov chain analysis to derive 2-D model for M/D/1/B queue. We show how can this model be used to analyze the impact of the packet arrival rate, number of ports, and queue size on the router performance. The experimental results show that changing the service rate significantly impacts the throughput and hence, the efficiency. A case study of MPEG4 video application is presented to explain how the proposed model could be used to minimize the network delay by selecting the best topology. Experimental results show that the average network delay could have been increased by 90.8% if a wrong topology was chosen.

This work has been published in the proceedings of the 6th International Workshop on System-on-Chip for Real-Time Applications [111], Lecture Notes in Computer Science [112], and in the Journal of Circuits, Systems, and Computers [113].

Chapter 5

NoC Design for Power, Performability, and Delay using Particle Swarm Optimization

In the past three chapters, analytical models for NoC power, performability, and delay have been proposed. We showed how these models could be used to optimally design the network topology so that it achieves the target design requirement for a given application. The discussion and analysis in each chapter were focused on each individual design parameter, which was important to study in detail the aforementioned design parameters and consider all variable that could impact their values.

In this chapter, a joint consideration of NoC power, performability, and delay is carried out simultaneously. We used Particle Swarm Optimization (PSO) to find the optimum network topology, that achieves minimum delay, maximum performability, and minimum power consumption, for a given NoC application.

5.1 Particle Swarm Optimization

Particle swarm optimization (PSO) is a population based stochastic heuristic optimization technique developed by Eberhart and Kennedy in 1995 [114, 115]. Inspired by social behavior of bird flocking or fish schooling, PSO is one of the Artificial Life (ALife) algorithms that shows how biological techniques could be used to tackle complex optimization problems [114, 115].

PSO is initialized with a group of random particles, which are the solutions. Then, PSO searches for the optimum solution by updating generations. In every iteration, each particle is updated by following two best values. The first one, personal best (pbest), is the best solution it has achieved so far, which is its position in the search-space. The second one, global best (gbest), is the the best value obtained so far by any particle in the population [116].

PSO shares the following common features with Genetic Algorithms (GAs):

1. Both algorithms start with a group of a randomly generated population.
2. Both have fitness values to evaluate the population.
3. Both update the population and search for the optimum with random techniques.

However, PSO does not have genetic operators like crossover and mutation. Particles update themselves with the internal velocity. They also have memory, which is important to the algorithm. Compared to GAs, the information sharing mechanism in PSO is significantly different [117]. In GAs, chromosomes share information with each other. So the whole population moves like a single group towards an optimal area. In PSO, only gbest gives out the information to others. It is a one-way information sharing mechanism. The evolution only looks for the best solution. Compared to

GA, all the particles tend to converge to the best solution quickly even in the local version in most cases [116].

Although researches used GAs to address several NoC issues such as: automating NoC low-power design [118], improving communication load distributions in the network [119], providing less network contentions [120], and addressing NoC mapping [121], PSO is becoming more attractive to researchers because it gives faster results and it has less computation complexity.

When modeling a D-dimensional optimization problem, each particle X_i in the swarm is represented by a vector in the D-dimensional space where [122]

$$X_i = (x_{i1}, x_{i2}, x_{i3}, ..., x_{iD})^T \tag{5.1}$$

Particle is global best position in the swarm at k^{th} step is given by $gbest_j^k$. Also, the best fitness evaluation *pbest* is designated by a vector $pbest_i$ where

$$pbest_i = (\text{pbest}_{i1}, \text{pbest}_{i2}, \text{pbest}_{i3}, ..., \text{pbest}_{iD})^T \tag{5.2}$$

V_i^k is the velocity vector of the i^{th} particle in the k^{th} step. V_i is given by [122]

$$V_i = (v_{i1}, v_{i2}, v_{i3}, ..., v_{iD})^T \tag{5.3}$$

The velocity and position of a particle at *k+1* step in D-dimensional space could be written as [116]

$$v_{ij}^{k+1} = wv_{ij}^k + c_1\text{rand}()(\text{pbest}_{ij}^k - x_{ij}^k) + c_2\text{Rand}()(\text{pbest}_j^k - x_{ij}^k) \tag{5.4}$$

$$X_{ij}^{k+1} = X_{ij}^k + V_{ij}^{k+1} \tag{5.5}$$

where c_1 and c_2 are constants known as acceleration coefficients and rand() and Rand()

are two separately random function in the range [0,1]; w is the inertia weight [116].

In this chapter, we used the proposed models for NoC power, performability, and delay in (2.7), (3.5), and (4.1), respectively to solve multi-objective optimization problem. In the following sections, we show how PSO could be used to find the optimum network topology/mapping that achieves minimum delay, power consumption, and maximum performability for a given application.

5.2 Problem Formulation

For a given application represented by a TDG and a traffic distribution matrix (λ), we want to choose the optimum network topology/mapping that achieves minimum delay, power consumption, and maximum performability, where the power consumption is calculated by (2.7)

$$\mathcal{P}_t \;=\; \sum_{i=1}^{n_r} \mathcal{P}_{ri} \;+\; \mathcal{P}_{gl} \tag{5.6}$$

the performability is calculated by (3.5)

$$\mathcal{R}_{\mathcal{H}} = \mathcal{P}_{\mathcal{H}n} \cdot \mathcal{P}_{\mathcal{H}r} \tag{5.7}$$

and the delay is calculated by (4.1)

$$\mathcal{T}_{n-av} \;=\; \frac{T_l \;+\; T_r}{\sum_{i=1}^{n} \;\sum_{j=1}^{n} \;\lambda_{i,j}} \tag{5.8}$$

Our possible solutions for the optimum network configuration (population) are set to be all possible mapping for the nine regular topologies shown in Figure 3.1, which is copied here in Figure 5.1 [41–49]. In this section, we develop a mathematical formula of a fitness function that considers the three design parameters (NoC power, performability, and delay) and, at the same time, allows different weights for each

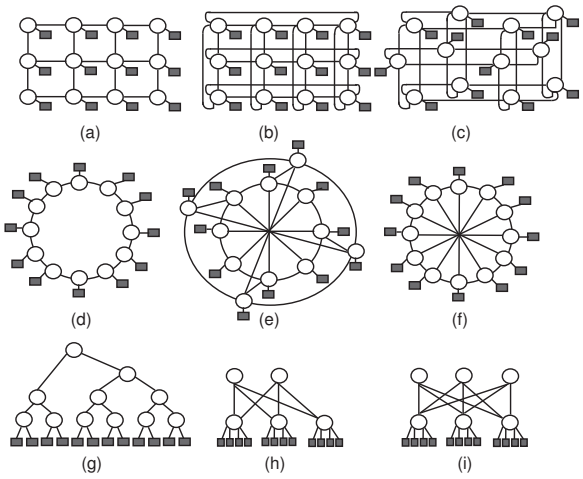


Figure 5.1: Nine regular NoC topologies: (a) Mesh. (b) Torus. (c) Folded Torus (Folded). (d) Ring. (e) Octagon (Oct). (f) Spidergon (Spider). (g) Binary tree (BT). (h) Butterfly Fat Tree (BFT). (i) SPIN. Routers are represented by white circles, whereas Processing Elements (PEs) are represented by dark squares.

parameter based on the target design requirements.

To select the optimum topology/mapping of a target application. PSO particles are constructed as shown in Figure 5.2. Each particle contains unique information that represents a possible topology type and the position (mapping) of each PEs in this topology. Moreover, it also contains possible choices of noise standard deviation (σ), voltage swing (V_s) and probability of edge failure (P_e). Based on the this particle contents, the fitness function is evaluated taking into consideration the target design constraints.

There are two approaches to formulate this problem mathematically, either using addition or using multiplication. If we consider addition, the design parameters (i.e. power, performability, and delay) have to be normalized to be able to add them as different measures/units in the same equation. The power and delay could be normalized to their maximum values whereas performability could be normalized to its minimum value within each topology so that each normalized value will have

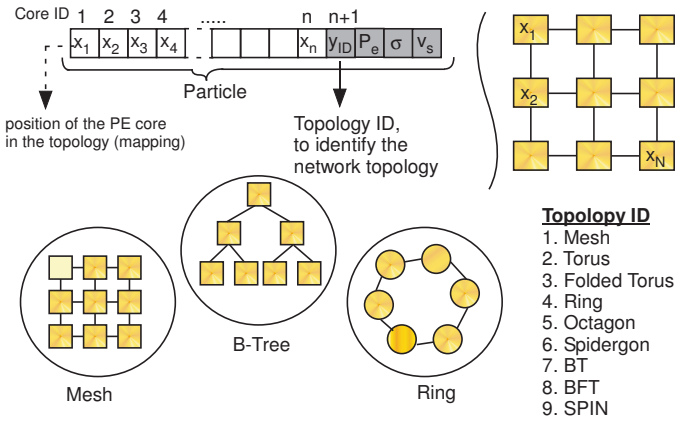


Figure 5.2: An example of PSO particle.

a range from 0 to 1. The maximum and minimum values could be calculated by assuming worst case scenario for each topology. As discussed in Subsection 2.3.2, the mapping/topology pairs are represented by a unique connectivity matrix C [57]. This matrix represents the minimum number of links a packet goes through during its transition from the source node to the destination node. Therefore, the worst case scenario will occur if all elements in the C matrix for a given topology $\mathcal{H}$ equals the network diameter, which is max(C). In such a case, $C_{max} = [c_{max_{ij}} = \max(C)]$.

However, this approach could result in unfair representation of the three design parameters. Although the values of all three normalized parameters are expected to range from 0 to 1, these values might not have the same weight and could vary by an order of magnitude. In such a case, the summation results, which is the fitness function, might be dominated by only one or two design parameters and the rest will be ignored. To avoid this problem and to achieve fair representation of each design parameter, the multiplication approach is considered. The fitness function in this case will change by the same ratio when any design parameter changes.

From the above discussion, the optimization problem could be formulated as follows. For a given application represented by a TDG and a traffic distribution ma-

trix (λ), it is required to find the optimum network topology/mapping pair by minimizing the following fitness function.

$$\text{Minimize } \Phi \;=\; \frac{\mathcal{P}_t^{\alpha_a} \;\cdot\; \mathcal{T}_{n-av}^{\alpha_c}}{\mathcal{R}_{total}^{\alpha_b}} \tag{5.9}$$

Subject to:

$$\begin{aligned} c_{ij} \quad &= \; c_{ji} \\ V_{swa} \quad &\leq \; V_{sw} \;\leq \; V_{swb} \\ \sigma_a \quad &\leq \; \sigma \;\leq \; \sigma_b \\ P_e \quad &\geq 0.1 \end{aligned}$$

where $C \;=\; [c_{ij}]$ represents the connectivity matrix as discussed in Subsection 3.4.2. V_{swa} and V_{swb} are the minimum and maximum permissible swing voltage according to design requirements of a given application. σ_a and σ_b are the margins of the expected noise standard deviation, P_e is the probability of edge failure. $\alpha_a,\ \alpha_b, and\ \alpha_c$ are user-defined weight coefficients for power, performability, and delay, respectively, $\alpha_a,\ \alpha_b,\ \alpha_c \;\geq\; 0$. The time complexity of PSO algorithm is O(n+4).

5.3 A Case Study

In this section, we present an example to show the significance of the the proposed design technique, in considering the delay, power, and performability simultaneously, on choosing the most suitable network architecture for a given application.

Figure 5.3 shows a typical TDG for H.263 MP3 decoder [81]. The numbers written on the arrows are the number of packets/time step transmitted and the numbers written on the circles represent PEs' numbers. The node descriptions of

Figure 5.3 is listed in Table 5.1.

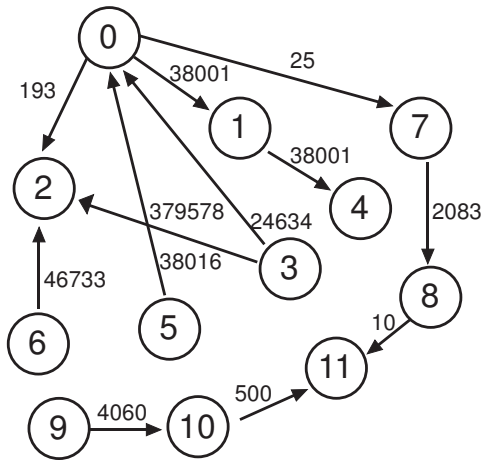


Figure 5.3: Traffic distribution graph of H.263 encoder MP3 decoder.

From this TDG, the traffic distribution matrix (λ), which represents the initial mapping of the MP3 decoder, shown in Figure 5.3, is given by

$$\lambda = \begin{bmatrix}
0 & 38001 & 193 & 0 & 0 & 0 & 0 & 25 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 38001 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
24634 & 0 & 37958 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
38016 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 46733 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2083 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 10 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 4060 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 500 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}$$

Following that, connectivity matrices are generated for all nine topologies shown in Figure 5.1. PSO algorithm is employed in MATLAB® to find the optimum topology/mapping pair that achieves minimum value of Φ under the following constraints:

Table 5.1: Node descriptions of Figure 5.3.

Node	H.263 MP3 decoder
0	Motion estimation (ME),
1	Discrete cosine transform unit
2	Floating Point (FP) arithmetic unit
3	Inverse Discrete cosine transform unit
4	Motion compensation (MC)
5	Variable Length Decoding unit
6	Memory
7	Bit resolution 1
8	Bit resolution 2
9	Inverse modified discrete cosine transform
10	SUM
11	Buffer

$$
\begin{aligned}
0.40 &\leq V_{sw} \leq 0.75 \\
0.05 &\leq \sigma \leq 0.08 \\
P_e &\geq 0.5 \\
\alpha_a &= \alpha_b = \alpha_c = 1
\end{aligned}
$$

The Matlab ® PSO tool box [123] is set to the following configuration:

1. Number of inputs to the function (dimension of problem) = 6 (Traffic rate, topology/mapping, voltage swing, noise standard deviation, probability of edge failure],

2. Maximum particle velocity for voltage swing, noise standard and probability of edge failure= [0.001 .0001 .0000001],

3. Matrix of ranges for each input variable: as defined in the design constraints mentioned above,

4. Maximum number of iterations 20,000,

5. Population size 9 (group of regular topologies), and

6. Initial positions all random.

Under these conditions, the optimum point found was

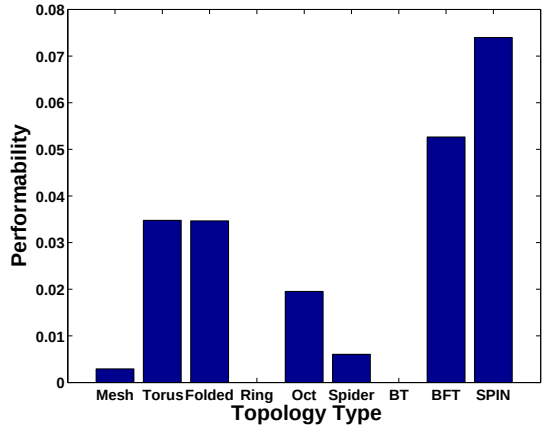
$$x_{out} = [V_{swf}\ \sigma_f\ P_{ef}] = [0.51\ 0.075\ 0.53]$$

and the results generated from the tool are summarized in Table 5.2. To illustrate

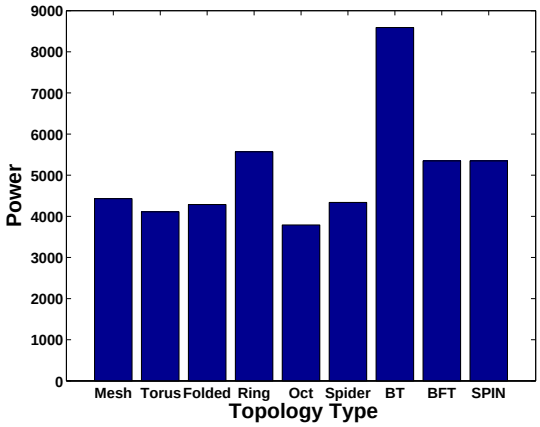
Table 5.2: PSO results.

Maximum performability value	0.3683	Topology type	SPIN
Minimum power value	3.7910e+003	Topology type	Octagon
Minimum delay value	31.5570	Topology type	Octagon
Minimum fitness value	3.7779e+005	Topology type	Torus

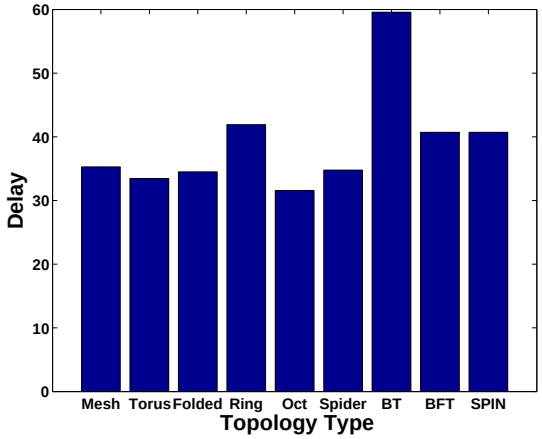
these results further more, a comparison between the network topologies is carried out and the results are shown in Figure 5.4. For each individual design parameter (power, performability, and delay), we compared the optimum values in each case. This comparison is extended to the fitness function as well. Due to the application traffic figure and based on the design constraints that were considered here, the optimum topology that achieves maximum performability for the MP3 encoder, based on individual consideration, was the SPIN topology. As shown in Figure 3.1, these results are consistent with the analysis discusses in Chapter 3 because the probability of edge failure in our case study was very high $P_e \geq 0.5$. On the other hand, Octagon was the optimum choice for both power and delay design parameters, which is also in line with the experimental results obtained in Chapter 2 and Chapter 4. In our case study we aim at minimizing the power consumption and delay and maximizing the performability. To address this trade off, the PSO algorithm found that a Tours



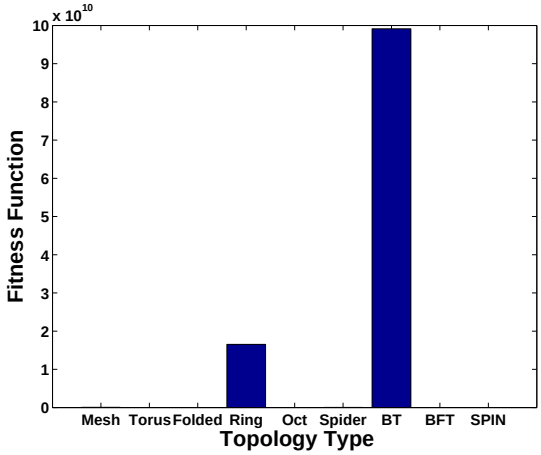
(a) Network performability of MP3 encoder application when mapped on the nine regular topologies in Figure 3.1



(b) Network power of MP3 encoder application when mapped on the nine regular topologies in Figure 3.1



(c) Network delay of MP3 encoder application when mapped on the nine regular topologies in Figure 3.1



(d) Optimum fitness function of MP3 encoder application when mapped on the nine regular topologies in Figure 3.1

Figure 5.4: Comparison between network topologies optimum design values for MP3 encoder application.

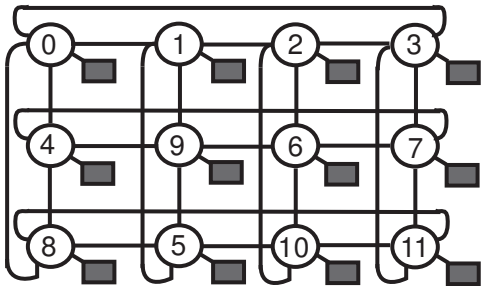


Figure 5.5: Mapping of H.263 MP3 decoder on Torus topology.

topology with the mapping shown in Figure 5.5 is the optimum choice for the MP3 encoder. These results were obtained under equal weight assumption. If any of the weights of the three individual design parameters is changed, the optimum value of the fitness function will change accordingly. Depending on the weight factor, the final topology might change if the weight factor causes one parameter to be dominant over the other two. To verify the efficiency of these results, Table. 5.3 shows the improvement ratio for each single design parameter if another topology (such as BT, which is taken as an example because it is the worst case scenario) was chosen for the MP3 encoder under the same experimental setups.

Table 5.3: Improvement ratios: A comparison between Torus and BT for different design parameters.

Design parameter	Tours	BT	Improvement ratio
Power	$4.1103e+003$	$8.5904e+003$	52%
Delay	33.4545	59.5653	43%
Performability	0.3640	$4.9985e-004$	727%

5.4 Chapter Summary

In this chapter, we presented a practical example that shows how to consider different design parameters simultaneously when designing NoC architecture for a given

application. A joint consideration of NoC power, performability, and delay is carried out using particle swarm optimization to find the optimum network topology, that achieves minimum delay, maximum performability, and minimum power consumption, for a given NoC application. Experimental results show that the selected topology, torus, outperforms the worst case scenario, the mapping of BT, by 52%, 43%, 727% in terms of NoC power, delay, and performability, respectively.

This work has been published in the proceedings of the International Conference of Circuits and Systems (ISCAS'10) [124].

Chapter 6

Contributions and Future Work

6.1 Summary

This section summarizes the research work presented in the dissertation.

In the first part of the research work, presented in Chapter 2, circuit and system analysis for global interconnection links and routers were performed to model the power consumption of NoC-based designs. Based on this analysis, a new methodology to acquire the optimum architecture that achieves the lowest power consumption for a given application, in comparison with standard topologies, is presented. Experimental results show a promising improvement in power consumption (8.55%), average number of hops (10.80%), and number of global links (56.25%) compared to the best known related works.

In the second part of the research work presented in Chapter 3, we proposed a novel topology-based performability model for NoC-based systems. The model is used to evaluate the performability of NoC-based systems at early design phases. A comparative study of nine commonly used network architectures is performed using the proposed model. Based on this model, a new methodology is proposed to improve the performability of a given application at early design phases. In this methodology,

a joint consideration of six design parameters (network topology, target application traffic distribution, mapping of processing elements, noise power, voltage swing, and probability of edge failure) is carried out simultaneously. Experimental results show that performability could have been degraded by 9.2%, for the VOPD application compared to the value achieved by the proposed methodology, if inappropriate parameters were chosen.

In the third part of the research work, presented in Chapter 4, We used Markov chain analysis to derive 2-D model for M/D/1/B queue. We showed how this model can be used to analyze the impact of the packet arrival rate, number of ports, and queue size on the router performance. That 2-D model is then extended to a topology-based average delay model for NoC-based systems. We showed how the NoC delay model could be used to improve the delay of a given NoC application at early design phases.

In the fourth part of the research work, presented in Chapter 5, a joint consideration of NoC power, performability, and delay is carried out simultaneously. We used the particle swarm optimization (PSO) technique to find the optimum network topology, that achieves minimum delay, maximum performability, and minimum power consumption, for a given NoC application. As a proof of concept, H.263 MP3 decoder is taken as a case study in that chapter. Experimental results show that the selected topology, torus, outperform the worst case scenario, the mapping of BT, by 52%, 43%, 727% in terms of NoC power, delay, and performability, respectively.

6.2 Contributions

The major contributions of this research work can be summarized as follow.

6.2.1 Contribution 1

We presented a new model for the power consumption of NoC routers and global interconnection links at different levels of abstractions. Using this model, a new methodology is developed to reduce the total power consumption of an application-specific NoC-based system by selecting the optimal network topology that matches its traffic characteristics. This work has been published in part in the proceedings of the International Conference of Circuits and Systems (ISCAS'08) [63] and in full at Elsevier Journal of Microprocessors and Microsystems [64].

6.2.2 Contribution 2

A comparative study of the impact of network topology on system performability using graph-theoretic concepts is performed. Based on this study, a new model for system performability is developed. This model is then used to develop a new methodology to improve the performability of a given NoC application at early design phases taking into consideration the possible changes in voltage swing, noise power and probability of edge failure. This work has been published in part in the the proceedings of the IEEE International Conference on Design and Technology of Integrated Systems in Nanoscale Era (DTIS'09) [87] and in full at the International Journal of Circuit Theory and Applications, Wiley [88].

6.2.3 Contribution 3

A topology-based average delay model for NoC-based systems has been developed using a new 2-D Markovian M/D/1/B queue model for NoC router. We showed how this model could be used to minimize the overall network delay at system level. This work has been published in part in the Proceedings of the 6th International Workshop on System-on-Chip for Real-Time Applications (IWSOC'06) [111], Lecture

Notes In Computer Science) [112], and in full at the Journal of Circuits, Systems, and Computers [113].

6.2.4 Contribution 4

Optimizing NoC power, performability, and delay simultaneously using Particle Swarm Optimization (PSO) to find the most efficient network topology for a given application. This work has been published in the proceedings of the International Conference of Circuits and Systems (ISCAS'10) [124].

6.3 Future Work

NoC still have many open research problems in application characterization, communication paradigm, and communication infrastructure [125]. We plan to extended this work along the wireless research directions. As technology scales down, more complex networks are expected to be integrated on a single chip, which brings great challenges to the current multi-core systems architectures. Moreover, new applications, such as complex body sensor networks (CBSN), that require heterogeneous network design which employs both wired and wireless networks, have been introduced. We plan to analyze the complexity of designing one NoC platform that supports both wired and wireless connectively, which makes NoC internal resources partially transparent and accessible from outside the chip through its wireless connection. Using NoC-based chips in nodes' implementation is a promising approach in Wireless Sensor Networks (WSN), especially for health applications.

Inspired by wireless computer networks, researchers started exploring the feasibility of WNoCs. The work related to WNoCs in the literature could be classified into two main categories:

1. Wireless networks on a single chip.

2. Wireless communication for NoC-based chips.

The research work of the first category implements several networks on the same chip and connects these networks through micro-antennas. This work is carried out to address such a case when IPs cannot be integrated on a single die due to the interconnection complexity. Although the first impression of such a solution might sound impractical, chip-based wireless communication techniques could be employed using recently-introduced technologies, such as Radio-on-Chip (RoC) [126]. Using RoC technology could lead to a significant improvement in the bandwidth utilization and elimination of the cross-talk noise [126]. Research in this area shows also a promising improvement in the network performance if the wired NoC communication system is replaced by WNoC. In [127], A. Ganguly et al. show that WNoC outperforms its wireline counterpart with respect to network throughput and latency. However, this revolutionary on-chip communication scheme encounters many challenges when it comes to physical implementation.

The research work of the second category discusses applications that employ multiple nodes, such as smart sensor networks. Each one of these nodes is supposed to be implemented as a NoC-based chip that has an internal micro-antenna. Then, these NoC-based chips are programmed to communicate through a wireless network. This approach is presented in [128] as a promising solution for biomedical health care and space exploration sensor-networks. I. Al Khatib et al. in [128] present an NoC platform that supports both wired and wireless connectively, which makes NoC internal resources partially transparent and accessible from outside the chip through its wireless connection. Using NoC-based chips in nodes' implementation is a promising approach in WSNs especially for health applications such as monitoring and detecting Seizures, which are caused by abnormal electrical discharges in the brain. Continuous patient monitoring at homes is becoming an urgent demand

due to the significant increase in hospital costs. However, the current detection techniques require large number of nodes to be connected to scalp electrodes, which are placed on the head of the patient. This design consume a considerable amount of energy, which is a problem for a battery-powered device [129]. Therefore, researchers started designing networks that employ more complex nodes, with more IPs on a single chip. According to Amdahl's law [130], denser multi-core chips increase the performance of the single large core. Since computational-processing energy is way lower than transmitting energy, this approach improves the overall performance by reducing the amount of traffic being transmitted to the base station. In [131], M. Wang et al. showed that distributed processing in WSNs can help reduce energy consumption, make efficient use of network bandwidth, and improve system response time. Although this technique improves the network scalability, there are many research challenges related to the network communication and control protocols. In our future work, we plan to address the following challenges.

6.3.1 Research Challenges in Wireless NoCs

In WNoC-based systems, routers are classified into two categories: central node routers and distributed routers. Figure 6.1 shows an example of multi-core body sensor network (MBSN) that employs WNoCs. Inside multi-core nodes (dotted circles): distributed routers are represented by white circles whereas central node routers are represented by dark circles and processing elements are represented by dark squares. As shown in Figure 6.1, central node routers connect different networks whereas distributed routers route local traffic within the same network. WNoCs are proposed as a concept to connect only central node routers. These routers will communicate through on-chip antennas, which could be implemented on the same chip.

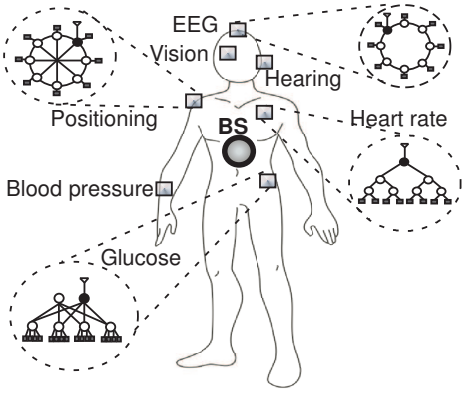


Figure 6.1: An example of MBSN that employs WNoCs. Inside multi-core nodes (dotted circles): distributed routers are represented by white circles whereas central node routers are represented by dark circles and processing elements are represented by dark squares.

Replacing wired links between central node routers with wireless interconnection is a vital approach for the following reasons. First, it helps integrating more IPs on a single chip, and hence, scaling down more end-products. Second, it benefits from the development of advanced micro-antenna technology and applies it to a commercial application to solve a fabrication complexity problem. Third, having WNoC-based system will allow new features to be embedded, such as wireless firmware upgrading, testing, and monitoring, which improve the end-user ability to debug and develop his end-product without cables.

However, WNoCs have several challenges such as the micro-antenna design and fabrication, the security threats due to wireless communication, and the power consumption of such a system. Low power micro-antennas have been analyzed in [132] for SoC applications. An adaptive antenna design is proposed in [132] to have all the communication and control circuitry on a single silicon substrate, which eases the fabrication process. Therefore, integrating these types of antennas with NoC-based systems is a feasible approach. Security threat is another important factor since

wireless communication opens the door for various attacks. One of the solutions to address security threat issues is to employ an efficient encryption technique to secure packets transmission. The power needed to feed all overhead wireless circuits is a significant design parameter. The amount of power consumption could be reduced through reducing the transmitted data, which could be achieved by employing efficient compression schemes.

The implementation of complex encryption and compression algorithms requires more data processing. Nevertheless, the cost of processing a byte is way less than the cost of transmitting one [133]. Also, employing multi-core systems allows the desired performance to be achieved at lower power level compared to a single-core system [134]. Therefore, employing multi-core processing for applications that operate under severe energy budget, such as BSN, becomes even more promising.

We plan to address all these challenges in our future work.

Bibliography

[1] U. Y. Ogras and R. Marculescu, "Application-specific network-on-chip architecture customization via long-range link insertion," in *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design (ICCAD'05)*, San Jose, CA, USA, Nov. 10–14, 2005, pp. 246–253.

[2] P. Pande, C. Grecu, A. Ivanov, R. Saleh, and G. De Micheli, "Design, synthesis, and test of Networks on Chips," *IEEE Design and Test of Computers*, vol. 22, no. 5, pp. 404–413, Sept. 2005.

[3] C. Seiculescu, S. Murali, L. Benini, and G. De Micheli, "Comparative analysis of NoCs for two-dimensional versus three-dimensional SoCs supporting multiple voltage and frequency islands," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 57, no. 5, pp. 364 –368, May 2010.

[4] L. Benini and G. D. Micheli, "Networks on chips: A new SoC paradigm," *IEEE Computer Magazine*, vol. 35, no. 1, pp. 70–78, Jan. 2002.

[5] S. Tota, M. Casu, M. Roch, L. Macchiarulo, and M. Zamboni, "A case study for NoC-based homogeneous MPSoC architectures," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 17, no. 3, pp. 384 –388, March 2009.

[6] T. Bjerregaard and K. mahadevan, "A survey of research and practices of Network-on-chip," *ACM Computing Surveys*, vol. 38, pp. 1–51, Mar. 2006.

[7] C. Zeferino, M. Kreutz, and A. Susin, "RASoC: A router soft-core for Networks-on-chip," in *Proceedings of the Design, Automation and Test in Europe Conference and Exhibition (DATE'04)*, Paris, France, Feb. 16–20, 2004, pp. 198–203.

[8] Open Core Protocol (OCP) Specification, Release 2.0. Technical Document. [Online]. Available: http://www.ocpip.org. Last accessed: December 2010.

[9] A. Haverinen, M. Leclercq, N. Weyrich, and D. Wingard, "Systemc based SoC communication modeling for the OCP protocol." [Online]. Available: http://www.ocpip.org/data/systemc.pdf. Last accessed: December 2010.

[10] A. Bunker and G. Gopalakrishnan. Formal specification of the virtual component interface standard in the unified modeling language. Technical Report, University of Utah Salt Lake City, UT 84112 USA, June, 2001. [Online]. Available: http://www.cs.utah.edu/research/techreports/2001/pdf/UUCS-01-007.pdf. Last accessed: December 2010.

[11] A. Adriahantenaina, H. Charlery, A. Greiner, L. Mortiez, and C. Zeferino, "SPIN: A scalable, packet switched, on-chip micro-network," in *Proceedings of Design, Automation and Test in Europe Conference and Exhibition (DATE'03)*, Munich, Germany, Mar. 3–7, 2003, pp. 70–73.

[12] AMBA AXI Protocol Specification, Release 1.0. Technical Document. [Online]. Available: http://www.arm.com. Last accessed: December 2010.

[13] A. Radulescu, J. Dielissen, K. Goossens, E. Rijpkema, and P. Wielage, "An efficient on-chip network interface offering guaranteed services, shared-memory

abstraction, and flexible network configuration," in *Proceedings of the Design, Automation and Test in Europe Conference and Exhibition (DATE'04)*, Paris, France, Feb. 16–20, 2004, pp. 878–883.

[14] ARM documentation set for AMBA protocol specifications and design tools. ARM InfoCenter. [Online]. Available: http://infocenter.arm.com. Last accessed: December 2010.

[15] V. Soteriou, R. Ramanujam, B. Lin, and L.-S. Peh, "A high-throughput distributed shared-buffer NoC router," *Computer Architecture Letters*, vol. 8, no. 1, pp. 21–24, January 2009.

[16] N. Kapre, N. Mehta, M. deLorimier, R. Rubin, H. Barnor, M. J. Wilson, M. Wrighton, and A. DeHon, "Packet switched vs. time multiplexed FPGA overlay networks," in *Proceedings of the IEEE Symposium on Field-Programmable Custom Computing Machines*, Napa, CA, USA, Apr. 24–26, 2006, pp. 205–216.

[17] C. Hilton and B. Nelson, "PNoC: A flexible circuit-switched NoC for FPGA-based systems," *IEE Proceedings of Computer and Digital Techniques*, vol. 153, no. 3, pp. 181–188, May 2006.

[18] F. Gebali, *Analysis of Computer and Communication Networks*, 1st ed. Berlin, Germany: Springer-Verlag, 2009.

[19] M. Hluchyj and M. Karol, "Queueing in high-performance packet switching," *IEEE Journal on Selected Areas in Communications*, vol. 6, no. 9, pp. 1587–1597, Dec. 1988.

[20] U. Y. Ogras, J. Hu, and R. Marculescu, "Key research problems in NoC design: A holistic perspective," in *Proceedings of the 3rd IEEE/ACM/IFIP*

International Conference on Hardware/Software Codesign and System Synthesis (CODES+ISSS'05), Jersey City, NJ, USA, Sept. 19–21, 2005, pp. 69–74.

[21] A. Ivanov and G. De Micheli, "Guest editors' introduction: The Network-on-Chip paradigm in practice and research," *IEEE Design and Test of Computers*, vol. 22, no. 5, pp. 399–403, Sept. 2005.

[22] B. H. Meyer, J. J. Pieper, J. M. Paul, J. E. Nelson, S. M. Pieper, and A. G. Rowe, "Power-performance simulation and design strategies for single-chip heterogeneous multiprocessors," *IEEE Transactions on Computers*, vol. 54, no. 6, pp. 684–697, Jun. 2005.

[23] R. Bondade and D. Ma, "Self-reconfigurable channel data buffering scheme and circuit design for adaptive flow control in power-efficient Network-on-Chips," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 57, no. 11, pp. 2890–2903, Nov. 2010.

[24] N. Banerjee, P. Vellanki, and K. S. Chatha, "A power and performance model for network-on-chip architectures," in *Proceedings of the Design, Automation and Test in Europe (DATE'04)*, Paris, France, Feb. 16–20, 2004, pp. 21 250–21 256.

[25] M. Arjomand and H. Sarbazi-Azad, "Power-performance analysis of Networks-on-Chip with arbitrary buffer allocation schemes," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 29, no. 10, pp. 1558 –1571, Oct. 2010.

[26] T. Simunic, S. Boyd, and P. Glynn, "Managing power consumption in networks on chips," *IEEE Transactions on Very Large Scale Integration (VLSI)*, vol. 12, no. 1, pp. 96–107, Jan. 2004.

[27] S. Bhat, "Energy models for networks-on-chip components," Master's thesis, Technische University Eindhoven, Eindhoven, Netherlands, Dec. 2005.

[28] Z. Fang, E. Hallnor, B. Li, M. Leddige, S. E. Lee, D. Dai, and S. Makineni, "Boomerang: Reducing power consumption of response packets in NoCs with minimal performance impact," *Computer Architecture Letters*, vol. 9, no. 2, pp. 49–52, Feb. 2010.

[29] E. Beigne, F. Clermidy, H. Lhermet, S. Miermont, Y. Thonnart, X.-T. Tran, A. Valentian, D. Varreau, P. Vivet, X. Popon, and H. Lebreton, "An asynchronous power aware and adaptive NoC based circuit," *IEEE Journal of Solid-State Circuits*, vol. 44, no. 4, pp. 1167–1177, April 2009.

[30] L. Benini, P. Siegel, and G. D. Micheli, "Saving power by synthesizing gated clocks for sequential circuits," *IEEE Design and Test*, vol. 11, no. 4, pp. 32–41, Oct. 1994.

[31] L.-F. Leung and C.-Y. Tsui, "Energy-aware synthesis of networks-on-chip implemented with voltage islands," in *Proceedings of the 44th ACM/IEEE Design Automation Conference (DAC'07)*, San Francisco, CA, USA, June 4–8, 2007, pp. 128–131.

[32] U. Ogras, R. Marculescu, D. Marculescu, and E. G. Jung, "Design and management of voltage-frequency island partitioned networks-on-chip," *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, vol. 17, no. 3, pp. 330–341, Mar. 2009.

[33] D. Shin and J. Kim, "Power-aware communication optimization for networks-on-chips with voltage scalable links," in *Proceedings of the Interna-*

tional Conference on Hardware/Software Codesign and System Synthesis (CODES+ISSS'2004), Stockholm, Sweden, Sept. 8–10, 2004, pp. 170–175.

[34] D. Schinkel, E. Mensink, E. Klumperink, E. van Tuijl, and B. Nauta, "Low-power, high-speed transceivers for Network-on-Chip communication," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 17, no. 1, pp. 12–21, Jan. 2009.

[35] U. Y. Ogras and R. Marculescu, "Analytical router modeling for networks-on-chip performance analysis," in *Proceedings of Design, Automation and Test in Europe Conference (DATE'07)*, Nice, France, Apr. 16–20, 2007, pp. 1096–1101.

[36] V. Chandra, A. Xu, and H. Schmit, "A low power approach to system level pipelined interconnect design," in *Proceedings of the 2004 international workshop on System level interconnect prediction (SLIP'04)*, Paris, France, Feb. 14–15, 2004, pp. 45–52.

[37] Y. Hu, H. Chen, Y. Zhu, A. Chien, and C.-K. Cheng, "Physical synthesis of energy-efficient networks-on-chip through topology exploration and wire style optimization," in *Proceedings of the IEEE International Conference on Computer Design: VLSI in Computers and Processors (ICCD'05)*, San Jose, CA, USA, Oct. 2–5, 2005, pp. 111–118.

[38] S. Murali and G. De Micheli, "SUNMAP: A tool for automatic topology selection and generation for NoCs," in *Proceedings of the 41st Design Automation Conference (DAC'04)*, San Diego, USA, June 7–11, 2004, pp. 914–919.

[39] U. Ogras and R. Marculescu, "It's a small world after all: NoC performance optimization via long-range link insertion," *IEEE Transactions on Very Large Scale Integration (VLSI)*, vol. 14, no. 7, pp. 693–706, July 2006.

[40] F. Regazzoni and M. Lajolo, "Hardware/software partitioning and interface synthesis in networks on chip," in *Proceedings of IP Based SoC Design Conference*, Grenoble, France, Dec. 7–8, 2005. [Online]. Available: http://www.us.design-reuse.com/articles/article13331.html. Last accessed: December 2010.

[41] S. Kumar, A. Jantsch, M. Millberg, J. Oberg, J.-P. Soininen, M. Forsell, K. Tiensyrja, and A. Hemani, "A Network on Chip architecture and design methodology," in *Proceedings of the IEEE Computer Society Annual Symposium on VLSI (ISVLSI'02)*, Pittsburgh, PA, USA, Apr. 25-26, 2002, pp. 105–112.

[42] W. Dally and B. Towles, "Route packets, not wires: On-chip interconnection networks," in *Proceedings of the 38th Design Automation Conference (DAC'01)*, Las Vegas, NV, USA, June 18–22, 2001, pp. 683–689.

[43] W. Dally and C. Seitz, "The torus routing chip," *Distributed Computing*, vol. 1, no. 4, pp. 187–196, Dec. 1986.

[44] J. Xu, *Topological Structure and Analysis of Interconnection Networks*, 1st ed. Norwell, MA, USA: Kluwer Academic Publishers, 2001.

[45] F. Karim, A. Nguyen, and S. Dey, "An interconnect architecture for networking Systems on Chips," *IEEE Micro*, vol. 22, no. 5, pp. 36–45, Sept. 2002.

[46] M. Coppola, R. Locatelli, G. Maruccia, L. Pieralisi, and A. Scandurra, "Spidergon: A novel on-chip communication network," in *Proceedings of 2004 International Symposium on System-on-Chip (SoC'04)*, Tampere, Finland, Nov. 16–18, 2004, p. 15.

[47] M. Moadeli, A. Shahrabi, W. Vanderbauwhede, and M. Ould-Khaoua, "An analytical performance model for the Spidergon NoC," in *Proceedings of*

the 21st International Conference on Advanced Information Networking and Applications, (AINA'07), Niagara Falls, ON, Canada, May 21–23, 2007, pp. 1014–1021.

[48] P. Pande, C. Grecu, A. Ivanov, and R. Saleh, "Design of a switch for network on chip applications," in *Proceedings of the 2003 International Symposium on Circuits and Systems (ISCAS'03)*, Bangkok, Thailand, May 25–28, 2003, pp. 217–220.

[49] P. Guerrier and A. Greiner, "A generic architecture for on-chip packet-switched interconnections," in *Proceedings of the Design and Test in Europe Conference (DATE'00)*, Paris, France, Mar. 27–30, 2000, pp. 250–256.

[50] K.-C. Chang and T.-F. Chen, "Low-power algorithm for automatic topology generation for application-specific networks on chips," *IET Computers & Digital Techniques*, vol. 2, no. 3, pp. 239–249, May 2008.

[51] A. Bona, V. Zaccaria, and R. Zafalon, "System level power modeling and simulation of high-end industrial network-on-chip," in *Proceedings of the Design, Automation and Test in Europe Conference (DATE'04)*, Paris, France, Feb. 16–20, 2004, pp. 30 318–30 324.

[52] T. T. Ye, G. D. Micheli, and L. Benini, "Analysis of power consumption on switch fabrics in network routers," in *Proceedings of the 39th Design Automation Conference (DAC'02)*, New Orleans, LA, USA, June 10–14, 2002, pp. 524–529.

[53] A. Banerjee, R. Mullins, and S. Moore, "A power and energy exploration of network-on-chip architectures," in *Proceedings of the First International Symposium on Networks-on-Chip (NOCS'07)*, Princeton, NJ, USA, May 7–9, 2007, pp. 163–172.

[54] M. Kim, D. Kim, and G. Sobelman, "Network-on-chip link analysis under power and performance constraints," in *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS'06)*, Island of Kos, Greece, May 21–24, 2006, pp. 4163–4166.

[55] T. Ahonen, D. A. Siguenza-Tortosa, H. Bin, and J. Nurmi, "Topology optimization for application-specific networks-on-chip," in *Proceedings of the 2004 International Workshop on System Level Interconnect Prediction (SLIP'04)*, Paris, France, Feb. 14–15, 2004, pp. 53–60.

[56] G. Agnarsson and R. Greenlaw, *Graph Theory: Modeling, Applications, and Algorithms*. Englewood Cliffs, NJ, USA: Prentice-Hall, 2006.

[57] J. L. Gross and J. Yellen, *Graph theory and its applications*, 2nd ed. Boca Raton, FL, USA: CRC Press, 2005.

[58] Predictive technology model (PTM). Arizona State University. [Online]. Available: http://www.eas.asu.edu/~ptm. Last accessed: December 2010.

[59] H. Bhatnagar, *Advanced ASIC Chip Synthesis Using Synopsys® Design Compiler™ Physical Compiler™ and PrimeTime®*, 2nd ed. Kluwer Academic Publisher, Netherlands, 2002.

[60] B. Hendrickson and R. Leland. Chaco partitioning tool. Sandia National Laboratories, Computer Scince Research Institute. [Online]. Available: http://www.cs.sandia.gov/~bahendr/chaco.html. Last accessed: December 2010.

[61] D. Bertozzi and A. Jalabert, "NoC synthesis flow for customized domain specific multiprocessor systems-on-chip," *IEEE Transactions on Parallel and Distributed Systems*, vol. 16, no. 2, pp. 113–129, Feb. 2005.

[62] B. Hendrickson. Algorithms and software for partitioning graphs. Sandia National Laboratories. [Online]. Available: http://www.sandia.gov/~bahendr/partitioning.html. Last accessed: December 2010.

[63] H. Elmiligi, A. Morgan, M. W. El-Kharashi, and F. Gebali, "Power-Aware Topology Optimization for Networks-on-Chips," in *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS'08)*, Seattle, WA, USA, May 18–21, 2008, pp. 360–363.

[64] H. Elmiligi, A. A. Morgan, M. W. El-Kharashi, and F. Gebali, "Power optimization for application-specific networks-on-chips: A topology-based approach," in *Microprocessors and Microsystems*, vol. 33, no. 5, August 2009, pp. 343–355.

[65] G. D. Micheli and L. Benini, *Networks on Chips: Technology and Tools*. San Francisco, CA, USA: Morgan Kaufmann Publishers, 2006.

[66] G. C. Ádm Rák, Balázs Gergely Soós, "Stochastic bitstream-based CNN and its implementation on FPGA," *Journal of Circuit Theory and Applications, Wiley*, vol. 37, no. 4, pp. 587–612, 2009.

[67] J. F. Meyer, "Models and techniques for evaluating the effectiveness of aircraft computing systems," University of Michigan, Ann Arbor, MI, Semi Annual Status Report NASA-CR-158992 Status Rpt, July 1978.

[68] R. M. Smith, K. S. Trivedi, and A. V. Ramesh, "Performability analysis: Measures, an algorithm, and a case study," *IEEE Transactions on Computers*, vol. 37, no. 4, pp. 406–417, Apr. 1988.

[69] C. Constantinescu, "Impact of deep submicron technology on dependability of VLSI circuits," in *Proceedings of the 2002 International Conference on*

Dependable Systems and Networks (DSN'02), Bethesda, MD, USA, June 23–26, 2002, pp. 205–209.

[70] S. R. M. and V. M., "Emulation-based analysis of soft errors in deep sub-micron circuits," in *Proceedings of the International Conference on Field-Programmable Logic and Applications (FPL'03)*, Lisbon, Portugal, Sept. 1–3, 2003, pp. 616–626.

[71] C. Grecu, L. Anghel, P. P. Pande, A. Ivanov, and R. Saleh, "Essential fault-tolerance metrics for noc infrastructures," in *Proceedings of the 13th IEEE International On-Line Testing Symposium (IOLTS'07)*, Crete, Greece, July 9–11, 2007, pp. 37–42.

[72] P. P. Pande, C. Grecu, A. Ganguly, A. Ivanov, and R. Saleh, *Networks-on-Chips: Theory and Practice*, 1st ed., ser. Embedded Multi-Core Systems. Florida, USA: Publisher: Taylor and Francis Group LLC - CRC Press, March 2009, ch. 7: Test and Fault Tolerance for Networks-on-Chip Infrastructures, pp. 191–222.

[73] R. R. Tamhankar, S. Murali, and G. D. Micheli, "Performance driven reliable link design for networks on chips," in *Proceedings of the 2005 Conference on Asia South Pacific Design Automation (ASP-DAC'05)*, Shanghai, China, Jan. 18–21, 2005, pp. 749–754.

[74] A. Ejlali, B. M. Al-Hashimi, P. Rosinger, S. G. Miremadi, and L. Benini, "Performability/energy trade-off in error-control schemes for on-chip networks," *IEEE Transactions on Very Large Scale Integration (VLSI)*, vol. 18, no. 1, pp. 1–14, 2010.

[75] A. Dalirsani, M. Hosseinabady, and Z. Navabi, "An analytical model for reliability evaluation of NoC architectures," in *Proceedings of the 13th IEEE*

International On-Line Testing Symposium (IOLTS'07), Crete, Greece, July 8–11, 2007, pp. 49–56.

[76] P. P. Pande, C. Grecu, M. Jones, A. Ivanov, and R. Saleh, "Performance evaluation and design trade-offs for network-on-chip interconnect architectures," *IEEE Transactions on Computers*, vol. 54, no. 8, pp. 1025–1040, Aug. 2005.

[77] A. Varma and C. Eaghavendra, *Interconnection Networks for Multiprocessors and Multicomputers Theory and Practice*. Los Alamitos, CA, USA: IEEE Computer Society Press, 1991.

[78] D. Bertozzi, L. Benini, and G. De Micheli, "Error control schemes for on-chip communication links: The energy-reliability tradeoff," *The IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 24, no. 6, pp. 818–831, June 2005.

[79] S. Murali, T. Theocharides, N. Vijaykrishnan, M. J. Irwin, L. Benini, and G. D. Micheli, "Analysis of error recovery schemes for networks on chips," *IEEE Design and Test of Computers*, vol. 22, no. 5, pp. 434–442, Sept. 2005.

[80] A. Ejlali, B. M. Al-Hashimi, P. Rosinger, and S. G. Miremadi, "Joint consideration of fault-tolerance, energy-efficiency and performance in on-chip networks," in *Proceedings of the Conference on Design, Automation and Test in Europe (DATE'07)*, Nice, France, Apr. 16–20, 2007, pp. 1647–1652.

[81] K. Srinivasan, K. Chatha, and G. Konjevod, "Linear-programming-based techniques for synthesis of network-on-chip architectures," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 14, no. 4, pp. 407–420, April 2006.

[82] G. Ascia, V. Catania, A. G. Di Nuovo, M. Palesi, and D. Patti, "Efficient design space exploration for application specific systems-on-a-chip," *Journal of System Architeture*, vol. 53, no. 10, pp. 733–750, 2007.

[83] A. Jantsch and H. Tenhunen, *Networks on Chip*. Norwell, MA, USA: Kluwer Academic Publishers, 2003.

[84] F. Worm, P. Ienne, P. Thiran, and G. D. Micheli, "A robust self-calibrating transmission scheme for on-chip networks," *IEEE Transactions on Very Large Scale Integration (VLSI)*, vol. 13, no. 1, pp. 126–139, Jan. 2005.

[85] C. Svensson, "Optimum voltage swing on on-chip and off-chip interconnect," *IEEE Journal of Solid-State Circuits*, vol. 36, no. 7, pp. 1108–1112, July 2001.

[86] A. Antoniou and W.-S. Lu, *Practical Optimization: Algorithms and Engineering Applications*. Berlin, Germany: Springer-Verlag, 2007.

[87] H. Elmiligi, A. A. Morgan, M. W. El-Kharashi, and F. Gebali, "A reliability-aware design methodology for networks-on-chip applications," in *Proceedings of the 6th IEEE International Conference on Design and Technology of Integrated Systems in Nanoscale Era (DTIS'09)*, Cairo, Egypt, Apr. 6–9, 2009, pp. 107–112.

[88] H. Elmiligi, A. A. Morgan, M. W. El-Kharashi, and F. Gebali, "Improving networks-on-chip performability: A topology-based approach," in *International Journal of Circuit Theory and Applications, Article published online: 29 APR 2010, DOI: 10.1002/cta.662 (In Press).*

[89] V. Pavlidis and E. Friedman, "Interconnect-based design methodologies for three-dimensional integrated circuits," *Proceedings of the IEEE*, vol. 97, no. 1, pp. 123–140, 2009.

[90] G. De Micheli, "An outlook on design technologies for future integrated systems," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 28, no. 6, pp. 777–790, 2009.

[91] M. Palesi, R. Holsmark, S. Kumar, and V. Catania, "Application specific routing algorithms for networks on chip," *IEEE Transactions on Parallel and Distributed Systems*, vol. 20, no. 3, pp. 316–330, 2009.

[92] V. Dumitriu and G. Khan, "Throughput-oriented NoC topology generation and analysis for high performance SoCs," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 17, no. 10, pp. 1433–1446, Oct. 2009.

[93] M. Nickray, M. Dehyadgari, and A. Afzali-kusha, "Power and delay optimization for network on chip," in *Proceedings of the 2005 European Conference on Circuit Theory and Design*, Cork, Ireland, Aug.28–2 2005, pp. 273–276.

[94] A. Chien, "A cost and speed model for k-ary n-cube wormhole routers," *IEEE Transactions on Parallel and Distributed Systems*, vol. 9, no. 2, pp. 29–36, Feb. 1998.

[95] J. Duato and P. Lopez, "Performance evaluation of adaptive routing algorithms for k-ary n-cubes," in *Proceedings of Parallel computer Routing and Communication Workshop (PCRCW)*, Seattle, WA, May 16–18, 1994, pp. 45–59.

[96] L.-S. Peh and W. J. Dally, "A delay model for router microarchitectures," *IEEE Micro*, vol. 21, no. 1, pp. 26–34, Jan. 2001.

[97] F. Jafari, Z. Lu, A. Jantsch, and M. Yaghmaee, "Buffer optimization in Network-on-Chip through flow regulation," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 29, no. 12, pp. 1973 –1986, December 2010.

[98] T. Krishna, A. Kumar, L.-S. Peh, J. Postman, P. Chiang, and M. Erez, "Express virtual channels with capacitively driven global links," *Micro, IEEE*, vol. 29, no. 4, pp. 48 –61, July 2009.

[99] X. Wang, T. Ahonen, and J. Nurmi, "Applying cdma technique to network-on-chip," *IEEE Transactions on Very Large Scale Integrated Systems*, vol. 15, no. 10, pp. 1091–1100, 2007.

[100] E. de Souza Carvalho, N. Calazans, and F. Moraes, "Dynamic task mapping for MPSoCs," *IEEE Design and Test of Computers*, vol. 27, no. 5, pp. 26 –35, Sept. 2010.

[101] K.-L. Tsai, F. Lai, C.-Y. Pan, D.-S. Xiao, H.-J. Tan, and H.-C. Lee, "Design of low latency on-chip communication based on hybrid noc architecture," in *Proceedings of the 8th IEEE International NEWCAS Conference (NEWCAS'10)*, June 2010, pp. 257 –260.

[102] J. Xu, W. Wolf, and W. Zhang, "Double-data-rate, wave-pipelined interconnect for asynchronous nocs," *Micro, IEEE*, vol. 29, no. 3, pp. 20 –30, May 2009.

[103] W. Zhou, Y. Zhang, and Z. Mao, "An application specific NoC mapping for optimized delay," in *Proceedings of the IEEE International Conference on Design and Test of Integrated Systems in Nanoscale Technology (DTIS'06)*, Tunis, Tunisia, Sept. 5–7, 2006, pp. 184–188.

[104] V. Pavlidis and E. Friedman, "Interconnect-based design methodologies for three-dimensional integrated circuits," *Proceedings of the IEEE*, vol. 97, no. 1, pp. 123–140, Jan. 2009.

[105] C. Seiculescu, S. Murali, L. Benini, and G. De Micheli, "Sunfloor 3d: A tool for networks on chip topology synthesis for 3-d systems on chips," *Computer-*

Aided Design of Integrated Circuits and Systems, IEEE Transactions on, vol. 29, no. 12, pp. 1987–2000, Dec. 2010.

[106] V. Pavlidis and E. Friedman, "3-D topologies for networks-on-chip," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 15, no. 10, pp. 1081–1090, Oct. 2007.

[107] L. Tedesco, A. Mello, L. Giacomet, N. Calazans, and F. Moraes, "Application driven traffic modeling for NoCs," in *Proceedings of the 19th Annual Symposium on Integrated Circuits and Systems Design (SBCCI'06)*, Ouro Preto, MG, Brazil, Aug. 28– Sept. 1, 2006, pp. 62–67.

[108] V. Soteriou, H. Wang, and L.-S. Peh, "A statistical traffic model for on-chip interconnection networks," in *Proceedings of the 14th IEEE International Symposium on Modeling, Analysis, and Simulation (MASCOTS'06)*, Monterey, CA, Sept. 11-13, 2006, pp. 104–116.

[109] K. Lahiri, A. Raghunathan, and S. Dey, "Evaluation of the traffic-performance characteristics of system-on-chip communication architectures," in *Proceedings of the 14th International Conference on VLSI Design (VLSID'01)*, Bangalore, India, Jan. 3–7, 2001, pp. 29–35.

[110] E. L. Witzke and S. D. Frese, "Some limitations of adjacency matrices in computer network analysis," *SIGCOMM Comput. Commun. Rev.*, vol. 18, no. 5, pp. 43–47, 1988.

[111] H. Elmiligi, M. W. El-Kharashi, and F. Gebali, "A delay model for networks-on-chip output-queuing router," in *Proceedings of the 6th International Workshop on System-on-Chip for Real-Time Applications (IWSOC'06)*, Cairo, Egypt, Dec. 27–29, 2006, pp. 95–98.

[112] H. Elmiligi, M. W. El-Kharashi, and F. Gebali, "Modeling and implementation of an output queuing router for networks-on-chips," *Lecture Notes In Computer Science*, vol. 4523, pp. 241–248, May 2007.

[113] H. Elmiligi, M. W. El-Kharashi, and F. Gebali, "Queue modeling and implementation for networks-on-chip router," *Journal of Circuits, Systems, and Computers*, vol. 16, no. 6, pp. 981–996, Dec 2007.

[114] J. Kennedy and R. C. Eberhart, "Particle swarm optimization," in *Proceeding of the IEEE International Conference on Neural Networks*, Perth, Australia, 1995, pp. 12–13.

[115] X. Hu. Particle swarm optimization tutorial. Swarm Intelligence. [Online]. Available: http://www.swarmintelligence.org/tutorials.php. Last accessed: December 2010.

[116] J. Li and X. Xiao, "Multi- swarm and multi- best particle swarm optimization algorithm," in *Proceeding of the 7th World Congress on Intelligent Control and Automation (WCICA'08)*, China, June 25–27, 2008, pp. 6281–6286.

[117] G. Leary, K. Srinivasan, K. Mehta, and K. Chatha, "Design of Network-on-Chip architectures with a genetic algorithm-based technique," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 17, no. 5, pp. 674 –687, May 2009.

[118] J. Qi, H. Zhao, J. Wang, and Z. Li, "A new hierarchical genetic algorithm for low-power network on chip design," in *Proceedings of the 2010 International Conference on Intelligent Control and Information Processing (ICICIP'10)*, Aug. 2010, pp. 159–162.

[119] N. Choudhary, M. Gaur, V. Laxmi, and V. Singh, "Genetic algorithm based topology generation for application specific Network-on-Chip," in *Proceedings of 2010 IEEE International Symposium on Circuits and Systems (ISCAS'10)*, June 2010, pp. 3156–3159.

[120] J. Yuan, J. He, and L. Zhong, "GA based congestion aware algorithm for application in NoC," in *Proceedings of the International Conference on Computational Intelligence and Software Engineering (CiSE'09)*, Dec. 2009, pp. 1–4.

[121] F. Moein-darbari, A. Khademzadeh, and G. Gharooni-fard, "Evaluating the performance of a chaos genetic algorithm for solving the network on chip mapping problem," in *Proceedings of the the International Conference on Computational Science and Engineering (CSE'09)*, vol. 2, Aug. 2009, pp. 366–373.

[122] G. Tewolde, D. Hanna, and R. Haskell, "Multi-swarm parallel pso: Hardware implementation," in *Proceeding of the IEEE Swarm Intelligence Symposium (SIS'09)*, Nashville, TN, USA, 2009, pp. 60–66.

[123] B. Birge. Aparticle swarm optimization toolbox. [Online]. Available: http://www.mathworks.com/matlabcentral/fileexchange/7506. Last accessed: December 2010.

[124] H. Elmiligi, A. Morgan, M. W. El-Kharashi, and F. Gebali, "Networks-on-chip topology optimization subject to power, delay, and reliability constraints," in *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS'10)*, Paris, France, 30 May - 2 June, 2010, pp. 2354 – 2357.

[125] R. Marculescu, U. Ogras, L.-S. Peh, N. Jerger, and Y. Hoskote, "Outstanding research problems in NoC design: System, microarchitecture, and circuit perspectives," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 28, no. 1, pp. 3–21, Jan. 2009.

[126] Y. Wang and D. Zhao, "Design and implementation of routing scheme for wireless network-on-chip," in *Proceedings of IEEE International Symposium on Circuits and Systems (ISCAS'07)*, New Orleans, USA, May27–30, 2007, pp. 350–355.

[127] A. Ganguly, K. Chang, P. Pande, B. Belzer, and A. Nojeh, "Performance evaluation of wireless networks on chip architectures," in *Proceedings of International Society for Quality Electronic Design (ISQED'09)*, CA, USA, Mar.16–18, 2009.

[128] I. A. Khatib, A. Jantsch, B. Kayal, R. Nabiev, and S. Jonsson, "Wireless network-on-chips as autonomous systems," in *Proceedings of IEEE INFOCOM 2005*, Miami, Florida, Mar.13–17, 2005.

[129] E. I. Shih, A. H. Shoeb, and J. V. Guttag, "Sensor selection for energy-efficient ambulatory medical monitoring," in *Mobisys '09: Proceedings of the 7th international conference on Mobile systems, applications, and services*, Wroclaw, Poland, 2009, pp. 347–358.

[130] M. D. Hill and M. R. Marty, "Amdahl's law in multicore era," *IEEE Computers Magazine*, vol. 41, no. 7, pp. 33–38, 2008.

[131] M. Wang, J. Cao, B. Chen, Y. Xu, and J. Li, "Distributed processing in wireless sensor networks for structural health monitoring," *Lecture Notes in Computer Science*, vol. 4611, pp. 103–112, 2007.

[132] N. Haridas, A. T. Erdogan, T. Arslan, and M. Begie, "Adaptive micro-antenna on silicon substrate," in *Proceedings of the first NASA/ESA conference on Adaptive Hardware and Systems*, Istanbul, Turkey, June15–18, 2006, pp. 43–50.

[133] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks Journal, Elsevier*, vol. 38, no. 4, pp. 393–422, 2002.

[134] R. Kumar, K. Farkas, N. Jouppi, P. Ranganathan, and D. Tullsen, "A multi-core approach to addressing the energy-complexity problem in microprocessors," in *Proceedings of the Workshop on Complexity-Effective Design (WCED'03)*, San Diego, CA, USA, June 2003, pp. 1–8.

Appendix A

List of Publications

A.1 Books

1. Fayez Gebali, **H. Elmiligi**, and M. W. El-Kharashi (Eds.), Networks-on-Chips: Theory and Practice, Taylor and Francis Group LLC - CRC Press, Florida, USA, March 2009, ISBN: 9781420079784.

A.2 Book Chapters

1. A. A. Morgan, **H. Elmiligi**, M. W. El-Kharashi, and F. Gebali, "Bio-inspired NoC architecture optimization," in *Autonomic Networking-on-Chip: Bio-inspired Specification, Development, and Verification*, P. Cong-Vinh, Ed. CRC Press, 2011.

A.3 Journals

1. **H. Elmiligi**, A. A. Morgan, M. W. El-Kharashi, and F. Gebali,"Improving networks-on-chip performability: A topology-based approach," *International*

Journal of Circuit Theory and Applications Article published online: 29 APR 2010, DOI: 10.1002/cta.662 (In Press).

2. **H. Elmiligi**, A. A. Morgan, M. W. El-Kharashi, and F. Gebali, "Power optimization for application-specific networks-on-chips: A topology-based approach," *Journal of Microprocessors and Microsystems*, vol. 33, no. 5-6, pp. 343-355, Aug. 2009.

3. **H. Elmiligi**, M. W. El-Kharashi and Fayez Gebali, "Queue modeling and implementation for networks-on-chip routers," *Journal of Circuits, Systems, and Computers*, vol. 16, no. 6, pp. 981-996, Dec. 2007.

A.4 Conferences and Workshops

1. **H. Elmiligi**, A. A. Morgan, M. W. El-Kharashi, and F. Gebali, "Networks-on-chip topology optimization subject to power, delay, and reliability constraints," in the *Proceedings of the 2010 IEEE International Symposium on Circuits and Systems (ISCAS'10)*, Paris, France, May 30-June 2, 2010, pp. 2354-2357.

2. **H. Elmiligi**, A. A. Morgan, M. W. El-Kharashi, and F. Gebali, "A delay-aware topology-based design for networks-on-chip applications," in the *Proceedings of the fourth International Design and Test Workshop (IDT'09)*, Nov. 15-17, 2009, pp. 1-5.

3. **H. Elmiligi**, A. A. Morgan, M. W. El-Kharashi, and F. Gebali, "A reliability-aware design methodology for networks-on-chip applications," in the *Proceedings of the fourth IEEE International Conference on Design and Technology of Integrated Systems in Nanoscale Era (DTIS'09)*, Apr. 6-9, 2009, pp. 107-112.

4. **H. Elmiligi**, A. A. Morgan, M. W. El-Kharashi, and F. Gebali, "Power-aware topology optimization for networks-on-chips," in the *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS'08)*, Seattle, WA, USA, May 18-21, 2008, pp. 360-363.

5. **H. Elmiligi**, A. A. Morgan, M. W. El-Kharashi, and F. Gebali, "A topology-based design methodology for networks-on-chip applications," in the *Proceedings of the second International Design and Test Workshop (IDT'07)*, Dec. 16-18, 2007, pp. 61-65.

6. **H. Elmiligi**, A. A. Morgan, M. W. El-Kharashi, and F. Gebali, "Performance analysis of networks-on-chip routers," in the *Proceedings of the 2nd International Design and Test Workshop (IDT'07)*, Dec. 16-18, 2007, pp. 232-263.

7. **H. Elmiligi**, M. Watheq El-Kharashi and Fayez Gebali, "Modeling and Implementation of an Output Queuing Router for Networks-on-Chip," in the *Proceedings of the Third International Conference on Embedded Systems and Software (ICESS'07)*, Daegu, Korea, 14-16 May, 2007, (LNCS) 4523, Springer-Verlag Berlin Heidelberg 2007, pp. 70-77.

8. **H. Elmiligi**, M. Watheq El-Kharashi and Fayez Gebali, "Introducing OperaNP: A reconfigurable NoC-based platform," in the *Proceedings of the Canadian Conference on Electrical and Computer Engineering (CCECE'07)*, Vancouver, Canada. Apr. 22-26, 2007, pp. 940-943.

9. **H. Elmiligi**, M. Watheq El-Kharashi and Fayez Gebali, "A delay model for networks-on-chip output-queuing router," in the *Proceedings of the 6th International Workshop on System-on-Chip for Real-Time Applications (IWSOC'06)*, Dec. 27-29, 2006, pp. 95-98.

10. A. A. Morgan, **H. Elmiligi**, M. W. El-Kharashi, and F. Gebali, "Multi-objective optimization of NoC standard architectures using genetic algorithms," in *Proceedings of the tenth IEEE International Symposium on Signal Processing and Information Technology (ISSPIT'10)*, Dec. 15-18, 2010.

11. A. A. Morgan, **H. Elmiligi**, M. W. El-Kharashi, and F. Gebali, "Multi-objective optimization for networks-on-chip architectures using genetic algorithms," in the *Proceedings of the 2010 IEEE International Symposium on Circuits and Systems (ISCAS'10)*, Paris, France, May 30-June 2, 2010, pp. 3725-3728.

12. A. A. Morgan, **H. Elmiligi**, M. W. El-Kharashi, and F. Gebali, "Area and delay optimization for networks-on-chip architectures using genetic algorithms," in the *Proceedings of the fourth International Design and Test Workshop (IDT'09)*, Riyadh, Saudi Arabia, Nov. 15-17, 2009, pp. 1-6.

13. A. A. Morgan, **H. Elmiligi**, M. W. El-Kharashi, and F. Gebali, "Area-aware topology generation for application-specific networks-on-chip using network partitioning," in the *Proceedings of the 2009 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM'09)*, Victoria, BC, Canada, Aug. 23-26, 2009, pp. 979-984.

14. A. A. Morgan, **H. Elmiligi**, M. W. El-Kharashi, and F. Gebali, "Networks-on- chip topology generation techniques: Area and delay evaluation," in the *Proceedings of the third IEEE International Design and Test Workshop (IDT'08)*, Monastir, Tunisia, Dec. 20-22, 2008, pp. 33-38.

15. A. A. Morgan, **H. Elmiligi**, M. W. El-Kharashi, and F. Gebali, "Application-specific networks-on-chip topology customization using network partitioning,"

in the *Proceedings of the First International Forum on Next-Generation Multicore/Manycore Technologies (IFMT'08)*, Nov. 24-25, 2008, pp. 1-6.

A.5 Application Notes

1. **H. Elmiligi**, "Measuring Power Consumption of RTL Designs Based on Switching Activity Simulations," *CMC Microsystems*, August 2009.