Dynamically Reconfigurable Logical Topologies for Fiber Optic LAN/MANs

A thesis submitted
in partial fulfillment of the requirements
for the award of degree of

DOCTOR OF PHILOSOPHY
in
COMPUTER SCIENCE

By

E. V. RAVI CHANDRA MOHAN REDDY

Department of Computer & Information Sciences
School of Mathematics & Computer/Information Sciences
UNIVERSITY OF HYDERABAD
HYDERABAD - 500 046, INDIA.
September 2001

Department of Computer & Information Sciences
School of Mathematics & Computer/Information Sciences
University of Hyderabad
HYDERABAD - 500 046, INDIA.

This is to certify that I, E.V. Ravi Chandra Mohan Reddy, have carried out the research embodied in the present thesis for the full period prescribed under the Ph.D. ordinances of the University.

I declare, to the best of my knowledge, that, no part of this thesis was earlier submitted for the award of any research Degree of any University.

E.V Ravi Chandra Mohan Reddy

Dr. K. Chandra Sekhar Reddy
Thesis Advisor
Department of Computer &
Information Sciences
University of Hyderabad
Hyderabad, A.P.
INDIA – 500 046

Head
Department of Computer &
Information Sciences
University of Hyderabad
Hyderabad, A.P.
INDIA – 500 046

Dean of the School
Dean
School of Maths & CIS
University of Hyd.
Hyderabad-500 134.

Abstract

The large bandwidth, in the range of tens of Terabits/sec, capacity of optical fiber made it an obvious choice for transmission medium of emerging and future networks. However, as the electronically operated computing nodes can not process information at these rates, network architectures must be designed to allow parallel transmissions among the computing nodes by dividing the bandwidth of optical fiber into multiple channels.

Multichannel multihop architectures use Wavelength Division Multiplexing (WDM) for concurrent transmissions at the (order of Gigabits/sec) speeds of computing nodes. The nodes in these networks communicate using a predefined connectivity pattern called *logical topology*. The LAN/MAN architectures usually employ a regular topology as nodes interconnectivity pattern.

This thesis expands upon the multichannel multihop LAN MAN architectures based on regular topologies and considers the use of reconfiguration as a regular topology design issue for tolerating changes in the network such as node failures and additions. By assuming a regular structure of larger size as a base topology wherein each node in the network occupies a unique position, reconfiguration is defined as changing the links of the neighbors of failed/ added node so that the resultant structure remains connected. We call this reconfiguration as *local perturbations* and the topologies designed using local perturbations as *reconfigurable topologies or perturbed topologies.*

This work studies the applicability of the proposed local perturbations paradigm to traditional regular topologies such as Torus, Hypercube and deBruijn graph and also designed regular logical topologies that treat reconfiguration as a new dimension to the topology design, while addressing other dimensions such as low diameter, less average internode distance and simple routing. The reconfigurable topologies are evaluated for uniform and non-uniform traffic conditions. Finally, LAN/MAN architectures using reconfigurable topologies are proposed.

To The Lord Venkateswara

Acknowledgements

I take this opportunity to thank all people who helped me in many ways in bringing out this thesis.

First and foremost, I would like to express my deep gratitude to my advisor Dr. K Chandra Sekhar Reddy for his incessant support, encouragement, guidance and advice that made this dissertation possible. His emphasis on the quality of research and writing has been extremely valuable in writing this dissertation. I also offer my warmest thanks to his family members for their patience in putting up with my schedules and above all for treating me like a family member.

I thank Prof. Arun Agarwal for his support and encouragement as the Head of the department. 1 thank Dean, School of MC1S for providing me the research environment. I thank Prof. Arun K. Pujari for his affection. 1 also thank Prof. H.K. Mohanty for his encouragement. Thanks are also due to other faculty including Dr. AS. Reddy, Dr. P.N. Girija and Dr. G. Uma for teaching me various courses in Computer Science and for their help since my association with the department as a graduate student. I would like to thank Dr. P.V. Reddy for sharing his computing resources during initial stages. 1 would also like to acknowledge the support of Dr. K.N. Murthy and Dr. Atul Negi while pursuing this mammoth task.

I thank Dr. Kumar Sivarajan, Indian Institute of Sciences, Bangalore, for his critical comments on reconfigurable topologies. My warmest thanks are due to my friend Dr. G. Prem Kumar for his constant support and promptly providing me the literature. Thanks are also due to Jean F. Labourdette, Lucent Labs, U.S.A., for providing me insight into some of his related works through personal discussion.

I thank my research colleagues and close friends Dr. Ravindra Sharma, Dr. O. Nalini Kumari and Dr. Vasudev Varma for their constant encouragement. In addition to these

people, other colleagues Dr. Shinymol Anthony, Dr. Nawabi and Bharadwaj made my life at the Department memorable. None the less the company of my friends in the hostel including DRK, Hari, Thaha, Mahammad, KLN[2], Phani[2], Madhav, Kantha Rao, Hemanth, Anand, Chiranjeevi, Thota, Saidulu, LC and Sandeep.

I thank staff members of Al Lab, Department of CIS, School of MCIS and Computer Center. Special mention goes to the staff of Al Lab for their help at odd times. I also thank library staff at University of Hyderabad and Indian Institute of Sciences

My gratitude to my grand mother and my parents for their infinite patience and implicit faith in my capabilities is boundless and cannot be expressed in sufficient words. I feel that my father who passed away recently is looking at this work from the Heaven. I am extremely grateful to my wife Bhagya for being a driving force whenever I tend to loose interest in completing this work. My thanks to my brothers-in-law Govardhan and Srinivas are limitless for their moral support. 1 thank other members of my family and my wife's family for their love and encouragement. I also thank Dr. Mohan Vemuri and his family members for their affection and help rendered by taking care of my one-year-old son, Nikhil, during the final revision of this work. *Last but not least, my son deserves special mention for withstanding the separation from me during this final version.*

This work was partly supported by CISR through their grant in the form of fellowships, Junior Research Fellow and Senior Research Fellow.

Contents

List of Figures

List of Tables

List of Algorithms

List of Theorems

List of Abbreviations

Abbreviation	Meaning
Mbps	Mega bits per second
Gbps	Giga bits per second.
Tbps	Tera bits per second.
%	Modulus operator. For example, x%y gives the reminder of x divided by y.
	Exclusive-OR operation on two binary numbers.

Chapter

1. Introduction

Networking has become an integral part of computing. Computing applications in all sectors are now dependent upon computer networks to routinely, accurately and rapidly convey data from one to another. Similarly, networking is the heart of audio and video communications. Today, Local Area Networks (LANs) with transfer rates of l0Mbps to 100 Mbps are commonplace. Networks with few hundred Mbps transfer rates are already replacing the existing 10Mbps networks and networks supporting a few Gbps per node are emerging. This development towards higher speeds has been a push and pull between users and suppliers, fueled by the technological developments in optical communication and the opportunities provided by the new applications (multimedia web applications, virtual environments, distributed computing).

This chapter presents the motivation and the need for research on logical topology design for fiber optic Local and Metropolitan Area Networks (LAN/MANs). Section 1.1 gives an introduction to the area of fiber optic local and metropolitan area networks. Section 1.2 discusses various logical topologies proposed for multichannel multihop fiber optic LAN/MANs. Section 1.3 presents the motivation behind this research. Section 1.4 states the problem of designing logical topology by considering reconfiguration as a design issue and section 1.5 presents the objectives of the present work. Last section describes the methodology followed in this thesis to address reconfigurable logical topology design problem.

1.1 Fiber Optic LANs/MANs

The first generation fiber optic LANs/MANs were developed on the principle of the traditional networks wherein all the nodes share the single communication channel. Examples of this generation include Expressnet [Tobagi et. al., 1983], CRMA [Nassehi,1990], DQDB [IEEE Std802.6/D15,1990], and FDDI [1SO9312-2,1989]. The

speeds of these networks are limited to the peak electronic processing speeds (2-5 Gbps) [Kazovsky et. al., 1992]. The dependency of nodes' transmission capacities on the network throughput and hence on number of nodes in these networks is a limiting factor for bandwidth hungry applications listed in CCITT recommendation 1.121 - such as video telephony, high-definition image transfer, high-speed data transfer, multimedia mail, video on demand and high-fidelity audio. To overcome this problem, next generation networks aim to utilize the tremendous bandwidth of optical fibers. This became possible by recent advances in photonic technology to multiplex and demultiplex hundreds of wavelengths on a single optical fiber. A detailed study on these generations of networks is given in [van As, 1994]. In this class of networks, the bandwidth of optical fiber is divided into a number of channels with each channel operating closer to the maximum speeds of the electronic interface units, i.e., of the order of Gbps. By using these channels for parallel transmission among various nodes, each node processes only that part of network traffic available on the channels allocated to it. Thus, the network capacity can be increased up to the bandwidth of optical fiber or beyond [Green, 1993; Sivarajan and Ramaswami, 1994; Mukherjee, 1997]. A brief discussion on multichannel networks is presented in the following section.

1.1.1 Multichannel Fiber Optic Networks

The communication among the nodes, in optical networks with multiple channels realized in a single fiber optic cable, can be established either by switching the channels using tunable transmitters/receivers or by switching the information (packets) across various nodes. Networks based on the former approach are called as *single-hop networks* while those of latter approach are *multihop networks* [Mukherjee, 1992a; Mukherjee, 1992b; Mukherjee, 1997]. In a single-hop network, the transmitted information remains in optical form up to its destination. Since the number of transmitters and receivers associated with each node is much less than the number of channels available in the network due to electronic bottleneck, significant amount of coordination among nodes is required. For a packet transmission to occur, one of the transmitters of the sending node and one of the receivers of the destination node must be tuned to the same wavelength and/or code for the duration of the packet's transmission. Moreover, the transmitters and receivers should be able to tune to different channels quickly, so that packets may be sent

or received in quick succession. In addition, single-hop networks should take into account different propagation delays among different pairs of nodes and dispersion of optical signal (i.e., signals of different wavelength propagate at different speeds).

Conversely, in multihop networks, packets pass through intermediate nodes where they may briefly leave the optical domain in order to be routed, and in some cases they may be buffered. A packet is routed to its destination according to a relatively static connectivity among transmitters and receivers of all nodes, and this static connectivity is not expected to change except when a new connectivity of all *transceivers* (transmitter and receiver pairs) is deemed to be more beneficial. The static connectivity among the transmitters and receivers of nodes is called as *logical topology* or *virtual topology*. As the logical topologies are relatively static, the tuning times of transceivers have little impact on the performance of the network. However, in multihop networks, a significant fraction of system capacity is lost due to data (packet) forwarding [Ganz and Koren,1991]. *Hence, in designing a "good" multihop system, one should address two other important issues.* First, the logical topology chosen must be close to optimality in some sense, e.g., the structure's average (hop) distance between nodes must be small, or the maximum number of hops a packet makes should be less. Second, processing complexity of the nodes also must be small because the high-speed environment allows very little processing time; consequently simple routing algorithms must be employed.

For both single-hop and multihop LANs/MANs, it is important to keep in mind that any design must be not only simple and implementable, but also scalable to large user population.

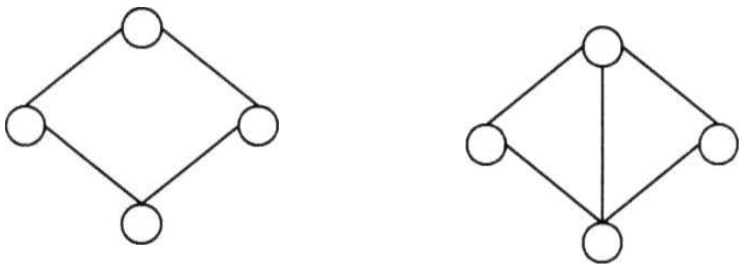
Single-hop networks need wavelength-agile components, as they require dynamic coordination between the nodes for the packet transmission duration. Given that the tuning times of current optical transceivers are significantly long (compared to the packet transmission duration), the challenge is to develop protocols for performing the proper transmission coordination in an efficient fashion. Conversely, in multihop networks, a packet is routed to its destination according to a relatively static connectivity (logical topology) among transmitters and receivers of all nodes. Hence, multihop networks provide guaranteed packet delivery. *The logical topology of multihop networks needs to*

be changed only when there is a change in the network (failure or addition of node) or in traffic. Hence, multihop networks can be developed with existing (relatively) slow tunable transceivers. The following section briefly discusses multichannel multihop fiber optic LANs/MANs.

1.2 Multichannel Multihop Fiber Optic LAN/MANs

Multichannel multihop fiber optic LAN/MANs use broadcast and select architectures, wherein different nodes broadcast messages on different WDM channels, typically, over passive star, linear bus, or tree topologies. *Access nodes in these networks receive messages routed by making use of the logical topology.* Logical topologies are embedded on to the physical topologies such as passive star, linear bus and tree topologies by tuning the transmitters and receivers of the access nodes. *Multichannel multihop networks are classified as regular and irregular topology networks based on the regularity of the logical topology [Mukherjee, 1992b].*

A topology in which each node is connected to exactly *k* nodes is called *k-regular or regular topology.* Otherwise, it is said to be *irregular.* Figure 1.1 gives an example of each class.



a) Regular Topology *b) Irregular Topology*

Figure 1.I Examples of Regular and Irregular topologies

1.2.1 Irregular Topologies

Irregular topologies are, usually, designed based on some optimality criterion. Generally, the optimization function is selected to minimize, the maximum flow among all logical channels. This is done by properly assigning the wavelengths to node pairs based on traffic between them and hence the optimization problem is widely known as flow and wavelength assignment (FWA) problem. The FWA problem is setup as a mixed integer optimization problem [Labourdette, 1991; Labourdette and

Acampora,1991; Zhang and Acampora,1994]. Since the search space of this problem rapidly increases with the size of the network, sub-optimal solutions are proposed in [Labourdette,1991; Labourdette and Acampora,1991; Zhang and Acampora,1994].

Alternatively, the optimization function is selected to minimize the average delay of a packet. The work in [Bannister et. al., 1990] reports that algorithms based on simulated annealing can be employed to solve the delay based optimization problem.

Examples of networks based on irregular topologies include [Labourdette, 1991; Labourdette and Acampora, 1991; Zhang and Acampora,1994; Ofek and Yung, 1995]. Most of these topologies are proposed for wide area networks. But, in [Ofek and Yung, 1995], irregular topologies are also demonstrated for local area networks.

1.2.2 Regular Topologies

Regular topologies are usually designed with an emphasis on the relationship among nodes. Several regular topologies have been extensively studied as interconnection networks for parallel computers [Wittie, 1981; Bhuyan and Agarwal, 1984; Dimopoulos et. al, 1990; Dandamudi and Eager, 1991; Ganesan and Pradhan,1993; Gaughan and Yalamanchili, 1993; Malluhi and Bayoumi, 1994; Szymanski,1995; Latifi and Srimani,1996; Opatrny et.al.,1996; Vadpalli and Srimani,1996]. These topologies became the basis for the study of logical topologies for lightwave networks. This is evident from the topologies proposed for lightwave networks in [Mukherjee, 1992b; Ayadi et. al, 1993; Banerjee et. al, 1994; Brassil et. al, 1994; Feng and Yang, 1994, Sen and Maitra, 1994; Sivarajan and Ramaswami, 1994; Wang and Hung, 1994; Banerjee et. al., 1999], which were studied earlier as interconnection networks. A more recent survey by Banerjee, Jain and Shah on regular topologies for lightwave networks presents various interconnection networks as logical topologies [Banerjee et. al., 1999].

Regular topologies can be further divided as basic and hierarchical topologies. While the basic topologies define simple structures, the hierarchical topologies are constructed by considering such basic topologies as building blocks at different levels. A topology is called as *k-level hierarchical topology* if the topology is constructed by treating nodes of the structure at level $l > 1$ as regular structures such as Hypercube, Ring and deBruijn

graph. [Liu, et. al., 1994a] proposed two-level hierarchical networks for optical communication. As the node degree, in simple (basic) regular topologies such as Hypercube, increases with the size of topology, hierarchical topologies are aimed at constructing the topologies with fixed node degree or aimed at reducing the rate of change of node degree with the size of network. For example, in the Hierarchical Hypercube (HHC) proposed in [Malluhi and Bayoumi,1994], the node degree is (log log *k*), compared to the node degree of (log *k)* for Hypercube of size *k.* Similarly, the maximum degree of a node in Cube Connected Cycles (CCC) is three [Wittie, 1981; Bruck et. al, 1995; Banerjee et. al, 1999]. However, the smaller node degree in hierarchical networks is achieved at cost of the average internode distance and/or diameter. Similarly, some other hierarchical topologies based on de Bruijn and ShuffleNet are proposed in [Liu et. al., 1994a].

Basic topologies are further divided into *point-to-point and bus-based topologies* depending on whether the logical link connects two or more nodes, respectively. Examples of point-to-point regular topologies include Shuffle Exchange Net (SX-Net) topology and its variations used in [Bannister et. al.,1990; Ayadi et. al.,1993; Ramaswami and Sivarajan,1994; Sen and Maitra,1994; Tong and Du,1994; Wang and Hung, 1994]; Manhattan Street Net (MS-Net) used in [Brassil et. al, 1994; Sen and Maitra,1994; Wang and Hung, 1994]; deBruijn graph used in [Feng and Yang, 1994; Sivarajan and Ramaswami, 1994]; Hypercube used in [Ganz et. al, 1994; Tong and Du, 1994], Torus used in [Banerjee et al, 1994]. In bus-based networks, because a logical link connects more than two nodes, some kind of arbitration is required in sharing the logical links in the topology. Usually, time division multiplexing is used as arbitration method for sharing logical links in these bus-based topologies. Bus-based Mesh and Hypercube topologies are proposed in [Liu et. al, 1994b].

Regular topologies, because of their structured node-connectivity pattern, have simplified routing methods. *However, their regularity restricts the set of solutions in addressing the topology optimization problem. Hence, the topology optimization problem in regular topologies is studied as optimal node placement problem*, i.e., mapping the nodes of regular logical topology to the nodes of the physical topology such that some predefined

criteria are satisfied. Generally, for the networks based on broadcast (physical) topologies, there are TV! ways in which TV nodes may be arranged in TV places of the logical topology. *Identifying the optimal placement from these* TV/ *ways of placement is a computationally intensive problem.* The works in [Banerjee and Sarkar, 1992; Banerjee and Mukherjee,1993; Ganz et. al.,1994] and [Yeung and Yum,1998] investigated fast heuristic solutions for constructing near-optimal structures. These solutions are classified as flow-based heuristics and delay-based heuristics, based on whether the optimization criteria are to minimize the maximum flow in a link or to minimize average packet delay.

Though the networks with irregular logical topologies are designed to optimize the network performance, the sub-optimal solutions used for finding optimal topology are usually computationally intensive. *Regular topologies are preferred for LAN MAN solutions for the reasons like they use identical hardware (because each node has same degree of connectivity) and allow simple routing.*

1.3 Motivation

The huge bandwidth potential of single mode optical fiber and electro-optic bottleneck made the researchers to design novel architectures, by invoking some form of concurrency to utilize the entire bandwidth of optical fiber, to satisfy the needs of bandwidth-hungry applications such as virtual environments and distributed computing. Several attempts have been made and a few test-beds were developed by industry giants to study the feasibility of the architectures that are intended to exploit the complete bandwidth of optical fiber [Kazovsky et. al., 1992]. Thus, the demand for high-speed networks and technical feasibility to develop practicable systems are the prime motivations for this research on fiber optic networks.

Out of the various network architectures, regular multihop networks are advantageous not only due to the sufficiency of using fixed or slowly tunable transmitters/receivers, but also due to the simple routing properties defined based on their logical structure. However, the fault-tolerant capabilities of these networks are limited to the node degree. *As the access nodes in multihop networks route the packets to destination nodes, failure of nodes may disconnect the network topology. Also, as the regular structures usually wouldn't allow arbitrary number of nodes, one can think of constructing a regular*

topology of arbitrary size as failure of nodes in a complete regular structure. Hence, reconfiguration of logical topologies - the process of defining connectivity among transceivers of nodes in the network - is essential for regular multihop networks for coping with the changes in the network such as failure or addition of nodes.

Presently, reconfiguration methods are only defined for tolerating traffic changes. These solutions treat reconfiguration as a combinatorial optimization problem and computes new logical connectivity [Labourdette, 1991; Banerjee and Sarkar, 1992; Banerjee and Mukherjee, 1993]. The reconfiguration process then switches the current connectivity diagram to the newly computed connectivity diagram, which may vary to a greater extent. In the process of reconfiguration due to traffic changes, virtual circuits established on the current connection diagram will be torn down, and later re-established on an entirely new connectivity diagram, with different routes. This will cause packet delay, packet loss and packet desequencing, because packets that belong to the same virtual circuit will take different paths to reach destinations [Labourdette, 1991]. *Moreover, the process of reconfiguration diverges if the frequency of reconfiguration is more than the rate at which reconfiguration process is completed* [Chlamtac and Farago, 1994].

As the node changes in the network occur randomly, present reconfiguration methods cannot be applied for tolerating changes in the network. Hence, for tolerating faults and expanding the network, the reconfiguration method should be defined as a natural process in such a way that it affects only that portion of the network where change occurs. Thus the need for reconfiguration to accommodate changes in the network such as addition and failure of a node has motivated us to investigate further.

1.4 Statement of the Problem

One of the major problems in regular multihop fiber optic LAN/MANs is dynamic reconfiguration of the logical topology whenever changes (node failures and/or additions) occur in the network. Present reconfiguration methods basically are proposed for tolerating changes in traffic. They cannot be applied for tolerating changes in the network. This is because node changes in the network occur randomly. *This work considers reconfiguration for tolerating changes in the network as a topology design*

issue rather than addressing separately as a combinatorial optimization problem. This view of reconfiguration is motivated by the following observations.

- *Reconfiguration affects the performance of the networks similar to the way other design issues of logical topology such as diameter, average internode distance and node degree affect.*

  It is known that network performance parameters such as throughput, utilization, average packet delay depends on the characteristics of logical topology such as diameter, average internode distance, number of links and node degree. Similarly, reconfiguration helps in reducing congestion either by finding an optimal connectivity pattern [Ramaswami and Sivarajan,1995] or by optimally placing nodes in a given logical topology [Banerjee and Sarkar, 1992]. The reduction of congestion plays a predominant role on the average packet delay, utilization and throughput of the network.

  Other network performance parameters such as reliability, maintainability, expandability and cost also depend on the logical topology characteristics such as node degree and number of links. But, the ability to reconfigure the logical topology (redefine the connectivity among nodes, i.e., logical topology) plays a crucial role in tolerating faults and accommodating new nodes and hence affects the performance parameters such as reliability, maintainability and expandability.

> *Reconfiguration and other design issues of logical topology are interdependent.*

  High node degree is desired to have high fault tolerance. But, with the ability to reconfigure the network, the fault tolerance capability can be increased even with less node degree.

  Similarly as the reconfiguration defines new connectivity among nodes, it affects diameter and average internode distance.

This thesis addresses the design of logical topologies by proposing the concept of a new paradigm called *local perturbations* [Reddy and Reddy, 2001a] for dynamic reconfiguration of (regular) logical topologies.

Local perturbations paradigm define a methodology for reconfiguring regular logical topologies. According to this paradigm, the network assumes a structure of the largest size for which the network performance is affordable. We call this structure as **base topology.** Each node in the network takes a unique position in the base topology, and is identified with the position in the base topology. Networks with smaller sizes can be constructed by assuming some of the nodes as failed in the base topology. *Whenever a node fails, the neighbors of failed node become interconnected (or reconfigured) such that connectivity is maintained and routing properties of base topology are preserved. Similarly, whenever a node is added to the network, it takes a unique position in the base topology and establishes connections accordingly.* Thus the *reconfiguration process changes the structure of logical topology with each reconfiguration* Topologies designed, using local perturbations paradigm, are referred as **dynamically reconfigurable topologies** or simply, **reconfigurable topologies.**

The reconfiguration process changes the logical topology by just changing the links in the topology where the change (such as node addition and failure) occurs in the network. *Hence, the reconfiguration process is simple and efficient.*

1.5 Objectives

The primary goal of this thesis is to provide a new solution for the reconfiguration of the logical topologies for tolerating node failures and additions in the multichannel multihop fiber optic LAN/MANs.

A key emphasis of this work has been to consider reconfiguration as a topology design issue. A framework has been developed by compiling all the design parameters and by studying their relationship with each other and with the performance of the network. The existing solutions for logical topologies have been investigated under this framework. Other objectives of this work include:

- study and understand optical networks.
- propose a set of alternative regular topologies whose dynamic reconfiguration is transparent to users. Ensure that the dynamically reconfigurable topologies proposed possess simple routing properties.

- study the performance of the proposed topologies under uniform and non-uniform traffic conditions.
- propose fiber optic LAN/MAN architectures.

1.6 Methodology

In order to design dynamically reconfigurable regular logical topologies for tolerating changes in the network, reconfiguration is treated as a design issue. Before developing such logical topologies, we studied how various properties such as diameter, average internode distance, node degree and reconfiguration affect the network performance parameters such as throughput, packet delays, reliability and expandability. We proposed a framework based on these properties, for evaluating logical topologies. Later, we compared and contrasted various logical topologies proposed in the literature - for multihop lightwave networks as well as parallel computers - based on the framework designed. *The evaluation of existing logical topologies showed the need to design logical topologies that consider reconfiguration as a design issue.* Chapter 2 describes more on this.

We then proposed local perturbations paradigm for reconfiguring the logical topologies. The key issues in designing a reconfiguration scheme based on local perturbations are to maintain the connectivity of logical topology and to retain structural properties of base topology. Applicability of local perturbations in traditional topologies is then investigated. A detailed discussion on this is presented in Chapter 3. Reconfigurable topologies designed - Perturbed Torus and Reconfigurable Circulant Graphs - are discussed in this chapter, by applying local perturbations to existing Torus, Circulant Graph topologies as their base topologies.

Chapter 4 evaluates the dynamically reconfigurable topologies, designed in chapter 3, based on the framework presented in chapter 2. The results show that performance of dynamically reconfigurable topologies is comparable to their static counterparts - topologies with base topology structure of same size. As the framework proposed for evaluating logical topologies does not consider non-uniform traffic conditions, greedy

solutions proposed for optimal node placement are used to compare the performance of topologies under non-uniform traffic conditions.

Chapter 5 discusses various implementation issues for fiber optic LAN/MANs based on the dynamically reconfigurable logical topologies proposed in chapter 3. The simulation results of these LAN/MANs are presented. The performance results based on simulation complemented the results obtained in chapter 3 and chapter 4.

Finally, chapter 6 summarizes the main contributions of this research and concludes with directions for further research.

Chapter

2. Regular Logical Topologies

This chapter presents a survey on various regular logical topologies used for fiber optic LAN/MANs, parallel computer architectures and wide area networks and compare some of the well-known logical topologies. Section 2.1 gives an introduction to the concept of regular logical topology. Section 2.2 describes a framework for evaluating regular logical topologies. Section 2.3 provides analysis of various regular logical topologies such as deBruijn graph, Manhattan Street Net, Torus, Hypercube and Shuffle Net, proposed for fiber optic LAN/MANs and parallel computer architectures. It also presents comparison of these well-known logical topologies based on the framework proposed in section 2.2. Finally, section 2.4 discusses how dynamic reconfiguration can be visualized as a topology design issue, and proposes local perturbations paradigm for reconfiguring logical topologies.

2.1 Introduction

The term *physical topology* has been used to refer the connectivity of *access nodes* (terminals and hosts) interconnected by means of physical medium, signal repeaters, transmitters and receivers [Bannister et. al., 1990], while the terms *logical topology and virtual topology* are used, interchangeably, to refer to the coordination among transmitters and receivers of access nodes [Bannister et. al., 1990; Sivarajan and Ramaswami, 1991]. Logical topologies have been referred as *interconnection networks* in parallel computers literature to describe the connectivity among processing elements (PEs) [Feng, 1981, Wittie, 1981; Bhuyan and Agarwal, 1984; Ayadi et.al., 1993; Bruck et. al., 1994].

Logical topologies are classified as regular and irregular topologies depending on whether each node in the topology is connected to same number of nodes or not. Regular topologies are advantageous over irregular topologies due to the richness in their

structure that allow to define simple routing properties and to use identical hardware at each node. In this chapter, we study some popular logical topology families, including Hypercube networks, shuffle exchange networks and hierarchical networks. These topologies vary in terms of their characteristics such as node degree, regularity and diameter.

2.2 A Framework for Evaluating Regular Logical Topologies

In order to provide a clear understanding of the logical topologies, we study the various topological characteristics and how they affect network performance. In describing various topological characteristics, the logical topology is considered as a graph with the nodes in the network as vertices and the logical links of the topology as edges. Hence, the terms — graph, network and logical topology are used interchangeably. Similarly, the terms ~ node and vertex are used interchangeably. Also, the terms ~ link, logical link and edge are used interchangeably.

- **Degree of a node:** According to graph theory, the number of links incident to (from) a node is called the incoming (outgoing) degree of that node. The incoming degree dictates the number of receivers, while the outgoing degree tells the number of transmitters a node requires. *In order to reduce the cost of the network, the degree of a node (both incoming and outgoing) should be small.* To be able to expand the network, it is also desirable that the degree of a node should not be changed with the change in size of the network (i.e., number of nodes). Even if the node degree changes with the size of the network, the rate of change should be minimal.

  However, to provide high reliability, the connectivity (minimum number of edge disjoint paths between all pairs of nodes) should be high. As the connectivity for a connected graph is the minimum node degree, node degree should increase with the network size. Thus reliability contradicts with the other network performance parameters -- cost and expandability. *However, since reliability can also be achieved through reconfiguration (a means for tolerating faults), it is desired to have less node degree.*

Regularity of a topology also helps in using identical hardware at each node. For a *k*-regular topology, the incoming degree and the outgoing degree of a node are equal to *k*. Hence, the network interface of each node in a *k*-regular topology uses *k* transceivers.

Number of Nodes: Number of nodes puts a limit on the expandability of the network. For point-to-point topologies, the number of channels obtained by dividing optical bandwidth limits the number of nodes in the network. *The effective topology should connect more number of nodes with less number of channels.* In other words, the number of nodes that can be accommodated is inversely proportional to node degree. *Alternatively, more number of nodes can be accommodated by sharing a channel in lightwave networks with two or more links of logical topology.* One such solution is proposed by [Chlamtac and Ganz, 1990] wherein WDM channels are shared using TDM. Bus-based topologies are another alternative for accommodating more number of nodes in a network.

Diameter of the Network: The diameter of the network - maximum internode distance - is simply the maximum number of hops (link traversals) that a message may take when it traverses through the shortest path from a source node to a destination node. It places a lower bound on the maximum delay a packet may make while propagating through the network. As the diameter of the network increases, packets need to make more hops and hence, the average queuing delay increases and the network throughput decreases. *It is desirable to have smaller diameter.*

Average Internode Distance: In contrast to the diameter of the network, average internode distance is the expected number of hops a typical message requires to reach its destination. This is a better indicator of average message delay than the network diameter. Unlike the network diameter, *the average internode distance depends on the message routing distribution.* This routing distribution specifies the probability with which different nodes exchange packets and ultimately depends on the communication requirements of the application. Mathematically, average internode distance is described as

$$\bar{d} = \sum_{i=1}^{N} \sum_{\substack{j=1 \\ j \neq i}}^{N} \frac{d(i,j)}{N*(N-1)}$$

where $d(i,j)$ is the distance between nodes i and j, in terms of number of hops; N is the number of nodes

For a symmetric topology, the average internode distance can be written as,

$$\bar{d} = \sum_{i=1}^{D} i * f(i) \qquad \textbf{(2.2)}$$

where D is the diameter; $f(i)$ is the probability of an arbitrary message from a node, makes /hops (i.e., routing distribution).

Different choices of /(/), depending on message routing distributions, lead to different average internode distances. For uniform message routing distribution $f(i)$ is same for all nodes and is given as $N_i/(N-1)$, where TV, is the number of nodes at a distance /'. Uniform routing distribution is more appealing because it makes no assumptions about the type of computation for generating messages; but the same is also a liability from a different perspective. Most message generation distributions exhibit some measure of communication locality. *Hence uniform routing distribution provides what is likely to be an upper bound on the average internode distance.*

It is always desirable to have average internode distance as short as possible. For irregular topologies, low average internode distance may require an unreasonable number of links for each node. This leads to have more number of communication ports.

- **Routing Algorithm:** When a message is to be routed from one node to another, the path it takes is determined by the routing algorithm. The routing algorithm is executed by the originating node and by every other node in the path to the destination. // *is desirable that the routing algorithm is simple and nodes do not requires complete knowledge of the network to route the packets.* It would be convenient by just knowing the source node address and destination node address to

obtain the shortest sequence of nodes the message should traverse to reach its destination.

Routing algorithm affects many of the network performance parameters such as channel utilization, fairness of media access, throughput and ease of implementation.

- **Reconfigurability:** Reconfiguration is the process of (re)defining the logical connectivity (topology) of the nodes. Reconfiguration is required for tolerating node failures as well as for adding nodes into the network. *Thus the ability to reconfigure (reconfigurability) a network not only improves reliability, but also provides expandability of the network. Also, reconfigurability relieves the node degree being used as the parameter for measuring the connectivity reliability.*

The characteristics of a topology described above node degree, number of nodes, diameter, average internode distance, routing algorithm and reconfigurability - constitute the framework for evaluating regular logical topologies [Reddy and Reddy, 2001b]. This framework is used for evaluating the traditional topologies studied as well as the reconfigurable topologies designed in this work.

2.3 Regular Topologies

In this section, we study various regular topologies proposed, by research community, such as Binary Hypercube, Torus, Manhattan Street Network (MS Net), deBruijn Graph, Shuffle Exchange Network (SX Net), Chordal Rings, Star Graph, Cayley Graph, Cube Connected Cycles and Hierarchical Hypercube. These topologies vary in terms of their structure and topological characteristics such as node degree, diameter and average internode distance.

In order to simplify the details of our study, these topologies are discussed as part of their family of topologies that describe general structure of the topologies. This section briefs Hypercube Network, Shuffle-Exchange Network, Linear Congruential Graph, Cayley Graph and Hierarchical Topology families.

2.3.1 Hypercube Networks Family

Many researchers attention was drawn towards Hypercube networks family because of their regular structure provides simple routing and allow easy mapping of various task graphs. This family of networks interconnects $N = m_n * m_{n-1} * \ldots * m_1$ nodes where n is dimensions of the hypercube with m_i nodes in /-th dimension, where i is an integer less than or equal to n. Some of the networks of this family include Generalized Hypercube (GHC) proposed in [Bhuyan and Agarwal, 1984; Gaughan and Yalamanchili, 1993], Hyperrectangular interconnection networks discussed in [Gaughan and Yalamanchili, 1993], Hypermeshes proposed in [Szymanski, 1995], Multidimensional Torus proposed in [Gaughan and Yalamanchili, 1993] and [Banerjee et. al., 1994], Binary Hypercube proposed in [Bhuyan and Agarwal, 1984] and Manhattan Street Network proposed in [Brassil et. al., 1994]. These networks vary in terms of the number of nodes in each dimension and how the nodes in each dimension are interconnected.

We discuss below two architectures of Hypercube networks family namely, Binary Hypercube and Multidimensional Torus.

2.3.1.1 Binary Hypercube

Binary Hypercube was first introduced in 1963 by Squire and Palais. Since then it has been pursued intensively as multiprocessor interconnection architecture by many researchers due to its various attractive features, involving a slow increase in diameter and degree, structural regularity, simple routing and the existence of rich parallel paths between pairs of nodes [Tzeng, 1990]. Several research and commercial hypercube machines have also been built, for example the cosmic cube, NCUBE and Intel iPSC [Das and Banerjee, 1992; Banerjee et. al., 1999]. Hypercube structure was proposed for computer networks by [Bhuyan and Agarwal, 1984; Dowd, 1992; Li and Ganz, 1992] and its topological properties are described in [Saad and Schultz, 1988].

Binary n-dimensional hypercube is an interconnection of 2^n nodes, which may be placed at the corners of an n-dimensional cube with each edge of the cube connecting two nodes in two corners of the edge. The number of nodes in the binary hypercube is given as

$$N_{BH} = 2^n \qquad (2.3)$$

A 3-dimensional binary hypercube is shown in Figure 2.1.

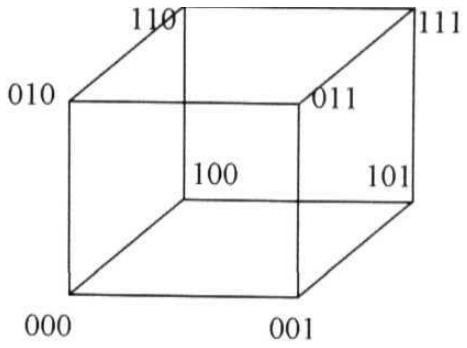


Figure 2.1 Three Dimensional Binary Hypercube

Each node is connected to one node in each dimension. Hence the degree of a node, r_{BH}, is equal to the number of dimensions of the Hypercube, n.

$$r_{BH} - n \tag{2.4}$$

The total number of links, L_{BH} is given by

$$L_{BH} = N_{BH} * n_{=} \tag{2.5}$$

The node number is represented in n-bit binary number and a link represents one bit change in binary codes of the connecting nodes. Hence the distance between any two nodes is the hamming distance of the binary codes that represent node numbers. Since there are n digits in binary representation of the node number, the maximum distance between any two nodes is n, which is nothing but the diameter, D_{BH}, of the topology.

$$D_{BH} = n \tag{2.6}$$

For nodes of distance h, a packet requires to make h hops by changing one dimension in each hop. Therefore, there exist h different edge disjoint shortest paths for nodes of distance h. Since n is the diameter, the maximum number of edge disjoint shortest paths for any pair of nodes is n. The average internode distance for binary hypercube [Mukherjee, 1997] is

$$a_{BH} - \frac{N}{2\ N_{BH}} \tag{2.7}$$

Shortest path routing in binary hypercube is quite simple. A popular shortest path routing algorithm, known as e-cube routing, is explained in [Hamdi, 1995]. According to this

algorithm, the packet travels from node X to node Y on links corresponding to 1 bit in X©Y starting from most significant 1 bit to the least significant 1 bit, where © is the bit wise Exclusive-OR operator. Adaptive routing protocols discussed in [Gaughan and Yalamanchili, 1993] do not bother about the sequence of links to be traversed in order to reach the destination.

The main constraint on the binary hypercube topology is the size of a system has to be a power of two. It leaves a large gap between two consecutive allowable system sizes To overcome this problem, variations to binary hypercube topology have been proposed — incomplete hypercube (IH) [Katseff, 1988, Tzeng, 1990; Tan and Du, 1993], incrementally extensible hypercube (IEH) [Sur and Srimani, 1992] and supercube [Sen and Maitra, 1994]. In addition to the above, some more alternatives have been studied to optimize other topological properties such as diameter; twisted cube [Chedid, 1995] and hyper peterson network [Sen and Maitra, 1994]. As the node degree and number of links increases with size of the network which necessitates additional expensive nodal interface and a larger set of channels, binary hypercube is not regarded as a practical solution for logical topology of multihop lightwave networks [Banerjee et. al., 1999]. Other solutions related to binary hypercube include binary hypercube with unidirectional links [Tan and Du, 1993;Hamdi, 1995].

2.3.1.2 Multidimensional Torus

Multidimensional Torus was proposed as logical topology for multihop lightwave network by [Banerjee et. al., 1994]. Conceptually, the Torus topology can be considered to consist of a number of rings in different dimensions, namely X^{l} rings in the /-th dimension. Lot of research work has been done on the Torus network in the literature, but most of it applies to the two dimensional unidirectional Torus, which is referred to as the Manhattan Street Network (MSN) [Brassil et. al., 1994; Sen and Maitra, 1994]. The Torus topology gives better performance when the number of nodes in each dimension is same. Hence, here we study both unidirectional and bidirectional Torus with *k* nodes in each dimension.

The multidimensional Torus with *k* nodes in each dimension, also called as *k*-ary *n*-cube interconnection network, is a generalization of Binary Hypercube such that *k* nodes are

connected as loop in each dimension. Since each dimension has *k* nodes, the number of nodes in *n*-dimensional network is given as

$$N_{MDT} = k^n \qquad (2.8)$$

A 2-dimensional Torus with 4 nodes in each row and column is shown in Figure 2.2.

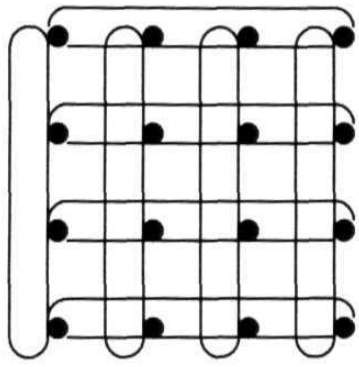


Figure 2.2 Two Dimensional Torus with 16 nodes

Since nodes in each dimension are connected as a loop, each node is connected to two nodes in each dimension. Hence in *n* dimensions, each node is connected to *2*n* nodes. Therefore, the degree of a node, r_{MDT}, is given as

$$r_{MDT} = 2*n \qquad (2.9)$$

Hence, the total number of links is given as

$$L_{MDT} = N_{MDT} *2*n$$

The diameter of the network is given as [Bhuyan and Agarwal, 1984]

$$D_{MDT} = k*n/2 \qquad (2.10)$$

The average internode distance [Banerjee et. al., 1994] is

$$\bar{d}_{MDT} = \left(\frac{k^2}{4}\right)*\frac{n}{k} \qquad (2.11)$$

The unidirectional two-dimensional Torus with links in adjacent rings point in opposite directions is called as Manhattan Street Net (MSNet). In other words, the nodes in MSNet are connected in a grid with adjacent rows and columns traveling in opposite direction, similar to the streets and avenues in Manhattan. Multidimensional extension to

MSNet was studied in [Banerjee et. al., 1994]. MSNet with 16 nodes is shown in Figure 2.3.

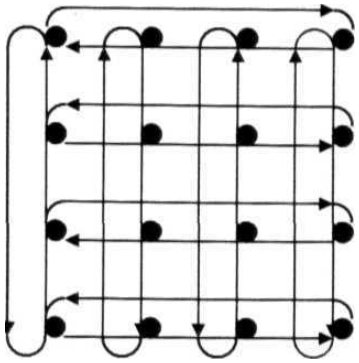


Figure 2.3 MS Net topology with 16 nodes.

Since links are unidirectional, the degree of a node in MSNet is half of that of bidirectional torus. Hence, the node degree, r_{NMSN}, in //-dimensional MSNet is *n*.

$$r_{NMSN} = n \tag{2.12}$$

The total number of links, L_{MSN}, in an N_{MSN} node MS Net is given as

$$L_{NMSN} = N_{NMSN} * n \tag{2.13}$$

The average internode distance of multidimensional MS Net is approximately given by [Banerjee et. al., 1994]

$$\bar{d}_{NMSN} \approx \bar{d}_b + \frac{\frac{1}{2} * p * n + \frac{3}{2} * p * \frac{2^n - 1 - n}{2^n} + \frac{5}{2} * p - 4}{N_{NMSN}} \tag{2.14}$$

where d_b is the average internode distance of bidirectional torus and $p = \frac{kn}{2}$.

The average internode distance of two-dimensional MSNet is given as [Banerjee et. al., 1994]

$$\bar{d}_{MSN} = \frac{k}{2} + 1 - \frac{4}{k^2} \tag{2.15}$$

To understand shortest path routing algorithm, we assume that a node is identified by its position in //-dimensional Euclidean space. According to this algorithm, a packet travels from node X to node Y such that it visits the node Y's coordinate in dimension 1, then

the coordinate in dimension 2 and so on. Another routing algorithm, Deflection routing algorithm is described in [Brassil et. al., 1994]. According to deflection routing, if more than one packet contends for an outgoing link, one of the packets is routed through that link while rest of the packets are deflected to possibly longer routes via other available outgoing links at the node.

2.3.2 Shuffle-Exchange Network Family

The main aim of the networks of this family is to provide $logN$ diameter with fixed node degree. Shuffle Exchange Net proposed in [Acampora, 1987], deBruijn graph proposed in [Sivarajan and Ramaswami, 1994] and GEMNet proposed in [Iness et. al., 1995] fall under this category. In this section we study these three architectures.

2.3.2.1 Shuffle Exchange Net (SX Net)

The Shuffle Net (Shuffle Exchange Net or SX Net) architecture was first proposed for a multichannel multihop lightwave application by Acampora in 1987.

In general, the (p,k)-shuffle net is a directed graph consisting of $N_{SXN} = kp^k$ nodes (k=1,2,...and p=2,3,...) arranged in k columns of p^k nodes each, where/? is the number of incoming and outgoing links of a node in the graph. The nodes in the k-th column are connected to the nodes in the first column so that the interconnection graph can be visualized as being wrapped around a cylinder. The connectivity between the nodes in adjacent columns of SX Net is a p-shuffle, which is analogous to the shuffling of p decks of cards. More precisely, assuming nodes in a column are numbered as 0 through $p^k - 1$, a node i ($0 < i \leq p^k - 1$) in a column is connected to nodes $j, j+1, \ldots, j+p-1$ in the next column where $j - (i * p)\%p^k$ • The node degree, r_{SXN}, is given as

$$r_{SXN} = p \tag{2.16}$$

The diameter of (p,k)-shuffle net is given as,

$$D_{SXN} = 2k - 1 \tag{2.17}$$

The average internode distance of Shuffle Net is given by

$$\bar{d}_{SXN} = \frac{kp^k(p-1)(3k-1) - 2k(p^k-1)}{2(p-1)(kp^k-1)} \qquad (2.18)$$

Figure 2.4 depicts a shuffle net with 8 nodes (k=2, p=2).

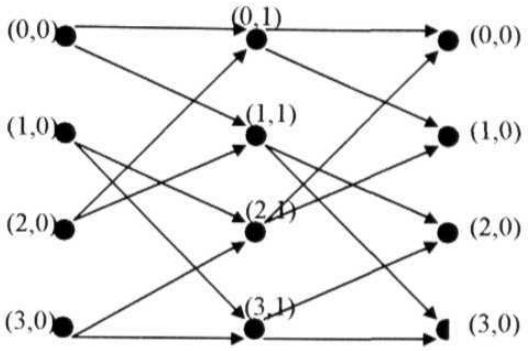


Figure 2.4 A (2,2)-shuffle net with 8 nodes.

The distributed shortest path routing algorithm is described in [Banerjee et. al., 1999]. According to this algorithm, a node at the r-th row and the 6-th column in (p,k)-shuffle net is assigned address (r,c) where $0 < r \leq p^k - 1$ and $0 < c < k$-\. An intermediate node (r_i,c_i) determines the column distance X between itself and the packet's destination (r_d,c_d) which can be given as: $X = (k + c_d - c_i)\%k$ when c_d is not equal to c_i, and $X=k$, otherwise. Out of p neighbors in the next column, it chooses the node at row $(r_i * p + X - 1)\%(p^k - 1)$. Note that the routing decision is made independently at each node without regard for source node of the packet.

In [Maxemchuk, 1989], deflection and store-and-forward routing techniques are investigated for shuffle net. It is observed that deflection reduces congestion, especially in large networks. It is also observed that increasing the number of buffers help in improving throughput. However, the throughput decreases significantly if the arrival rate exceeds the arrival rate for maximum throughput. To shorten the alternate path traveled by misrouted packets in deflection routing, in an alternative solution, all the neighbors of a node in a shuffle net are connected through an auxiliary link to form a loop. The resultant topology, called Augmented Shuffle Net, provides improved performance as it reduces the additional hop distance for misrouted packets [Wang and Hung, 1994]. Other solutions related to shuffle net -- Bilayered ShuffleNet proposed in [Ayadi et.al., 1993] and Extended ShuffleNet proposed in [Camarda and Cocolicchio, 1992] study the use of

additional links to the nodes in the previous column and the use of sharing channels to accommodate more nodes in the network, respectively.

2.3.2.2 ***deBruijn*** *Graph*

deBruijn graph structure was first proposed by de Bruijn in 1949 [Banerjee et. al., 1999] and was adopted as logical topology for optical networks by [Sivarajan and Ramaswami, 1994].

(k,D)-deBruijn graph is the k-regular directed graph with $N_{dBG} = k^D$ nodes where $k > 2$ and $D > 2$. Each node is represented as D-digit number in k-ary number system. A link from node $A = a_1 a_2 \ldots a_D$ to $B = b_1 b_2 \ldots b_D$ exists iff $b_i = a_{i+1}$; $a_i, b_i \in \{0,1,\ldots,k-1\}$, $1 < i < D-1$. In other words, there is a link joining node A with node B if node B can be reached from node A with one left shift and a new input digit at the right end. As the right end digit can take any value between 0 and k-1, each node is connected to k nodes. Thus the degree of a node is given as

$$r_{dBG} = k \qquad (2.19)$$

The total number of links, L_{dBG} is given as

$$L_{dBG} = (k * N_{dBG}) \qquad (2.20)$$

The maximum number of digits to move from a node A to reach any node B is D. Hence the diameter of the network is given as

$$D_{dBG} = D \qquad (2.21)$$

The average internode distance, $\bar{d}_{dBG}$ of deBruijn graph is bounded by the following inequality [Sivarajan and Ramaswami, 1994].

$$D\frac{N}{N-1} - \frac{k}{(k-1)^2} + \frac{D}{(k^D-1)(k-1)} \le \bar{d}_{dBG} \le D\frac{N}{N-1} - \frac{1}{k-1} \qquad (2.22)$$

where $k \ge 2, D \ge 2$.

Shortest path routing in the de Bruijn graph is simple. A link from node A to node B can be represented by $(D+1)$ k-ary digits, the first D of which represent node A and the last D digits represent node B. In a similar fashion, any path of length x can be expressed by

$D+x$ digits. In determining the shortest path from node A = $a_1 a_2 \ldots a_D$ to node B = $b_1 b_2 \ldots b_D$, one needs to consider the last several digits of A and the first several digits of B to obtain a perfect match over the largest possible number of digits. If this match is of x digits, i.e., $b_1 b_2 \ldots b_{D-x} = a_{x+1} a_{x+2} \ldots a_D$, then the x-hop shortest path from node A to node B is given by $a_1 a_2 \ldots a_D b_{D-x+1} b_{D-x+2} \ldots b_D$.

An undesirable characteristic of the de Bruijn graph is that, even if the offered traffic to the network is fully symmetric, the link loadings can be unbalanced. This is due to the inherent asymmetry in the structure, e.g., in the (2,3)-de Bruijn graph shown below in Figure 2.5, the self-loops on nodes 000 and 111 carry no traffic (and hence are wasted) and the link 1000 only carries traffic destined to 000 while link 1001 carries all remaining traffic generated by or forwarded through node 100. This asymmetry in link loads affect throughput [Sivarajan and Ramaswami, 1994].

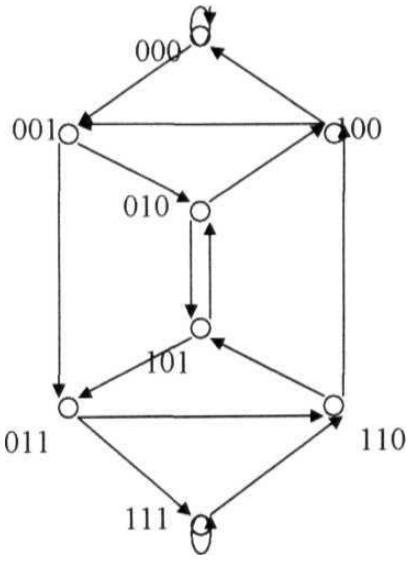


Figure 2.5 A (2,3)-deBruijn graph

Another disadvantage of de Bruijn graph is the restriction on the number of nodes. Since there is a large gap between k^D and k^{D+1}, smooth expansion of the network is a problem as one may face the choice of too little or too many available nodes. However, the requirement of k^D nodes seems to be inherent in D-vector k-ary number system and there is no obvious way to extend de Bruijn graphs for arbitrary number of nodes. To

overcome this problem, in [Du and Hwang, 1988], nodes of de Bruijn graphs are represented by ordinary numbers so that the link set consists of $i \rightarrow (ki + r) \% k^D$, for $0 \le i \le k^D - 1$, $r < k$-\. This extension of de Bruijn graph, called generalized de Bruijn graph, allows arbitrary number of nodes because we can easily replace k^D with some arbitrary number *N*.

2.3.2.3 GEMNet

GEMNet is a generalization of shuffle exchange networks described earlier including the shuffle net and de Bruijn graph.

A (K,M,p) GEMNet interconnects $N_{GEM}=K*M$ nodes arranged in K (>1) columns and $M(\ge p)$ rows, with each node being connected to p nodes in the next column. Each node is identified with its position in the grid (that is row number followed by column number). A node (r,c) positioned at row r and column c is connected to the node $((r * p + i) \% M, (c + 1) \% K)$ where $i<p$. A (2,6,2) GEMNet with 12 nodes is shown in figure below.

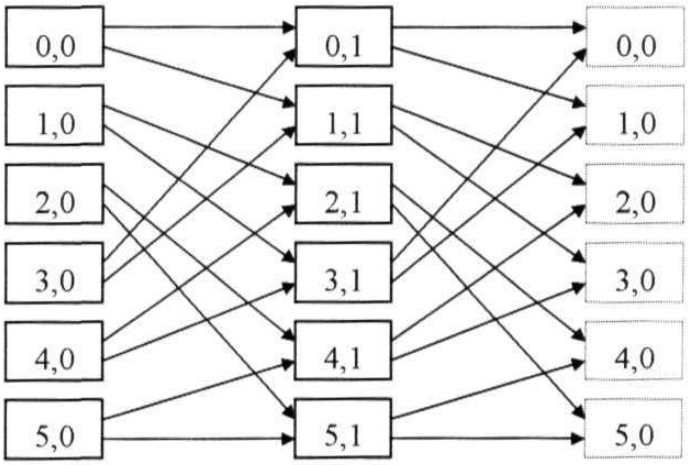


Figure 2.6 A (2,6,2) GEM Net

It may be observed that GEMNet reduces to (p,K) ShuffleNet when $M=p^K$. GEMNet also reduces to a de Bruijn graph of Diameter D when $M=p^D$ and $K=1$. Hence, GEMNet, which is based on a generalized shuffle exchange connectivity pattern, has more flexible structure than that of shuffle net or de Bruijn graph because the number of nodes is not restricted in any form.

The diameter, D_{GEM}, of GEMNet is given by

$$D_{GEM} = \lceil \log_p M \rceil + K - 1 \qquad (2.23)$$

There is no closed form expression for the average hop distance in GEMNet. However, tight upper and lower bounds (in closed form) on the average internode distance are given as follows [Mukherjee, 1997].

$$\bar{d}_{GEM\max} = \frac{MK}{MK-1}\left(D_{GEM} + \frac{1}{2} - \frac{1}{2}K\right) - \frac{K}{MK-1}\left(\frac{p^{D_{GEM}-K+1}-1}{p-1}\right) \qquad (2.24)$$

and

$$\bar{d}_{GEM\min} = \sum_{i=0}^{L} ip^i + \sum_{i=L+1}^{L+K} i\left(M - p^{i\%k}\sum_{j=0}^{\lfloor i/k \rfloor - 1} p^{kj}\right) \qquad (2.25)$$

where, $L = \begin{cases} D_{GEM}-K & if, p^{(D_{GEM}-K)\%K} X \frac{(p^K)^{F+1}-1}{p^K-1} < M, \text{ where } F = \lfloor \frac{D_{GEM}-K}{K} \rfloor. \\ D_{GEM}-K-1 & otherwise \end{cases}$

In general, the lowest average hop distance occurs for GEMNets with K=1. Also, for the GEMNets with $p \geq 3$, in general, the best average hop distance is achieved by a GEMNet with K=1.

The shortest path routing is described in [Mukherjee, 1997]. Let (r_s, c_s) and (r_d, c_d) be the source node and the destination node, respectively. The column distance, 6, is defined as the minimum hop distance in which the source node touches (covers) a node (not necessarily the destination node) in the destination node's column. Then the column distance is given as $\delta = (K + c_d - c_s)\%K$ The hop distance from source node to the destination node is given by the smallest integer h of the form $(\delta + jK)$, j=0,1,2,..., satisfying the following expression:

$$R = \left\lfloor M + r_d - \left(r_s x p^h\right)\%M \right\rfloor \%M < p^h$$

R, called the route code, specifies a shortest route from the source node to the destination when it is expressed as a sequence of h base-p digits. In general, if $R = [\alpha_1 \alpha_2 \ldots \alpha_h]_{basep}$, then the node about to send the packet on /-th hop will route on α_j-th outgoing link. As

the shortest path routing algorithm does not consider link loads, alternative routing schemes that perform better by considering alternate shortest paths are discussed in [Mukherjee, 1997].

2.3.3 Linear Congruential Graphs

Linear Congruential Graphs were formally developed by [Opatrny and Sotteau, 1991] for interconnection networks. The class of networks interconnects N nodes such that node x is connected to $f(x)\%N$, where $f(x) \in F$ is a linear function and F is the set of linear functions, called as generator set, defines node connectivity.

The class of linear Congruential graphs is a very broad family of graphs. By imposing some restrictions on the values of constants of the generators, we can obtain subfamilies of linear Congruential graphs. Some of the subfamilies that belong to this class include Kautz graphs [Kautz, 1969; Panchapakesan and Sengupta, 1995], generalized de Bruijn graphs [Du and Hwang, 1988], Circulant Graphs [Khurshid et. al., 1989; Bruck et. al., 1993] and DCC linear Congruential graphs [Opatrny et. al., 1996]. The following section briefly discusses Circulant Graphs.

2.3.3.1 Circulant Graphs

Circulant graphs were proposed for multichannel optical networks by [Khurshid et. al., 1989] and were studied in [Bruck et. al., 1993] for parallel computers. Circulant graphs are also referred as *Chorda/ Rings.*

A graph is a **Circulant graph** or **Generalized Chordal Ring** (GCR) if its nodes can be labeled with integers and there is a divisor q of N such that node i is connected to node j iff node $(i+q)\%N$ is connected to node $(j+q)\%N$. The link that connects node i to node j through divisor q is called as *chord or jump* of length q. A Chordal Ring (CR) is a special case of a GCR, in which every node has chords of length +1 and -1. Figure 2.7 shows Circulant graph of 8 nodes with chords of length +1, -1, +2 and -2. Note that the jumps of length +1 and -1 are represented as bidirectional links. Similarly, jumps of length +2 and -2 are represented as bidirectional links.

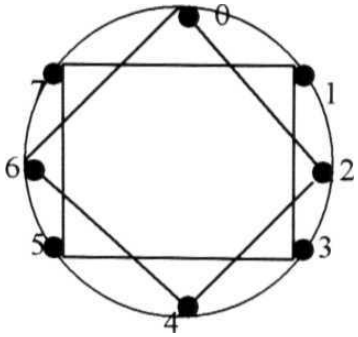


Figure 2.7 An 8 node Circulant Graph with jumps of length \ 1, -1, + 2 and 2.

Different configurations are possible with different choices of jumps. Hence, node degree, diameter and average number of internode distance of a Circulant graph depend on the number of chords, chord lengths and number of nodes. One of the popular configuration of Chordal rings, called daisy-chain, is studied in [Reed and Schwetman, 1983]. Daisy-chain has jumps of length 2, in addition to jumps of length 1. Another configuration logarithmic node degree and diameter, called Hyper-Ring is described in [Khurshid et. al., 1989]. An *N* node Hyper-ring has jumps of length 1, 2, 4, ..., $\lceil log_2^N 1$

Simple routing method can be formulated for Circulant graphs by reducing distance between source and destination. According to this algorithm, called distance-reduction algorithm, the packet travels on a link that reduces the distance between the current node and destination node.

2.3.4 Cayley Graph Family

The Cayley graph model was formally developed by [Akers and Krishnamurthy, 1989] for processor interconnection networks. These networks are based on permutation groups and include a large number of families of graphs, like Star graphs [Day and Thripathi, 1994], Hypercubes, Pancake graphs [Akers and Krishnamurthy, 1989], Borel Cayley graphs [Tang and Arden, 1992] and Cayley graph with constant node degree four [Vadapalli and Srimani, 1996]. An excellent survey of these Cayley graphs can be found in [Lakshmivarahan et. al., 1993]. In the following section, we study Star Graphs and Cayley Graph of Constant Degree Four.

2.3.4.1 Star Graph

A Star Graph S_n of dimension *n* is a Cayley graph with *n*\ nodes, each labeled with a distinct permutation of the set of integers $\{1,2,\ldots,n\}$. The adjacency among nodes is defined based on a set of permutations. G, referred to as generators, where $G = \{g(i,j), = j\ 1,2,\ldots,(i-1),(i+1),\ldots,n\}$. Without loss of generality, it is assumed / = 1. The neighbors of a node are obtained by composing the generators with the label of the node. For example, Star graph S_3 contains six nodes labeled as 123, 231, 312, 132, 213 and 321. The set of generators G={g(l,2),g(l,3)}. Each node of S_3 is connected to two nodes. For example, node 123 is connected to 213 and 321 using g(l,2) and g(l,3) generators.

Since each node is connected to one node using a generator, the node degree of Star graph is given as

$$r_{Star} = |G| = n - 1 \qquad (2.26)$$

Hence, the number of links in Star graph is

$$L_{Star} = (n-1) * n! \qquad (2.27)$$

The diameter of Star graph is given by [Latifi, 19931 and [Latifi and Srimani, 1996]

$$D_{Star} = \lceil 3(n-1)/2 \rceil \qquad (2.28)$$

The shortest path routing between two nodes in the star graph is done according to the following rules:

1. If the leftmost digit of current node is same as the destination, move the packet on link with generator $(1,j)$where *j*-th digit of current node and destination node is not same.
2. If *x* is the leftmost digit, where $x \leq n$, move the packet on link $(1,j)$where /-th digit of destination is x.

2.3.4.2 Cay ley Graph of Constant Degree Four

Cayley Graph of Constant Degree Four was proposed by Vadapalli and Srimani as an interconnection network in [Vadapalli and Srimani, 1996].

Unlike many Cayley graphs proposed in the literature, this graph has constant node degree four. The Degree Four Cayley graph is defined as a graph, G_n, connecting $N_{Cayley4} = n*2^n$ nodes for any integer n, $n \geq 3$; each node is represented by a circular permutation of n symbols in lexicographic order where each symbol may be represented in either uncomplemented or complemented form. Thus, for n distinct symbols, there are exactly n different cyclic permutations of the symbols in lexicographic order and since each symbol can be present in either complemented or uncomplemented form, the number of nodes in the graph is

$$N_{Cayley4} = n * 2^n \tag{2.29}$$

For example, for n=3, the number of vertices in the graph is 24; **abc,** cab, cab are valid nodes while **acb** or bac are not. The following four generators define the edges of the graph.

$$g(a_1 a_2 \ldots a_n) = a_2 a_3 \ldots a_n a_1 \tag{2.30}$$

$$f(a_1 a_2 \ldots a_n) = a_2 a_3 \ldots a_n a_1 \tag{2.31}$$

$$g^{-1}(a_1 a_2 \ldots a_n) = a_n a_1 \ldots a_{n-1} \tag{2.32}$$

$$f^{-1}(a_1 a_2 \ldots a_n) = a_n a_1 \ldots a_{n-1} \tag{2.33}$$

The diameter of Degree Four Cayley Graph, G_n, is given by [Vadapalli and Srimani, 1996]

$$D_{Cayley4} = \left\lfloor \frac{3n}{2} \right\rfloor \tag{2.34}$$

A simple routing scheme is explained in [Vadapalli and Srimani, 1996]. Let $a_1 a_2 \ldots a_n$ be the source node and $b_1 b_2 \ldots b_n$ be the destination node. The following algorithm computes a path from source node to the destination.

1 Compute k, $1 < k < n$, such that $a_k = b_n$, where b_n denote either b_n or $b_{\bar{n}}$.

2. If $k > \lfloor n/2 \rfloor$, then go along successive g^{-1} edges $(n-k)$ times

   else go along successive g edges k times.

3. for $i=1$ to n do

   go to the node $b_{i+1}^{*} b_{i+2}^{*} \ldots b_n^{*} b_1 b_2 \ldots b_i$ by either the g or the f edge.

2.3.5 Hierarchical Topologies

Hierarchical topologies are built by considering regular structures of smaller dimensions as the building blocks (nodes) for the regular structures at higher levels, i.e., each level of hierarchy comprises a network of modules that in turn contain a set of lower-level modules interconnected in some fashion. Thus in a two-level hierarchical network, the N nodes in the system are grouped into K modules of $n_1 = N/K$ nodes each. Each module of n_1 nodes is linked together internally by a level 1 interconnection network. One or more nodes from each module act as interface nodes to the rest of the system. These interface nodes are interconnected by a level 2 network.

Though the structures can be built with any number of levels of hierarchy, most of the networks proposed in the literature have only two levels. Some of the hierarchical networks studied in the literature include Cube Connected Cycles (CCC) proposed in [Wittie, 1981; Bruck et. al., 1995], Hierarchical Hypercube proposed in [Dandamudi and Eager, 1991; Malluhi and Bayoumi, 1994], Cayley Graph Connected Cycles discussed in [Banerjee et. al., 1999] and Hierarchical networks based on deBruijn graphs and Shuffle Net which is proposed in [Ganesan and Pradhan, 1993] and [Liu et. al., 1994a].

Hierarchical networks aim at accommodating large number of nodes with relatively low node degree. Also, hierarchical topologies are capable of exploiting the locality of reference (communication). Following sections discusses two architectures of Hierarchical networks family namely, Cube Connected Cycles and Hierarchical Hypercube.

2.3.5.1 Cube Connected Cycles (CCC)

Cube Connected Cycles was proposed for computer networks by [Wittie, 1981].

A cube connected cycles (CCC) of dimension *D* contains $D*2^D$ nodes arranged in a cycle of D nodes at each of the 2^D vertices of a binary hypercube of D dimensions. Each node is connected to exactly three others by dedicated bidirectional links; two links to neighbors on the same cycle and one link crossing hypercube in one of the D dimensions to the corresponding node in another cycle. The number of nodes and node degree of CCC are expressed as

$$N_{CCC} = D*2^D \tag{2.35}$$

$$r_{CCC} = 3 \tag{2.36}$$

Figure 2.8 shows a CCC with 24 nodes with D=3.

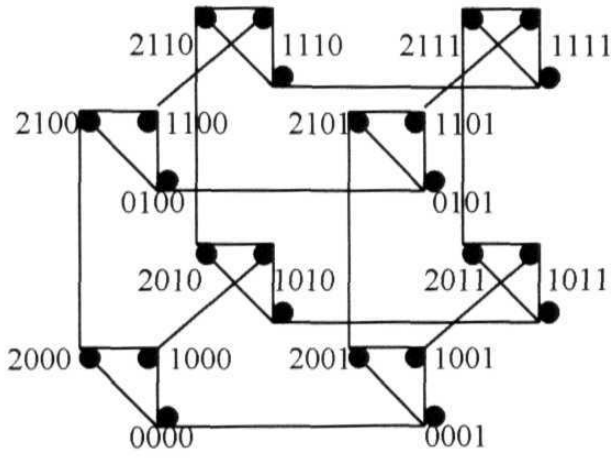


Figure 2.8 A 24 node CCC with D-3.

For analysis purpose, we consider each node being identified by the position in the cycle followed by a coordinate in a 2^D lattice. For example, node $pc_0c_1...c_{D-1}$ positioned at p in a cycle at $c_0c_1...c_{D-1}$ in the 2^D lattice, where $0 \le p \le D-1$ and c_i=0 or 1, for all $0 \le i \le D-1$. Node $pc_0c_1...c_{D}$ is connected to $(p+1)\%Dc_0c_1...c_{D-1}$ and $(p+D-1)\%Dc_0c_1...c_{D-1}$ in the cycle and $pc_0...c_p...c_{D}$, where c_p is the complement of the p^{th} bit in binary representation of the node's position in the 2^D lattice.

The diameter of CCC is given as

$$D_{CCC} = D * \frac{5}{2} - 1 \qquad (2.37)$$

The average internode distance is given by [Wittie, 1981]

$$\bar{d}_{CCC} = D * \frac{7}{4} - 3 + \frac{D+1}{2^{D-1}} \qquad (2.38)$$

An efficient routing algorithm for CCC is similar to that for binary hypercube topology. To explain routing we consider the nodes are identified by the node's position in the cycle followed by the cycle position in the cube. For each source node containing the message, compare binary coordinate c_i of the source to that of the destination where i=p, the cycle position of the source node. If the coordinates differ, use the /-th dimension cross-link to relay the message to a new node with the same index as the source except in the c_i coordinate. If the /-th coordinates are equal, but the message has not yet reached the cycle of the final destination (i.e., not all the binary coordinates match), use a cycle-link to move to the node in the next position within the same cycle (i.e., to the node with index: $(p+1)\%Dc_0c_1\ldots c_{D-1}$). If the message has reached the final cycle, determine whether the destination node is closer in the forward $(p+1)$ or the backward $(p-1)$ direction along the cycle and use a cycle-link to move one position closer to the final destination. Repeat the comparison and use of a cross-link or cycle-link, until the message reaches the destination node.

An improvement over CCC obtained by rearranging the hypercube links is discussed in [Banerjee et. al., 1999]. This network, called HCRNet, significantly lowers average internode distance.

2.3.5.2 Hierarchical Hypercube

Hierarchical Hypercube (HHC) has been proposed as interconnection network in [Dandamudi and Eager, 1991; Malluhi and Bayoumi, 1994].

An N node Hierarchical Hypercube (HHC) with N=2^k, for k=2^m+m, where AW is a positive integer, is a symmetric structure consisting of a father hypercube whose nodes are by themselves hypercubes rather than simple nodes.

The *N* nodes are grouped into clusters of 2^m nodes each, where m is positive integer, and the nodes in each cluster are connected to form an m-cube called the Son-cube or Scube. The set of Scubes constitutes the second level of hierarchy. A father cube, called the Fcube, connects the $2^{k-m} = 2^{2^m}$ Scubes in a hypercube fashion. Edges of Scubes are called internal edges and edges of the Fcube are referred to as external edges. An Scube having 2^m nodes is connected to exactly 2^m external edges with each edge is incident to a node of a Scube.

Figure 2.9 shows two adjacent Scubes in an 11-HHC (an HHC with k=l 1 and m=3).

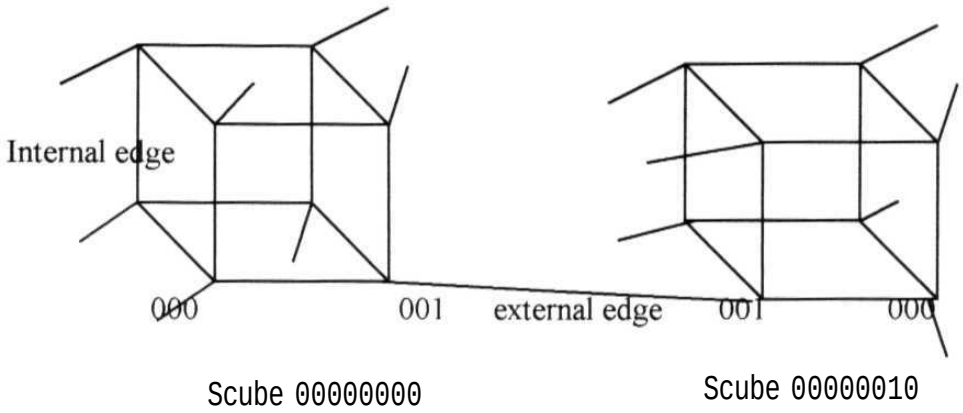


Figure 2.9 Two adjacent Scubes in 11-HHC

Formally, for $k=2^m+m$, an k-HHC is a 2 -nodes graph where each node is represented in binary number system as $b_{k-1}b_{k-2}\ldots b_0$. The address of a node is divided into two parts, F part and S part, and is represented as a two-tuple (f,s). The F part is the $(k-m)$ bits binary number $b_{k-1}b_{k-2}\ldots b_m$, representing the address of Scube in which the node is located. The S part is the m bits binary number $b_{m-1}b_{m-2}\ldots b_0$, representing the address of the node within the Scube. A node (f,s) is connected to m nodes in the same Scube with addresses (f,s® 2'), 0<j<m and to the node with address $(f\oplus 2^s,s)$ in the Fcube. Thus the degree of a node in HHC is given as

$$r_{HHC} = m+1 \qquad (2.39)$$

where m is the dimension of Scube. The diameter of k-HHC is given as

$$D_{HHC} = 2*m \qquad (2.40)$$

The average internode distance is given as [Dandamudi and Eager, 1991]

$$\bar{d}_{HHC} = \frac{m2^{m-1}}{N} + \left(\frac{2^k - 2^m}{N}\right)\left[m + \frac{(k-m)2^{k-m-1}}{2^{k-m}-1}\right] \qquad (2.41)$$

Simple routing procedures can be developed by dividing the routing problem logically into two parts. The first is concerned with packet transfer within Fcube (between Scubes) and the second part is for transferring the packet within the Scube. Shortest routing algorithms **are** used for each of these two parts.

2.3.6 Discussion

The topologies discussed above are now evaluated using the framework proposed in Section 2.2. Table 2.1 summarizes the properties - number of nodes, node degree, diameter and average internode distance - of the regular topologies namely Binary Hypercube, Multidimensional Torus, MS Net, deBruijn Graph, Shuffle Net, GEM Net, **Star** Graph, Cayley Graph of Constant node degree 4, Cube Connected Cycles and Hierarchical Hypercube. From the table, it is clear that MS Net, Cayley Graph of Constant Degree Four and Cube Connected Cycles have fixed node degree. Among the topologies that have variable node degree depending on the number of nodes, it may be also observed that Hierarchical Hypercube can accommodate more number of nodes with less node degree, while Binary Hypercube has the highest node degree.

The node degree in Shuffle Net and de Bruijn Graph can be varied without depending on the number of nodes. But, node degree has impact on diameter and average internode distance. Depending on the performance (diameter and average internode distance) required, the cost of the network (transceivers) for Shuffle Net and de Bruijn Graph based networks varies. Hence MS Net, Cayley Graph of Constant Node Degree Four, Cube Connected Cycles, Shuffle Net, de Bruijn Graph and Hierarchical Hypercube are better candidates for cost conscious solutions, as they need less number of transceivers.

Topology	No. of Nodes	Node Degree	Diameter	Average Internode Distance
Binary Hypercube	$N=2^n$	n	n	$\frac{N*n}{2(N-1)}$
Multidimensional Torus	$N=k^n$	2*n	k*n/2	$\lfloor\frac{k^2}{4}\rfloor*\frac{n}{k}$
MS Net	$N=k^2$	2	$k=\sqrt{N}$	$\frac{k}{2}+1-\frac{4}{k^2}$
de Bruijn graph	$N=k^D$	k	$D=\log_k^N$	
Shuffle Net	$N=k*p^k$	p	2k-1	$\frac{k\,p^k(p-1)(3k-1)-2k(p^k-1)}{2(p-1)(k\,p^k-1)}$
GEM Net	N=K*M	p≤M	log_p^M+K-1	
Star Graph	N=n!	n-1	$\lceil 3(n-1)/2 \rceil$	
Cayley Graph of Constant Degree 4	$N=n*2^n$	4	3n/2	
Cube Connected Cycles	$N=D*2^D$	3	$D*\frac{5}{2}-1$	$D*\frac{7}{4}-3+\frac{D+1}{2^{D-1}}$
Hierarchical Hypercube	$N=2^{m+2^m}$	$m+1$	$2*m$	$\frac{m2^{m-1}}{N}+(\frac{2^k-2^m}{N})[m+\frac{(k-m)2^{k-m-1}}{2^{k-m}-1}]$, where $k=m+2^m$

Table 2.1 Properties of some of the popular regular logical topologies

A comparison of the diameter of various regular topologies discussed above is presented in Figure 2.10(a) and Figure 2.10(b). The node degree in Shuffle Net, de Bruijn graph

was chosen as 2 for the comparison of diameter. The results show that Star Graph has lowest diameter of all the topologies studied.

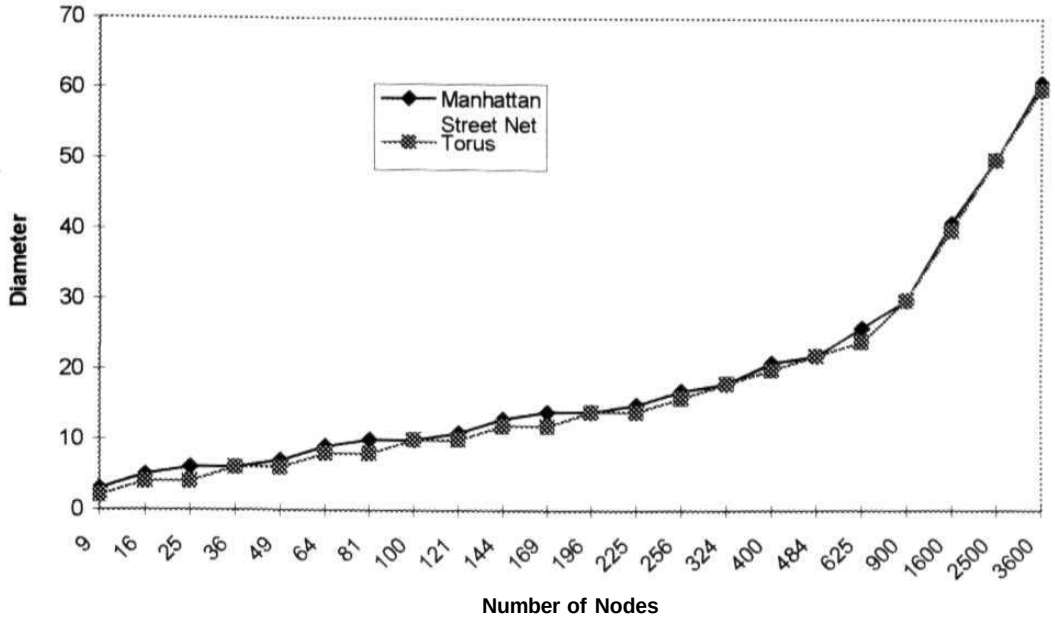
Figure 2.11 (a) and Figure 2.1 l(b) give a comparison of the average internode distance of various regular topologies. It is observed that Binary Hypercube has lowest average internode distance. However, the node degree of binary hypercube increases logarithmically.

From these figures, it is observed that Cayley Graph of Constant Degree Four has comparable performance (diameter and average internode distance) with the best, among the topologies studied. Hierarchical Hypercube may be a better alternative, if one wishes to make use of locality of reference, as it has low average internode distance, low diameter and very low increase of node degree with size of network.

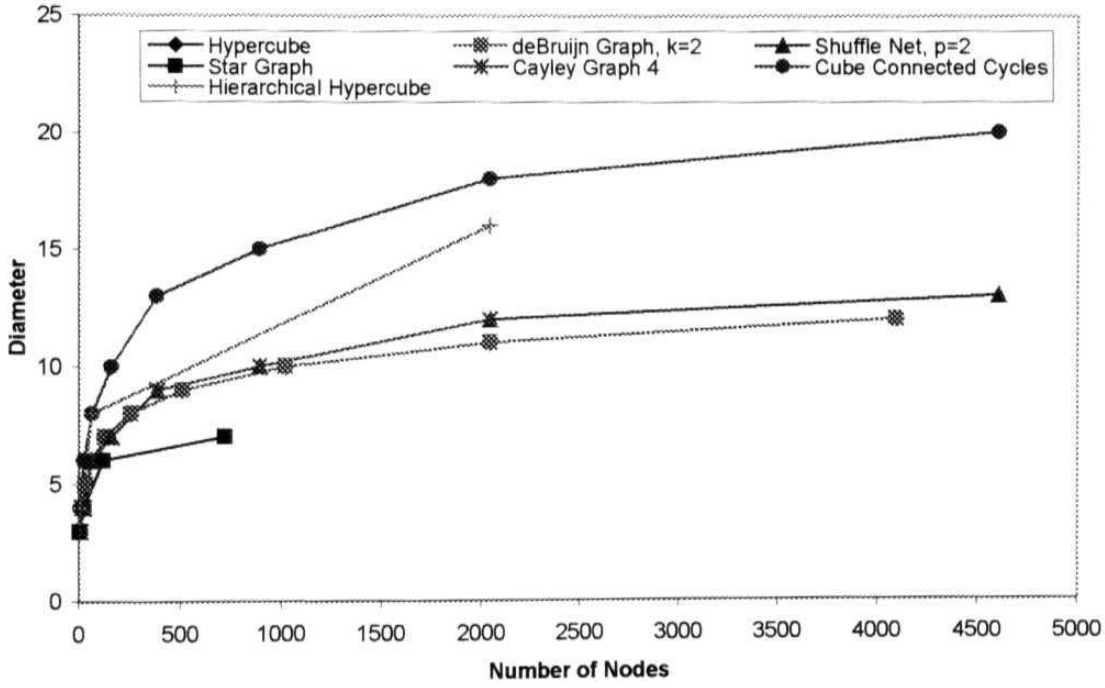
Simple routing can be achieved in all these regular topologies because the interconnection among the nodes is governed by different well-defined mathematical functions.

However, fault tolerance or reliability of networks based on these regular logical topologies is limited to the connectivity of the logical topology, which is bounded by the node degree [Banerjee et. al., 1999]. This dependence of fault tolerance on connectivity (node degree) can be attributed to the fact that most of the regular topologies proposed as logical topologies of lightwave networks are adopted from parallel computer interconnection network architectures.

As the channels in a lightwave network based on a broadcast topology represent logical division of optical bandwidth and assuming that tuning ranges of network interfaces (transmitters and receivers) are large enough to cover the all channels, reconfiguration (redefining the links of nodes in the network) of the topology to cope with node failures and additions can be used to maintain connectivity and hence to achieve the fault tolerance.

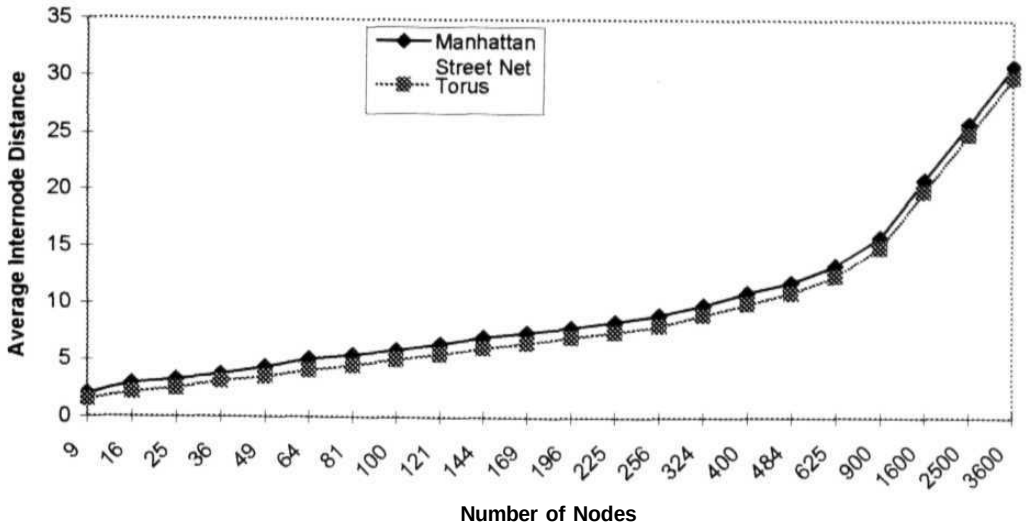


(a)

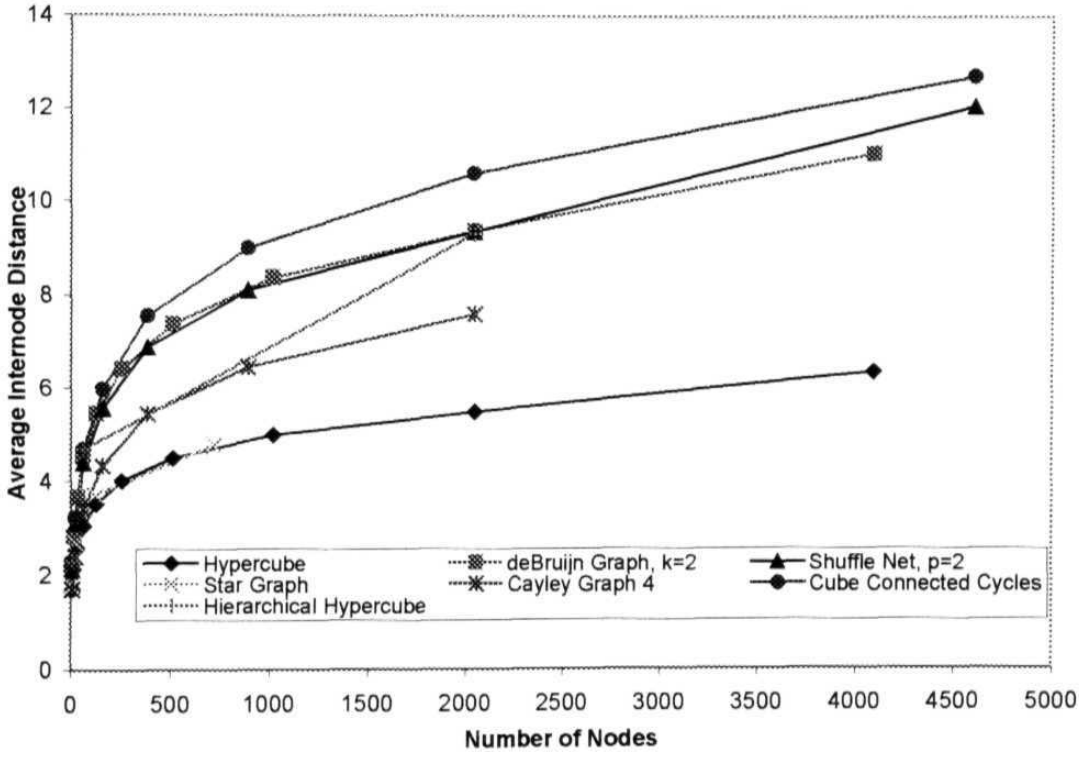


(b)

Figure 2.10 Comparison of Diameter in various regular topologies.



(a)



(b)

Figure 2.11 Comparison of Average Internode Distance in various regular topologies.

Since the node changes in the network occur randomly, reconfiguration for tolerating faults and expanding the network should be defined as a natural process. Current solutions for dynamic reconfiguration on logical topologies are aimed at optimizing the topology according to the changes in network traffic [Labourdette, 1991; Banerjee and Sarkar, 1992; Banerjee and Mukherjee, 1993]. *These solutions cannot be applied for tolerating changes in the network because the new optimized connectivity computed may vary with old connectivity diagram to a greater extent.*

As discussed earlier, reconfiguration affects the network performance in a similar way like other properties (diameter and average internode distance, etc.) does (refer Chapter 1). Hence, the reconfiguration method for fault tolerance and network expandability should be defined together with topology design to ensure that the topology is reconfigurable.

2.4 Reconfiguration As A Logical Topology Design Issue

By considering reconfiguration as a topology design issue, we mean that the reconfiguration algorithm should be defined while designing the logical topology for bypassing node failures and for adding new nodes. When changes (node failures or additions) occur in the network, the topology size either decreases or increases. *Defining reconfiguration methods for all existing logical topologies may not be possible due to the following facts.*

- *Many of the existing logical topologies do not allow arbitrary sizes.*
- *The connectivity among nodes is defined based on network size. That is, the links are defined as a function of number of nodes in the network. Hence, reconfiguration may change many links each time it is invoked.*

To overcome these problems, we assume a structure (topology) of larger size for which the network performance is affordable. This structure is called *base topology* Each node in the network takes a unique position in the base topology, identified with node id. *The position of a node can be changed according to traffic changes using optimal node placement algorithms.* Networks with smaller sizes can be constructed by assuming some of the nodes in the base topology as failed nodes. As the network size varies up to

a maximum of the size of base topology, the structure of topology changes depending on which nodes are present in the base topology. Hence, the topology that also considers reconfiguration as a design issue is called as *dynamically* ***reconfigurable*** *topology,* or simply, ***reconfigurable topology.***

Whenever a node fails, the neighbors of failed node get interconnected such that connectivity is maintained and routing properties of base topology are preserved Similarly, whenever a node is added to the network, it takes a unique position in the base topology and establishes connections accordingly. Hence, *this method of reconfiguration dynamically changes the structure of logical topology moving back and forth from the base topology.* We call this method of reconfiguration that makes changes to the neighboring nodes of a place to which a node is allocated or deallocated, as *local perturbations.* Reconfigurable topology that is designed using local perturbations is referred as ***perturbed topology.*** *Because reconfigurable topologies are designed using local perturbations, they named after the base topologies with one of the prefixes - reconfigurable or perturbed.*

To understand the concept of reconfiguration based on local perturbations, let us consider a base structure, 2-D Mesh with 9 places as shown in Figure 2.12(a). Assume that 7 nodes placed in the base structure of Figure 2.12(a), resulting a logical topology shown in Figure 2.12(b). Now, assume that a node wishes to add at (2,3) position in the logical topology shown Figure 2.12(b). Then nodes at (2,2), (1,3) and (3,3) positions connect to the newly added node resulting in the logical topology shown in Figure 2.12(c). Now assume that node at (3,2) position wishes to be deleted, then its neighbors in positions (3,1) and (3,3) get interconnected leading to the structure shown in Figure 2.12(d).

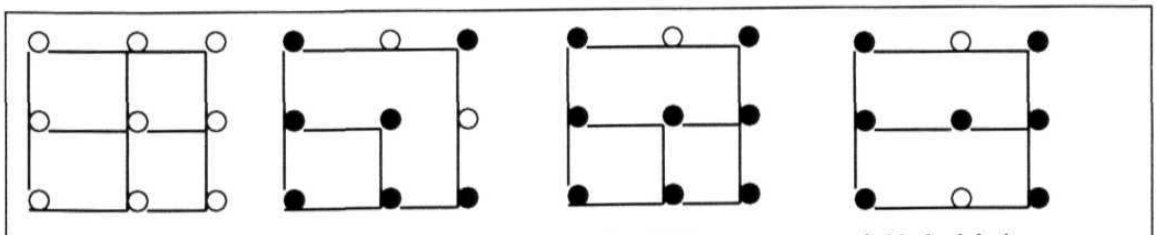


Figure 2.12 Reconfiguration in maximal structure approach

It may be noted that there are k! permutations (where k is the node degree or regularity of the logical topology) for interconnecting the neighbor nodes, when a node failure addition occurs. However, not all the permutations retain the structural properties of base topology. Hence, it is necessary to select a way of interconnecting neighboring nodes that retain the properties of base topology. For this reason, reconfigurable topologies are identified with their base topology.

It may be observed that reconfiguration based on local perturbations has the following advantages.

- The disturbance to the network during reconfiguration phase is minimum.
- Reconfiguration process can be invoked simultaneously.
- Assuming that allocation of a place to a node is done based on the traffic requirements, the throughput, utilization and delay performance after reconfiguration would remain close to optimal values.

The next chapter discusses the possibility of using local perturbations paradigm for base topologies of well-known regular structures such as Torus and design novel topologies that are easily reconfigurable and are having comparable performance with respect to well-known topologies.

Chapter

3. Design of Dynamically Reconfigurable Logical Topologies

In this chapter, we study the application of local perturbations paradigm for dynamic reconfiguration of well-known topologies such as Torus and also *design a new set of topologies that treat reconfiguration as a logical topology design issue.* Section 3.1 gives introduction to the design of dynamically reconfigurable logical topologies and presents local perturbations as a reconfiguration paradigm. In section 3.2, we investigate the application of local perturbations for well-known topologies such as Torus. Section 3.3, first, discusses the need for a Hamiltonian circuit for maintaining connectivity in logical topology and then discusses the design of dynamically reconfigurable logical topologies based on Circulant graphs with edge disjoint Hamiltonian circuits. Finally, section 3.4 compares the dynamically reconfigurable logical topologies designed in this chapter with the well-known logical topologies studied in the previous chapter.

3.1 Introduction

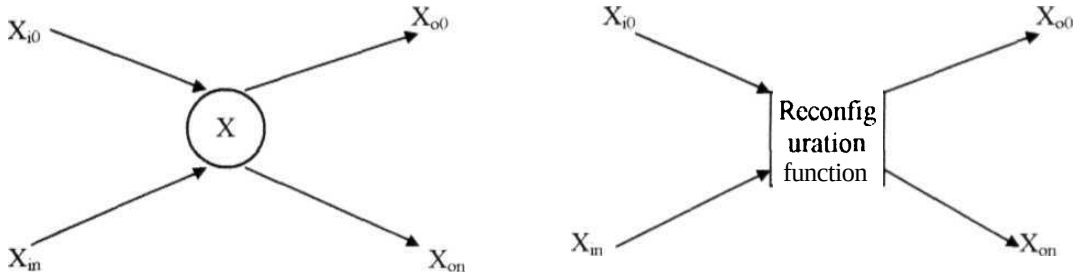
Many regular topologies have been studied in literature as logical topologies for multi-hop fiber optic LAN/MANs. This is mainly because regular topologies facilitate simple routing and use identical hardware. Simple routing is achieved because the interconnection among the nodes is governed by well-defined mathematical functions.

Most of the regular topologies were initially proposed as interconnection networks for parallel computers. *When these regular topologies were adopted as logical topologies for fiber optic LANs/MANs, reconfiguration of logical topology for tolerating node faults and additions was not considered as an issue. However, we believe that dynamic reconfiguration is essential for tolerating network changes.* Access nodes in fiber optic LANs/MANs that use regular topologies as their logical topology, route the packets to the destination nodes and failure of nodes may disconnects the topology. As the LAN/MANs

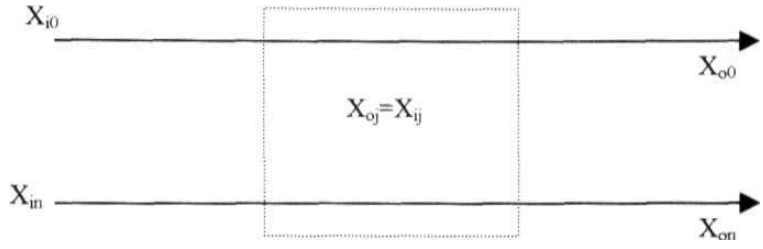
use broadcast (physical) topologies as physical interconnection, reconfiguration of logical topology can be done independent of physical topology by (re)tuning wavelengths of nodes' transmitters/ receivers. *Further, we assume that reconfiguration should be an integral part of logical topology design.* This is because the reconfiguration affects the network performance parameters such as reliability and expandability similar to the way the properties of logical topology such as diameter and average internode distance affects network throughput and packet delays. It is also observed that reconfiguration and properties of topology such as node degree are interdependent.

To facilitate the integration of dynamic reconfiguration into logical topology design, this work proposes Local Perturbations paradigm in the [Reddy and Reddy 2001]. *Local Perturbations paradigm assumes a base topology of maximum size, which gives the most affordable performance.* Each node has a unique position in the base topology. In this work, local perturbations refer to the changes made to immediate neighbors in the network to accommodate dynamic changes within the network. In other words, whenever a node gets added or deleted from the network, immediate neighbors change their logical links in order to retain the connectivity.

For example, consider node X as connected to *n* nodes via incoming links and to *n* nodes via outgoing links, as shown in Figure 3.1 (a). Failure of node X changes the logical links of immediate neighbors as per a **reconfiguration function** that retains connectivity by mapping nodes of incoming links to nodes of outgoing links, as shown in Figure 3.1(b). An example of the mapping where an incoming link i to outgoing link i is shown in Figure 3.1(c).



(a). Connection of incoming and outgoing nodes to node (b). Reconfiguration function



(c). An example mapping between incoming and outgoing nodes of node X

Figure 3. /Perturbations of Node X

It may be observed from the Figure 3.1(b), that there are $n!$ possible mappings between neighboring nodes of incoming and outgoing links of a failed node. *The selection of mapping between neighboring nodes not only needs to maintain connectivity, but also needs to maintain the structural properties of the network such as routing and diameter.*

A node can get added according to its position in the base topology and establishes links accordingly. *Change of logical link is particularly easy for multichannel lightwave networks as it can be done by tuning transmitters and/or receivers of the nodes representing the logical link to a different wavelength.*

Local perturbations have the following advantageous.

- *The number of link changes is minimal.* The number of link changes due to a reconfiguration based on local perturbations is equivalent to the degree of a node.
- *Routing tables of immediate neighbors only need to be changed.*
- *Reconfiguration process takes very little time.* Since the links in a regular topology are usually defined as a mathematical function of node ids, identification of neighbor during reconfiguration can also be computed mathematically.

> *Effect of reconfiguration on network traffic is minimum.* The preservation of connectivity by this paradigm minimizes packet loss, delay during the reconfiguration phase.

The following sections discuss the applicability of local perturbations paradigm for dynamic reconfiguration in existing topologies and proposes a set of new topologies that consider dynamic reconfiguration as an integral part of topology design.

3.2 Local Perturbations in Existing Topologies

In this section, we investigate the possibility of applying local perturbations as reconfiguration methodology for some of the existing topologies such as Ring, Torus, and Hypercube.

It is observed that Ring topology is able to bypass faulty components (nodes or links) and it can tolerate unlimited number of faults. By viewing Ring topology as a ***Hamiltonian*** *circuit* (a circuit which visits every node precisely once), reconfiguration process for bypassing failed nodes can be defined as connecting the previous node with the subsequent one in the network.

Hypothesis 3.1. Connectivity in any topology is maintained if reconfiguration is done along a Hamiltonian circuit.

•

Application of local perturbations to the topologies proposed in the literature, such as Torus, Hypercube, de Bruijn graph and Star graph, is considered as defining reconfiguration functions, from incoming nodes to outgoing nodes, in such a way that routing properties are preserved. Such a reconfiguration function may not guarantee the connectivity in the topology. For example, in Torus topology, each node is connected to four nodes in North, South, East and West directions. Routing between two arbitrary nodes X and Y can be done by moving in row of X to the column of Y and then moving row-wise to reach node Y. Of the different possible mappings among the incoming and outgoing nodes, as shown in Figure 3.2, connecting North-South neighbors and East-West neighbors preserves the routing properties.

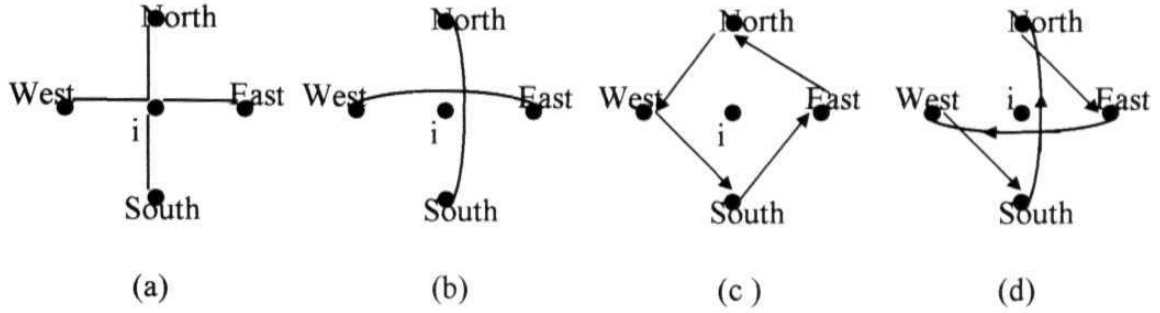


Figure 3.2 Connections and possible perturbations of node i

However, such a reconfiguration leads to disconnection of the topology. For example, consider the Torus topology with 16 nodes shown in Figure 3.3. By assuming that nodes 2,3,5,6,7,8,9,12 and 13 are failed and by applying reconfiguration, the resultant structure becomes a disconnected topology as shown in Figure 3.4.

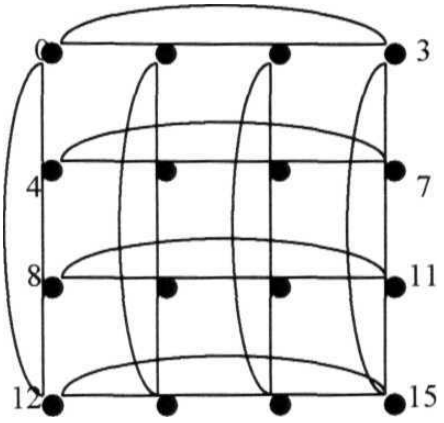


figure 3.3 Torus with 16 nodes

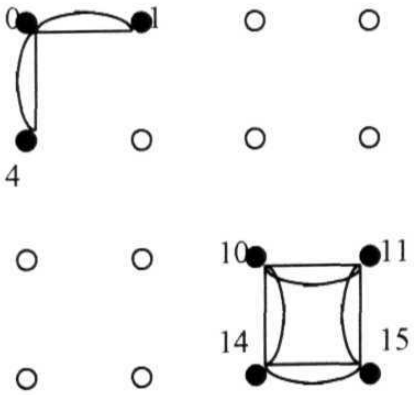


Figure *3.4 Torus structure with perturbations made to the neighbors of failed nodes 2,3,5,6,7,8,9,12 and 13.*

Though reconfiguration along Hamiltonian circuit guarantees connectivity, defining local perturbations such that a Hamiltonian circuit is maintained, becomes a complex task. Moreover, such reconfiguration cannot retain routing properties of the base topology. Hence, additional methods are required for maintaining connectivity in existing topologies. In the following section, we will study Perturbed Torus derived from the base structure, Torus.

3.2.1 Perturbed Torus

Perturbed Torus is a dynamically reconfigurable topology derived from the base Torus structure in which whenever a change in topology occurs, perturbations are made by connecting North-South neighbors and East-West neighbors [Mohan Reddy and Reddy, 1996a]. *However, to retain the connectivity of the nodes in the Perturbed Torus network, we propose to maintain a row (column) so that a node shall be present in every column (row) whenever there exists another node in that column (row).* This row (column) is called as **base row (column).**

To understand how connectivity is maintained in Perturbed Torus using base row (column), consider that the network maintains the first row as the base row. Theorem 3.1 proves the preservation of connectivity in Perturbed Torus, with first row as base row, by showing the existence of a path between two arbitrary nodes. The proof can be generalized for a Perturbed Torus that maintains a base column.

Theorem 3.1. Maintenance of base row (column) and perturbations made by connecting North-South neighbors and East-West neighbors preserves the connectivity in the Perturbed Torus.

Proof: Whenever a node exists in a column, according to the concept of base row, there exists a node in the same column of the base row.

Nodes in a row or column of Torus topology are connected in the form of ring. Local Perturbations maintain these rings in rows and columns. It is because perturbations are made by connecting North-South neighbors and East-West neighbors of a deleted (failed) node.

To prove the existence of path between any two nodes, consider two nodes X and Y shown in Figure 3.5. Node Y can be reached from node X by following the steps below.

a) Move in the column ring of node X to the base row,

b) Then move in the base row to the column of the node Y, and finally

c) Move in the column ring of node Y to reach node Y.

Thus there exists a path between X and Y, and hence they are said to be connected. Hence, the connectivity of the network is maintained. •

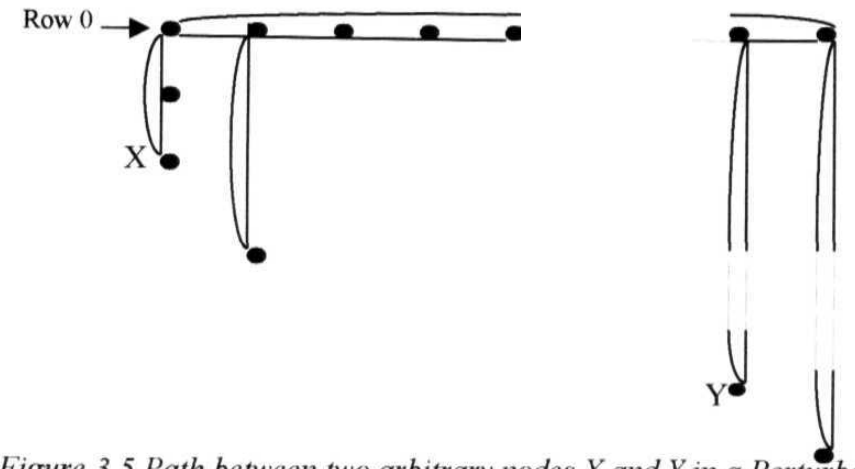


Figure 3.5 Path between two arbitrary nodes X and Y in a Perturbed Torus

Reconfiguration.

Reconfiguration in Perturbed Torus is done when a node is either gets deleted from or added to the base topology. To simplify the discussion on reconfiguration, without loss of generality, we assume row 0 is used as base row. To explain the reconfiguration process, we distinguish the node's id from the node's position in base topology. Node's position is identified with the row and column in the base Torus structure. Usually, in a base Torus structure with n rows and n columns, node i takes the position $(i/n, i\%n)$, that is at i/n row and $i\%n$ column of the base topology.

Algorithm 3.1 describes the reconfiguration when node with id i is deleted. According to this algorithm, whenever a node in the base row gets deleted, the South neighbor, if present, is replaced into the base row and perturbations are made at the South neighbor of the deleted node. If the deleted node is not in the base row, perturbations are made by connecting its East neighbor with West neighbor and North neighbor with South neighbor.

Algorithm 3.1. Reconfigure the topology when node i *is deleted from the Perturbed Torus derived from a base Torus structure with* n *rows and* n *columns.*

if position of the deleted node i is (0,i) then
*if South neighbor with id j at position (j n,j%n) where j%n*i*, is present*
Move node j to position (0,i) and establishes links with East, West, North and South neighbors at position (0,i).
Establish links between East and West neighbors and North and South neighbors of position (j/n,j%n).
else
Establish links between East and West neighbors at position (0,i).
endif.
else if the position of deleted node i is (i/n,i%n), i/n≠0 then
Establish links between East and West neighbors and North and South neighbors of position (i/n,i%n).
endif.

To explain the process of node deletion, consider a base Torus topology shown in Figure 3.6(a). Assume that node 5 has been deleted. By connecting neighbors of node 5, we get the resultant topology as shown in Figure 3.6(b). Now, assuming node 1 is deleted, the resultant topology after moving node 9 into the position (0,1) becomes as shown in Figure 3.6(c).

Continuing the perturbations with the loss of nodes 4, 6 and 7 in the perturbed topology shown in Figure 3.6(b), we get the resultant structure equivalent to a regular Torus reduced by a row. Similarly, *deletion of all nodes of a column will reduce the structure by one column. The reduced structure behaves as a regular Torus of lower dimension.*

Addition of a node in Perturbed Torus is explained in Algorithm 3.2. According to this algorithm, when a node is added, it occupies its default position in the base Torus structure, if there exists a node in the base row whose default position is either in the base row or closer to the base row than the node being added. Otherwise, the node takes the position in the base row, by moving the node, if exists, in the base row to its default position.

To explain the process of node addition, consider Perturbed Torus with 14 active nodes and two deleted nodes (node 1 and 5) shown in Figure 3.6(c). If node 5 wishes to enter the network, it informs the network and occupies position in the base row by moving node 9 to its default position. The resultant structure is shown in Figure 3.6(d). Now

assuming the addition of node 1, the resultant topology becomes as shown in Figure 3.6(a).

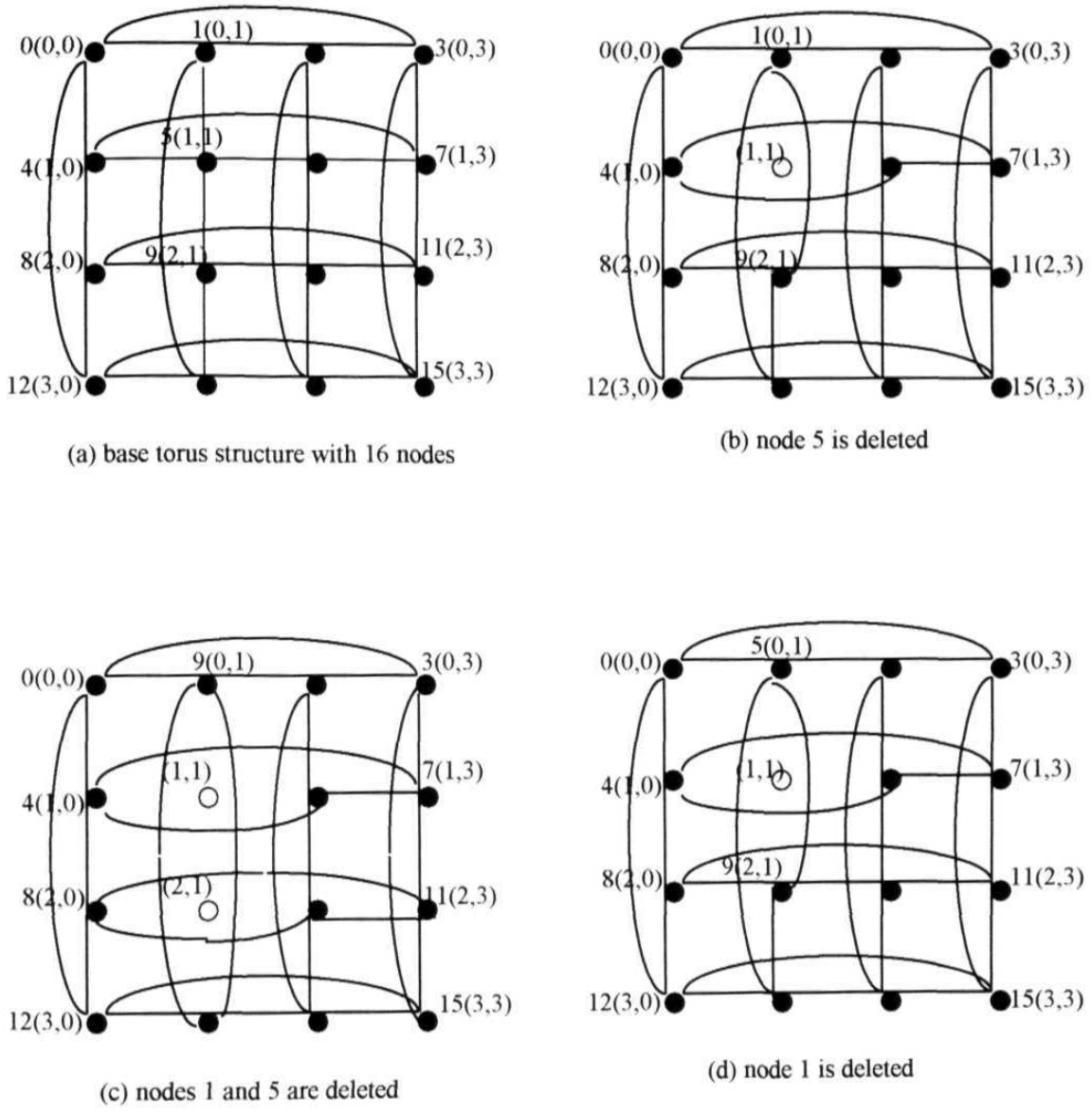


Figure 3.6 Reconfiguration in Perturbed Torus with base Torus structure of 16 nodes.

Algorithm 3.2. Reconfigure the topology when node i *is added to the Perturbed Torus derived from a base Torus structure with* n *rows and* n *columns.*

if i/n = = 0 then
if there exists a node j at position (0,i) then
Move nodej to position (j/n,j%n) where j%n = i and establish links for node j with East, West, North and South neighbors at position (j n,j%n).
Establish links for node i with East, West, North and South neighbors at position (0,i).
else
Establish links for node i with Exist, West, North and South neighbors at position (0,i).
endif.
else if i/n≠0 then
if there exists a nodej at position (0,i%n) then
if j/n -0 then
Establish links for node i with East, West, North and South neighbors at position (i/n,i%n).
else if (j/n > i/n) then
Move node j to position (j/n,j%n) and establish links for node j with East, West, North and South neighbors at position (j n,j%n).
Establish links for node i with East, West, North and South neighbors at position (l),i%n).
else then
Establish links for node i with East, West, North and South neighbors at position (i/n,i%n).
endif.
else
Establish links for node i with East, West, North and South neighbors at position (0,i%n).
endif.
endif.

Routing.

Perturbations are selected in order to preserve the routing properties of Torus topology. Regular Torus uses a simple routing technique - visit the column of the destination node, first, and then visit the destination node by moving in that column. This simple routing is achieved by making use of the rings in rows and columns of a regular Torus. For example, consider the regular Torus shown in Figure 3.6(a). Assume that node 5 wants to send a packet to node 15. To reach the column of node 15, the packet will be routed

through nodes 6 and 7, by moving in the row ring of node 5. Finally, the packet is moved to node 15 through node 11 by moving in the column of node 15.

A simple variation of the above routing technique, explained in Algorithm 3.3, is proposed for Perturbed Torus. According to this algorithm, whenever a source node wants to send a packet to a destination node, the packet first moves in the row of source node to reach the column of the destination node. If the column of the destination node cannot be reached, the packet continues to move in the column nearest to the column of the destination node towards the base row, each time trying to reach the column of the destination node. Finally, the packet reaches the destination node by moving in the column of the destination node.

Algorithm 3.3. Send a message from the current node C to the destination node D which is originated at source node S in the Perturbed Torus derived from a base Torus structure with n *rows and* n *columns.*

if(C = = D) then
    Send message to local node C.
else if(C%n = = D%n)
    if the neighbor in North-South ring close to D knows that D is not present then
        Send negative acknowledgement to the Source indicating that destination node D is failed.
    else
        Move the packet in North-South ring to the neighbor close to D than C.
else if neighbor at position (C/n,C%n) in East-West ring which is close to I) is present then
    Move the packet in East-West ring to the neighbor close to D than C.
else if the neighbor in East-West ring close to D knows that D is not present then
    Send negative acknowledgement to the Source indicating that destination node D is failed.
else
    Move towards the base row.
endif.

For example, in the Perturbed Torus of 15 nodes with node 5 as a deleted node, shown in Figure 3.6(b), let us assume node 7 wants to send a packet to node 13. The packet first moves in the row of node 7 to node 6. At node 6, since node 5 is failed, the packet will be routed to the base row to node 2. From node 2, the packet will be sent to node 1 which is in the column of the destination 13, from there the packet reaches the destination

node 13. As another example, consider a packet addressed to node 5 from node 15 will be routed to node 13 through node 14. Then from node 13, the packet will reach node 9. At node 9, the packet will be discarded as node 9 finds its north direction link connected to node 1 instead of node 5. An acknowledgement packet will be sent by node 9 to node 15 about the status of node 5.

Thus whenever a source node wishes to communicate to a destination node, it just sends the packet assuming that the destination node is present. If the destination node is not present, the neighboring nodes send acknowledgement packet indicating the status of the destination node. Hence, *each node need not keep the status of all nodes in the network. It is sufficient to keep the status of neighboring nodes.*

Diameter.

The following theorem proves the maximum diameter of a Perturbed Torus derived from a base Torus structure of size N is $2*\lfloor \lceil \sqrt{N} \rceil /2 \rfloor + 1$.

Theorem 3.2. The maximum diameter of a Perturbed Torus derived from a Torus of N nodes is $2*\lfloor \lceil \sqrt{N} \rceil /2 \rfloor + 1$.

Proof: Consider any two arbitrary nodes X and Y in the Perturbed Torus. For simplicity, we assume that base row is maintained. The same can be easily proved for maintaining base column.

Case I. Let us assume both of the nodes X and Y belong to either the same row or the same column. Then as the nodes in the same column or row of Perturbed Torus are connected in the form of bi-directional ring of maximum size $\lceil \sqrt{N} \rceil$, the distance between X and Y is less than or equal to $\lfloor \lceil \sqrt{N} \rceil /2 \rfloor$.

Case 11. Let the nodes X and Y belong to different rows and columns. If there exists a node Z such that either row(Z)=row(X) and column(Z)=column(Y) or row(Z)=row(Y) and column(Z)=column(X). Then we have a path from node X to node Y through node Z. The distance between the nodes X and Z is less than or equal to $\lfloor \lceil \sqrt{N} \rceil /2 \rfloor$. This is because they belong to the same row or column of size $\lceil \sqrt{N} \rceil$. Similarly, the distance between the nodes Z and Y is less than or equal to $\lfloor \lceil \sqrt{N} \rceil /2 \rfloor$ as both of them belong to the same column or row of size $\lceil \sqrt{N} \rceil$. Hence the distance between the nodes X and Y, which

is the sum of the distance between the nodes X and Z and the distance between the nodes Z and Y, is less than or equal to $2*\lfloor\lceil\sqrt{N}\rceil/2\rfloor$.

Case III. Let the nodes X and Y belong to different rows and columns. Also assume that there exists no node Z such that either it is in the column of node X and row of node Y, or it is in the row of node X and column of node Y.

Let X' and Y' are the nodes in the base row corresponding to the columns of nodes X and Y, respectively. Assume no other pair of nodes in columns of nodes X and Y belongs to the same row. Then the total number of nodes in the columns of X and Y is less than or equal to $\lceil\sqrt{N}\rceil+1$. Let m and n be the number of nodes in the columns of nodes X and Y, respectively. Then $m+n\leq\lceil\sqrt{N}\rceil+1$. The distance between the nodes X and X^1 is less than or equal to $m/2$ and the distance between the nodes Y and Y' is less than or equal to $n/2$. Hence the sum of the distance between nodes X and X', and the distance between nodes Y and Y' is less than or equal to (m+n)/2 which is less than or equal to $\lfloor\lceil\sqrt{N}\rceil/2\rfloor+1$. The distance between X' and Y' is less than or equal to $\lfloor\lceil\sqrt{N}\rceil/2\rfloor$. Hence the distance between the nodes X and Y, which is the sum of the distance between nodes X and X', the distance between nodes X' and Y', and the distance between nodes Y' and Y, is less than or equal to $2*\lfloor\lceil\sqrt{N}\rceil/2\rfloor+1$.

Suppose that X" and Y" are the nodes in the columns of nodes X and Y that belong to the same row and are nearer than the nodes X' and Y' from nodes X and Y, respectively. Also assume that no other pair of nodes are nearer than the nodes X" and Y" from nodes X and Y, respectively. Then the sum of the distances between nodes X and X", and the distance between nodes Y and Y" is less than the sum of the distance between nodes X and X', and the distance between nodes Y and Y', i.e., $\lfloor\lceil\sqrt{N}\rceil/2\rfloor+1$. Also the distance between the nodes X" and Y" is less than or equal to $\lfloor\lceil\sqrt{N}\rceil/2\rfloor$.

Hence the distance between X and Y, which is the sum of the distance between nodes X and X", the distance between nodes X" and Y", and the distance between nodes Y" and Y, is less than or equal to $2*\lfloor\lceil\sqrt{N}\rceil/2\rfloor+1$.

Therefore the maximum diameter of Perturbed Torus is $2*\lfloor\lceil\sqrt{N}\rceil/2\rfloor+1$.

It may be noted that *Perturbed Torus retains simple routing properties of Torus structure and also has the maximum diameter equivalent to that of base Torus topology.*

It may be observed that defining reconfiguration using local perturbations in traditional topologies may lead to disconnected graph upon the application of reconfiguration for successive node failures.

To understand further, consider Binary Hypercube topology as a base topology. As the neighbor nodes in Binary Hypercube differ by one bit, packets are routed along the neighbors that reduce the bit distance to the destination. To preserve this routing property, reconfiguration due to node failure should change the links of neighbors so that the bit distance between neighbors is always limited to one. This leaves with the choice of reconfiguration where neighbor nodes develop self-loops along the direction of failed node. Application of reconfiguration due to the failure of all neighbors in a Binary Hypercube would cause isolating the node from rest of the nodes in the network.

Defining special methods similar to the one used in Perturbed Torus (maintaining base row), so that the reconfiguration always maintain connectivity and preserves structural properties in traditional topologies, is found to be infeasible. This is because such special methods complicate the reconfiguration process, which is against the spirit of local perturbations.

Hence, we investigated novel base topologies that allow local perturbations for maintaining connectivity while preserving other structural properties. *In the following section, we discuss a subset of Circulant graphs that are constructed with edge-disjoint Hamiltonian circuits and design reconfigurable topologies using local perturbations paradigm.*

3.3 Local Perturbations in Circulant Graphs with Edge-disjoint Hamiitonian Circuits

As studied in **the** previous section, reconfiguration along a Hamiitonian circuit in a logical topology guarantees network connectivity. This fact motivated us to consider Circulant graph as a base topology since it is a set of rings with at least one Hamiitonian circuit. In this section, we study a subclass of Circulant graphs constructed with a set of edge-disjoint Hamiitonian circuits. These Circulant graphs are used as base topologies for **the** dynamically reconfigurable topologies proposed in this thesis. We start our discussion by first defining Circulant graph.

Circulant Graph. A p-node graph is Circulant with respect to a given set S of integers, if each node has a unique label in the range 0 through p-\ and each node x is connected to all nodes with labels $(x + s) \% p$, where $s \in S$. We call the set S as *connection set* and the members of the set *as jumps* or *offsets.* The p-node Circulant graph with connection set S is denoted by $C_{p,s}$.

An example of Circulant graph with 8 nodes and a connection set $\{+1,-1,-3,+3\}$ is shown below.

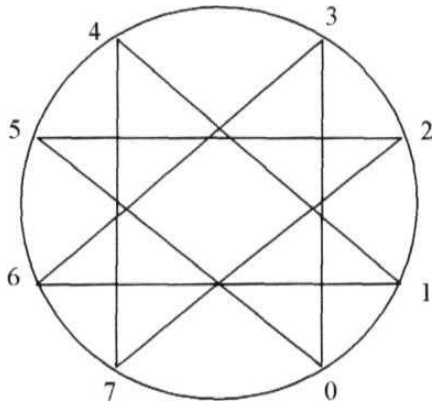


Figure 3.7 Circulant graph with 8 nodes and connection set {-3,-1, t 1, + 3}

In the figure above, we can identify the paths associated with a jump as the path obtained with edges representing that particular jump. For example, the path associated with jump + 1 is 0,1,2,3,4,5,6,7,0. Similarly, the path associated with jump -3 is 0,5,2,7,4,1,6,3,0. Interestingly, these paths represent Hamiitonian circuits in the graph. The following

theorem proves that such Hamiltonian circuits exist if the jump *s* is relatively prime with respect to the number of nodes/?.

Theorem 3.3. Let $V_0, V_1, \ldots, V_p$ *be the sequence of numbers satisfying the following conditions.*

1. $V_i = (V_{i-1} + s)\%p$
2. $V_0 < p$; $s < p$
3. *s is relatively prime with respect to p.*

Then all the elements in the sequence are distinct except for $V_0 = V_p$.

Proof: By definition of $V_i = (V_{i-1} + s)\%p$, $V_i = V_{i-1} + s - k_i * p$, for some $k_i \geq 0$. By applying this definition recursively, we get

$$V_i = V_0 + i * s - K_i * p \qquad (3.1)$$

where $K_i = \sum_{m=1}^{i} k_m$. Let us assume $V_i = V_j$ for some *i* and *j*. Using Equation (3.1), we get

$$V_0 + i * s - K_i * p = V_0 + j * s - K_j * p$$

This implies

$$(i - j) * s = (K_i - K_j) * p \qquad (3.2)$$

Since *s* is relatively prime with respect to *p*, the above Equation (3.2) is valid, i.e., two numbers of the sequence V_i and V_j are equal, only if (*i-j*) is a multiple of *p*. Since *i* and *j* can take values in the range 0 to *p*, (*i-j*) is a multiple of *p* only for *i*=0 and *j*=*p*. Hence all the numbers of the sequence are distinct except for $V_0 = V_p$. •

Theorem 3.4. In a Circulant graph $C_{p,S}$, *if a jump,* $s \in S$, *is relatively prime with respect to p, then the path associated with jump s forms a Hamiltonian circuit.*

Proof: Consider any node V_0 and the sequence of nodes $V_0, V_1, \ldots, V_p$, which satisfies the condition: $V_i = (V_{i-1} + s)\%p$, where $s \in S$ and s is relatively prime with respect to *p*. The sequence is a Hamiltonian circuit for the following two reasons.

Firstly, the sequence is a path. This is because any two successive nodes of the sequence are adjacent (By the definition of Circulant graph).

Secondly, because *s* is relatively prime with respect to *p,* all the nodes in the sequence are distinct except for $V_0=V_p$ (according to Theorem 3.3). •

In the Circulant graph of Figure 3.7, all the four jumps of the connection set *S* are relatively prime with respect to *p.* Hence, according to the Theorem 3.4, we have four Hamiltonian circuits, each one being associated with a jump. Moreover, these Hamiltonian circuits are edge disjoint as the jumps of the connection set *S* are distinct. This means that the Hamiltonian circuits associated with different jumps, in the connection set *S* of the circulant graph $C_{p,S}$ are edge disjoint.

Hypothesis 3.2. The Hamiltonian circuits associated with different jumps in the connection set S *of the circulant graph* $C_{p,S}$ *are edge disjoint.*

In the following subsections, we study dynamically reconfigurable topologies that use three different Circulant graphs with edge-disjoint Hamiltonian circuits as the base topologies. These base topologies are akin to 2-D Torus, n-D Torus and Binary Hypercube respectively, in their performance.

3.3.1 Reconfigurable Circulant Graph - T

In this section, we design a dynamically reconfigurable topology derived from the base Circulant graph $C_{p,S}$ with $p=n^2-1$ and $S = \{-n,-1,+1,+n\}$ which reconfigures as per the local perturbations paradigm [Mohan Reddy and Reddy, 1996b]. The base Circulant graph $C_{p,S}$ with $p=n^2-1$ and $S = \{-n,-1,+1,+n\}$ is referred as **Circulant Graph–I or CG-I,** in the rest of the thesis. Before designing this reconfigurable topology, we first study the properties of the base Circulant graph in order to prove that the structural properties of base topology are retained.

An example of the base CG-I topology, $C_{p,S}$ with $p=8$ and connection set $S = \{-3,-1,+1,+3\}$ is shown in Figure 3.7. A mesh-like representation of the same is shown in Figure 3.8.

Before studying some more properties of this base topology, first we will define an image of anode.

Image of a Node. X and Y are said to be images of each other, iff X%n= Y%n.

In the CG-I topology shown in Figure 3.8, node 0,3 and 6 are said to be images of each other. Similarly, nodes 2 and 5 are said to images of each other. It is observed from Figure 3.8 where the nodes are represented in 2-D Euclidean space that an image of node can be reached in maximum of *n* hops along the jumps +1 or –1.

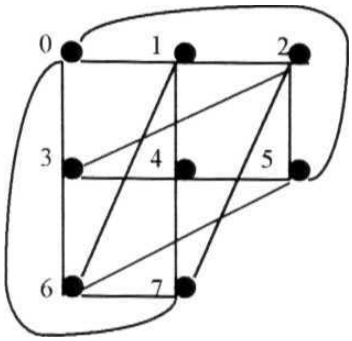


Figure 3.8 Mesh-like representation of CG-I topology $C_{p,S}$ with p 8 and S {-3,-

Hypothesis 3.3. In the CG-I topology, $C_{p,S}$, an image of a node can be reached in maximum of n*hops along the jumps +1 and -I.*

■

It may be observed from the Figure 3.7 and Figure 3.8 that there exist four edge-disjoint Hamiltonian circuits, each being associated with a jump in the connection set *S.* Corollary 3.1 proves the existence of such Hamiltonian circuits in the base Circulant graph.

Corollary 3.1. There are four edge-disjoint Hamiltonian circuits in CG-I topology, $C_{p,S}$ with p n^2-1 and S = $\{-n,-1,+1,+n\}$

Proof: According to Hypothesis 3.2, since all the jumps in the connection set S are distinct, it is sufficient to show that the paths associated with jumps are Hamiltonian

circuits. Further, according to Theorem 3.4, it is sufficient to prove that all these jumps are relatively prime with respect to p.

+1 and -1 are relatively prime with respect to p. Suppose n is not relatively prime with respect to p. Then p/n is an integer, say, k. But $p/n = (n^2-1)/n$ is not an integer. Hence, there is a contradiction in our assumption. Therefore, n has to be relatively prime with respect to p. By similar argument, we can show that $-n$ is relatively prime with respect to $p = n^2-1$.

Thus all the jumps of connection set S are relatively prime with respect to p. Hence, the Hamiltonian circuits associated with four different jumps are edge disjoint. •

Routing in base Circulant Graph-I.

A distributed shortest path routing method in the base Circulant Graph-I topology is described in the following Algorithm 3.4.

Algorithm 3.4. Send a message from the current node C to the destination node D which is originated at source node S in a CG-I topology, C_s with $p = n^2-1$ and connection set $S = \{-n,-1,+1,+n\}$.

```
if (C = D) then
        Send message to local node C
else
        Compute the distance and jump to move on.
        Distance = (D-C+p) %p
        if (Distance % n n/2)
                Move along jump -1.
        else if (Distance % n - - 0)
                if (Distance > p/2)
                        Move along jump -n.
                else
                        Move along jump + n.
                endif.
        else
                Move along jump + 1.
        endif.
endif
```

Algorithm 3.4 describes routing a packet by making use of the rings present in the mesh-like representation of Circulant graph. It may be observed that Hamiltonian circuits associated with +1 and -1 jumps form a bi-directional ring. Similarly, Hamiltonian

circuits associated with $+n$ and $-n$ jumps form another bi-directional ring. To reach the destination node, a packet first moves along the bi-directional ring of jumps -1 and +1, and then along the bi-directional ring of jumps $+n$ and -//.

Diameter of base Circulant Graph-I.

The diameter of Circulant Graph-I is found to be n. Theorem 3.5 proves this.

Theorem 3.5. The diameter of Circulant graph $C_{p,S}$ with $p=n^2-1$ and connection set

Proof: According to Corollary 3.1, there exist four Hamiltonian circuits, along the jumps $-n$, -1, +1 and n. The Hamiltonian circuits along the jumps +1 and -1 together form a bi-directional ring while Hamiltonian circuits along the jumps $+n$ and $-n$ together form another bi-directional ring. The maximum hop-distance between any two arbitrary nodes along the bi-directional ring of jumps +1 and -1 is $(n^2-1)/2$. This hop-distance is further reduced by making use of $+n$ and -A; jumps.

Let the hop-distance, d, between two arbitrary nodes X and Y on bi-directional ring of jumps +1 and -1 be expressed as $d=i*n+j$, where $j<n$. This means that the distance d can be reached by making i hops along the bi-directional ring of jumps $+n$ and $-n$ and j hops along the bi-directional ring of jumps +1 and -1.

It may be observed from Figure 3.8 that there exists bi-directional rings of size $(n+1)$ formed with n links of jumps +1 (or -1) and one link of jump $-n$ (or $+n$). Hence, the hop-distance between X and Y can be expressed as

$$h(XY) = \begin{cases} i + j & ,if \quad j \le n/2 \\ (i+1)+(j-n/2) & ,otherwise \end{cases} \qquad (3.3)$$

As the maximum value of d is $(n^2-1)/2$, the maximum value of i is $\lfloor (n^2-1)/(2*n) \rfloor < n/2$. Hence, the maximum value of $h(X,Y)$ is $(n/2)+(n/2)$ = n. (Since, the number of hops with jumps of magnitude 1 is ///2.)

Thus the diameter of CG-I topology, $C_{p,S}$ with $p=n^2-1$ and $S = \{-n,-1,+1,+n\}$, the maximum of hop-distance between all pair of nodes is n. •

// may be observed that the Circulant Graph-I, $C_{p,s}$ with $p=n^2-1$ and $S = \{-n,-1,+1,+n\}$ can be considered equivalent to 2-D Torus. This is because the CG-I topology also has a fixed node degree four and a diameter of $\sqrt{p+1}$, where p is the number of nodes in the network.

Having discussed the properties of base Circulant graph, we shall now define the Reconfigurable Circulant Graph-I (RCG-I). This reconfigurable topology is designed to preserve the properties of the base Circulant Graph-I, $C_{p,s}$ with $p=n^2-1$ and

Reconfigurable Circulant Graph-I. Let the Circulant Graph-I, $C_{p,s}$ with $p=n^2-1$ and $S = \{-n,-1,+1,+n\}$ be the base topology and q be any positive integer such that $q < p$. The *q-node Reconfigurable Circulant Graph-I (RCG-I)* with connection set $S = \{-n,-1,+1,+n\}$, denoted by $R_{q,s}$, consists of q nodes. Each node in $R_{q,s}$ has unique label in the range 0 through *p*-\ [Mohan Reddy and Reddy, 1996b]. Each node , is connected to *{i+j*s)%p* where $s \in S$ and./ be a positive integer such that for any positive integer $k < j$, nodes with labels *(i+k*s)%p* are failed.

Figure 3.9 shows an example of a 7-node RCG-I derived from a base CG-I topology, $C_{p,s}$ with $p=8$ and $S=\{-3,-1,+1,+3\}$ and with a failed node 4.

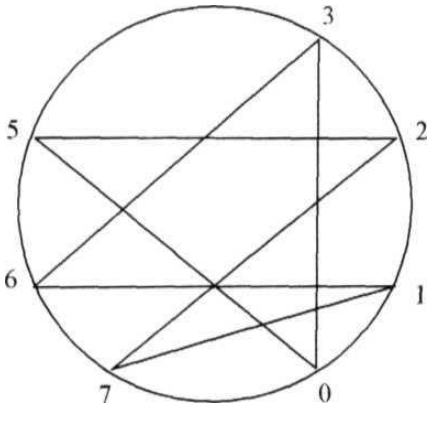


Figure 3.9 RCG-I with base topology $C_{p,s}$, $p=8$ *and* $S=\{-3,-1,+1,+3\}$ *and failed node 4.*

Reconfiguration in RCG-I.

The reconfiguration of RCG-I is explained as follows. Whenever a node fails, the neighboring nodes connect to the next node on the corresponding Hamiltonian circuit of jump *s* through which the failed node is connected. Similarly, when a node recovers from failure, it occupies its position in the base circulant graph. Algorithm 3.5, shown below, explains the computation of new neighbor nodes.

Algorithm 3.5. Finding a new neighbor node for node i *in the direction of jump* se S. p *is the size of the base circulant graph.*

```
for(m=1; m<=p; m++)
        if node with id (i\ m*s mod p)is active
                Set node (i + m *s mod p)as neighbor
                break
        endif.
endfor.
```

It may be observed that the above reconfiguration algorithm can be executed in a distributed manner to simultaneously result the new logical topology. Also the number of link changes is minimum that can be achievable by any reconfiguration algorithm.

To explain the process of reconfiguration in case of a node failure, consider a base topology shown in Figure 3.7. Assume that node 4 is failed. By connecting the neighbors of failed node to next nodes on the corresponding Hamiltonian circuits, we get the resultant topology as shown in Figure 3.9. Continuing the reconfiguration with failure of node 3 results in the structure shown in Figure 3.10.

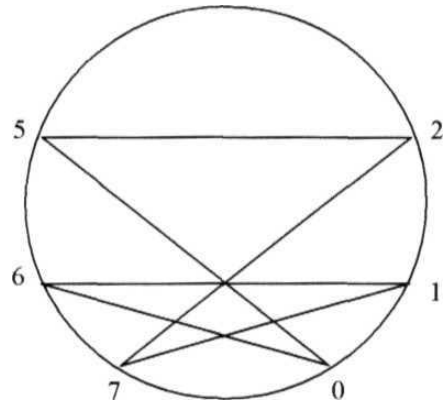


Figure 3. 10 RCG-I with base topology $C_{p,S}$ with p 8 and S={-3,-1, + 1,+3} and failed nodes 3,4.

To explain the process of node addition, consider RCG-I with six active nodes and two failed nodes 3 and 4 as shown in Figure 3.10. If node 3 wishes to enter the network, it informs the network and occupies its position in the base circulant graph by forming links with its neighbors, i.e., with nodes 2,5, 6, and 0. This results in the structure shown in Figure 3.9.

Routing in RCG-I.

A distributed routing method in RCG-I, $R_{q,S}$ with $q \leq p = n^2 - 1$ and $S = \{-n, -1, +1, +n\}$ is given in Algorithm 3.6. According to this algorithm, a packet is routed by making use of the rings present in the mesh-like representation of Circulant graph. To reach the destination node, a packet first tries to move along the Hamiltonian circuit of jump +1 when the reminder of distance on Hamiltonian circuit of jump +1 divided by *n* is not zero. Otherwise, the packet is routed along the bi-directional ring of jumps +n and -n.

To understand routing, consider the RCG-I, R_S with six active nodes and two failed nodes with ids 3 and 4, shown in Figure 3.10. For example, assume node 0 wants to send a packet to node 4. The packet is first routed to node 1 from here it learned that node 4 is not present. Node 1 then informs node 0 about the absence of node 4. As another example, assume that node 1 has a packet destined to node 7. Since the distance to node 7 is 6, the packet is routed on jump +3. Incidentally, the destination node 7 is the neighbor on this jump.

Let us see another example where node 0 has a packet destined to node 5. Though node 5 is a neighbor to node 0, the packet takes the route 0 to 7 to 2 to 5. This shows that the routing algorithm doesn't guarantee shortest path. By incorporating the knowledge about the neighbors, a better algorithm can be designed.

However, *it is not possible to design a shortest path routing algorithm for a reconfigurable topology that reconfigure using local perturbations because nodes doesn 't have the knowledge of the structure of the topology at a given point of time.*

But, packets are guaranteed to reach the destination node, if it is active. This is because any node can be reached on the Hamiltonian circuit of jump +1, provided it is active, and

this routing algorithm always tries, first, to move along the Hamiltonian circuit of jump +1.

Algorithm 3.6. Send a message from the current node C to the destination node D which is originated at source node S in a RCG $R_{q,S}$ with $q \le p = n^2$-and $S = \{-n,-1,+1,+n\}$.

```
if (C = D) then
        Send message to local node C
else
        // Compute the distance and jump to move on.
        Distance = (D-C+p) %p
        if((Distance % n = 0)
                if(Distance > p/2)
                        if((p-Distance) Magnitude of length of jump -n)
                                Send negative acknowledgement to the source S
                                indicating that the destination node D is failed
                        else
                                Move along jump -n.
                        endif
                else
                        if((Distance Magnitude of length of jump + n)
                                Send negative acknowledgement to the source S
                                indicating that the destination node D is failed
                        else
                                Move along jump + n.
                        endif.
                endif.
        else
                if((Distance Magnitude of length of jump +])
                        Send negative acknowledgement to the source S indicating
                        that the destination node D is failed
                else
                        Move along jump + 1.
                endif.
        endif.
endif
```

Diameter of RCG-l.

The maximum diameter of RCG-l is found to be *3n/2.* Theorem 3.6 proves this.

Theorem 3.6. The maximum diameter of RCG, $R_{q,S}$ with q<n -1 *and S* - $\{-n,-1,+1,+n\}$, *is less than 3n/2.*

Proof: According to theorem 3.5, the maximum hop-distance between any two arbitrary nodes X and Y, *h(X,Y)* is *n.*

As the nodes between X and Y keep failing, some of the nodes along jumps of magnitude *n* may not be accessible. This leads to movement in the bi-directional ring of jumps +1 and -1. Let us assume that Y is reached from X using Algorithm 3.6. Let *k* be the number of hops required to reach Y from X. Then, Y can be expressed in terms of the hops from X as

$$Y = \text{flf-t } h_1 + h_2 + \ldots + h_k)(n^2-1) \quad (3.4)$$

where h, is an integer multiple of +1, +*n* or -*n*.

Without loss of generality, we assume X < Y and (Y-X) < $(n^2-1)/2$. Then Y is reached from X using jumps +*n* and +1 and Y is expressed as sum of hops from X as,

$$Y = (X + h_1 + h_2 + \ldots + h_k) \quad (3.5)$$

Out of the *k* hops, let us assume that *m* hops are made along the bi-directional ring of jumps +*n* and -*n*. Then *m* < *n/2*. The remaining hops *(k-m)* are made along the Hamiltonian circuit of jump +1. Let us assume that $(k-m) > n$ Then, out of $(k-m)$ hops, there exists a, *b* and *c* such that

$$\sum_{i=a}^{b} h_i = c * n \quad (3.6)$$

and $$(b-a) > c > 1 \quad (3.7)$$

This is because, according to Hypothesis 3.3, there exists an image of a node in every *n* hops along jump +1 or –1. Thus hops h_a to h_b can be reduced to at most *c* hops along bi-directional ring of jumps +*n* and -*n*. By applying Hypothesis 3.3 repeatedly, the number of hops along the Hamiltonian circuit of jump +1, $k-m$, can be reduced to less than n However, the hops *m* along the bi-directional ring of jumps +*n* and -// remain less than $n/2$ due to the fact that (Y-X) < $(n^2-1)/2$.

Therefore, the number of hops, k, required to reach an arbitrary node Y from an arbitrary node X is less than 3n/2.

Thus the maximum diameter of RCG-I, $R_{q,S}$ with $q<n^2-]$ and $S = \{-n,-1,+1,+n\}$, i.e., the maximum of hop-distance between all pair of nodes is less than $3n/2$. •

Though we could prove the maximum diameter of RCG-I, $R_{q,S}$ with $q<n^2-\}$ and $S = \{-n,-1,+1,+n\}$ is less than 3/?/2, *it is observed empirically in Chapter 4 that the maximum diameter is n.* It is probably because $(k-m) > n/2$ jumps along Hamiltonian circuit of jump +1 can be replaced with one jump along the bi-directional ring $+n$ and $-n$ and $n-k+m$ jumps along the bi-directional ring +1 and -1. This is evident from the Figure 3.8 that *n* hops along the bi-directional ring of jumps -1 and +1 (or $-n$ and $+n$) can be replaced with one hop along the bi-directional ring of jumps $-n$ and $+n$ (or -1 and +1).

The values of n,p and S in RCG-I are selected not only to keep p as relatively prime with respect to the connection set S, but also to achieve the performance similar to a 2-D Torus structure. The facts that 2-D Torus structure gives maximum performance when the total number of nodes are n^2 and the links of a node are identified by jumps of length +1, -I, +n and -n, assuming (hat the nodes in a 2-D Torus are represented as integers instead of their position in the grid (row and column), motivated the design of RCG-I.

The successful design of this RCG-I provided an insight into construction of a base Circulant graph that is on par with n-D Torus. In the following section, we design a dynamically reconfigurable topology derived from a base Circulant graph that is equivalent to n-D Torus.

3.3.2 Reconfigurable Circulant Graph - II

This section presents another dynamically reconfigurable topology derived from the base Circulant graph $C_{p,S}$ with $p=k^n-1$ and $S=\{-k^{n-1},-k^{n-2},..,-1,+1,..+k^{n-2},+k^{n-1}\}$. We refer this base Circulant graph as ***Circulant Graph-II*** or ***CG-II*** and study its properties in order to prove the reconfigurable topology designed retains the structural properties. Circulant Graph-II is different from Circulant Graph-I in terms of connection set, *S* and the number of nodes,/?.

An example of the base Circulant graph $C_{p,S}$ with $p=26$ and connection set $S=\{-9,-3,-1,+1,+3,+9\}$ is shown in Figure 3.11.

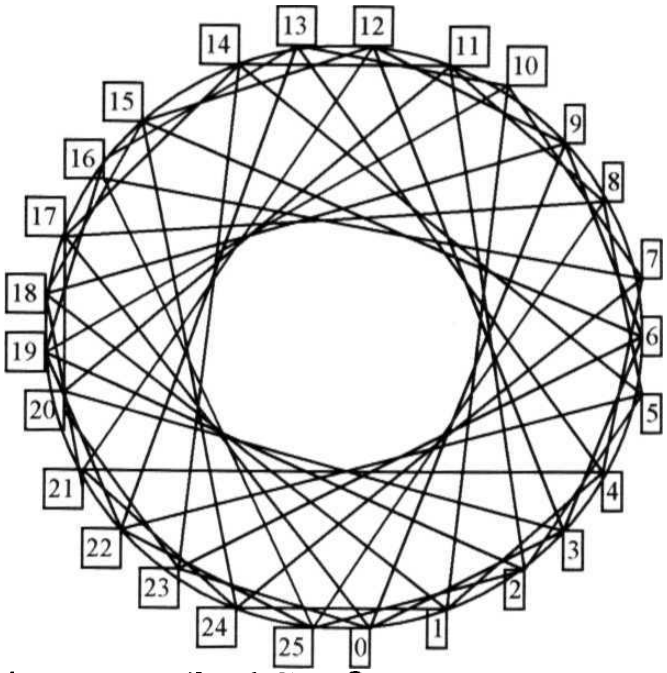


Figure 3.11 Circulant Graph-II, $C_{p\ s}$ *with p= 26and S= {-9,-3,-1, + 1, + 3, + 9}.*

By visualizing this graph in n-dimensional Euclidean space where each node is identified with an *n*-digit *k*-ary number, links of magnitude *k* are treated as moving in /-th dimension. First, let us define an image of node.

linage of a Node. X and Y are said to be images of each other in /-th dimension, if and only if X% k^{l} =

In the CG-I1 topology shown in Figure 3.11, node 0,3 and 6 are said to be images of each other. Similarly, nodes 10 and 19 are said to images of each other. It may be observed from Figure 3.11 that an image of node along the jumps + *k* or - *k* can be reached in a maximum of *k* hops.

Hypothesis 3.4. In the Circulant Graph-II, $C_{p\,s}$*, an image of a node along the jumps* t k^{l} *and* -k^{l} *can be reached in a maximum of* k*hops.*

•

It may be observed from the Figure 3.11 that there exist six edge-disjoint Hamiltonian circuits, each one being associated with a jump in the connection set *S.* Corollary 3.2 proves the existence of such Hamiltonian circuits in the base Circulant graph.

*Corollary 3.2. There are 2*n edge-disjoint Hamiltonian circuits in CG-II,* $C_{p,S}$ *with* $p=k^n-1$ *and* $S=\{-k^{n-1},-k^{n-2},..,-1,+1,..,+k^{n-2},+k^{n-1}\}$.

Proof: According to Hypothesis 3.2, since all the jumps in the connection set S are distinct, it is sufficient to show that the paths associated with jumps are Hamiltonian circuits. Further, according to Theorem 3.4, it is sufficient to prove that all these jumps are relatively prime with respect to *p.*

+1 and -1 are relatively prime with respect to/?. Suppose k', $0\le i<n-1$, is not relatively prime with respect to p. Then p/k' is an integer. But $p/k'=(k^n-1)/k'$ is not an integer. This contradicts our earlier assumption. Therefore, k' has to be relatively prime with respect to *p.* By similar argument, we can show that $-k'$ is relatively prime with respect to $p= k''-\backslash$.

Thus all the jumps of a connection set *S* are relatively prime with respect to p. Hence, the Hamiltonian circuits associated with 2*// different jumps are edge disjoint. •

Routing in base Circulant Graph-II.

A distributed shortest path routing method in the Circulant Graph-II $C_{p,S}$ with $p=k^n-1$ and $S=\{-k^{n-1},-k^{n-2},..,-1,+1,..,+k^{n-2},+k^{n-1}\}$ is given in Algorithm 3.7.

Algorithm 3.7. Send a message from the current node C to the destination node D which is originated at source node S in a Circulant Graph-II, $C_{p,S}$ *with* $p=k^n-1$ *and connection set* $S=\{-k^{n-1},-k^{n-2},..,-1,+1,..,+k^{n-2},+k^{n-1}\}$.

```
if (C = = D)then
        Send message to local node C
else
        // Compute the distance and jump to move on.
        Distance - (D-C+p) %p
        jump = 1
        while (Distance % (jump*k) = = 0)
                jump -jump * k
        endwhile.
        if(Distance % (jump*k) > (jump*k/2))
                jump = (-1) * jump
        endif.
endif.
```

According to this algorithm, a packet is routed by making use of the bi-directional rings present in the Circulant Graph-II. It may be observed that Hamiltonian circuits

associated with +1 and -1 jumps form a bi-directional ring. Similarly, Hamiltonian circuits associated with $+k^i$ and $-k^i$ jumps, 0<i<n-l, also form bi-directional rings. To reach the destination node, a packet first moves along the bi-directional ring of jumps -1 and +1, then along the bi-directional ring of jumps +k and -k, then along the bi-directional ring of jumps $+k^2$ and $-k^2$, and so on.

Diameter of base Circulant Graph-II.

The diameter of Circulant Graph-II is found to be *n*k/2*. Theorem 3.7 proves this.

Theorem 3.7. The diameter of Circulant Graph-II $C_{p,S}$ with $p = k^n - 1$ *and connection set* $S=\{-k^{n-1},-k^{n-2},..,-1,+1,..,+k^{n-2},+k^{n-1}\}$ *is* $n*k/2$.

Proof: The maximum hop-distance between any two arbitrary nodes along the bi-directional ring of jumps +1 and -1 is $(k^n-1)/2$. The hop-distance is further reduced by making use of jumps of higher magnitude, i.e., $+k,-k,+k^2,-k^2,\ldots,+k^{n-1}$ and $-k^{n-1}$.

Let *h* be the hop-distance between two arbitrary nodes X and Y, on bi-directional ring of jumps +1 and -1. Let *h* be expressed in k-ary number system as $h=h_{n-1}h_{n-2}\ldots h_0$, where $0\le h_i<k$, $V\backslash<n$. Each bit $h_i\neq 0$, i<n in h can be reduced to zero by moving along the bi-directional ring of $+k'$ and $-k'$, in a maximum of $k/2$ hops. Thus, all the bits of h can be reduced to zero, i.e., the distance *h* can be reached in a maximum of $n*k/2$hops.

Hence, the diameter of CG-I1, $C_{p,S}$, with $p=k^n-1$ and $S=\{-k^{n-1},-k^{n-2},..,-1,+1,..,+k^{n-2},+k^{n-1}\}$, the maximum of hop-distance between any pair of nodes is *n*k/2.* •

It may be observed that the Circulant Graph-II, $C_{p,S}$ with $p=k^n-1$ and $S=\{-k^{n-1},-k^{n-2},..,-1,+1,..,+k^{n-2},+k^{n-1}\}$ can be considered at par with n-D Torus. This is because the Circulant graph $C_{p,S}$ also has a node degree of *2*n* and a diameter of $n*k/2$, where $p=k^n-1$ is the number of nodes in the network.

Having been discussed the properties of base Circulant Graph-II, we shall now define the reconfigurable topology, Reconfigurable Circulant Graph-II (RCG-II). This topology is designed to preserve the properties of the base Circulant Graph-II.

Reconfigurable Circulant Graph-II. Let the Circulant Graph-II, $C_{p,S}$ with $p=k^n-1$ and $S=\{-k^n, -k^{n-2}, .., -1, +1, .., +k^{n-2}, +k^{n-1}\}$ be the base topology and q be any positive integer such that $q < p$. The *q-node Reconfigurable Circulant Graph (RCG-Il)* with connection set $S=\{-k^{n-1}, -k^{n-2}, .., -1, +1, ., +k^{n-2}, +k^{n-1}\}$, denoted by $R_{q,S}$, consists of q nodes. Each node in $R_{q,S}$ has unique label in the range 0 through p-\. Each node i is connected to $(i+j*s)\%p$ where $s \in S$ and j be a positive integer such that for any positive integer $k < j$, nodes with labels $(i+k*s)\%p$ are failed.

Figure 3.12 shows an example of a 25-node RCG-II derived from a base Circulant Graph-II, $C_{p,S}$ with p=26 and $S=\{-9,-3,-1,+1,+3,+9\}$ and with a failed node 4.

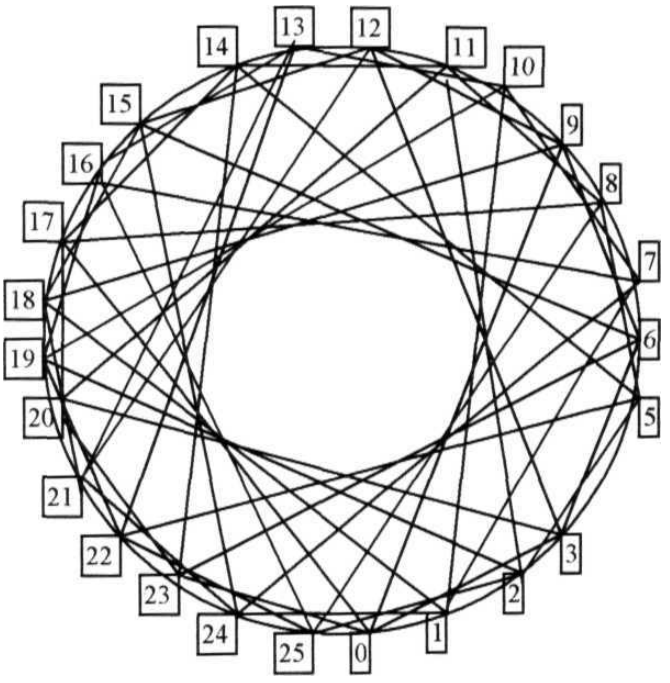


Figure 3.12 RCG-II with base topology $C_{p,S}$, $p=$ and $S=\{-9,-3,-1,+1,+3,+9\}$ and failed node 4.

Reconfiguration in RCG-II.

The reconfiguration of RCG-II is similar to that of RCG-I, as described in Algorithm 3.5. Whenever a node fails, the neighboring nodes connect to the next node on the corresponding Hamiltonian circuit of jump s through which the failed node is connected.

Similarly, when a node recovers from failure, it occupies its position in the base circulant graph.

To explain the process of reconfiguration in case of a node failure, consider a base topology shown in Figure 3.11. Assume that node 4 is failed. By connecting the neighbors of failed node to next nodes on the corresponding Hamiltonian circuits, we get the resultant topology as shown in Figure 3.12. Continuing the reconfiguration with failure of node 3 results in the structure shown in Figure 3.13.

To explain the process of node addition, consider RCG-II with six active nodes and two failed nodes 3 and 4 as shown in Figure 3.13. If node 3 wishes to enter the network, it informs the network and occupies its position in the base topology by forming links with its neighbors, i.e., with nodes 2, 5, 6, 0, 12 and 23. This results in the structure shown in Figure 3.12.

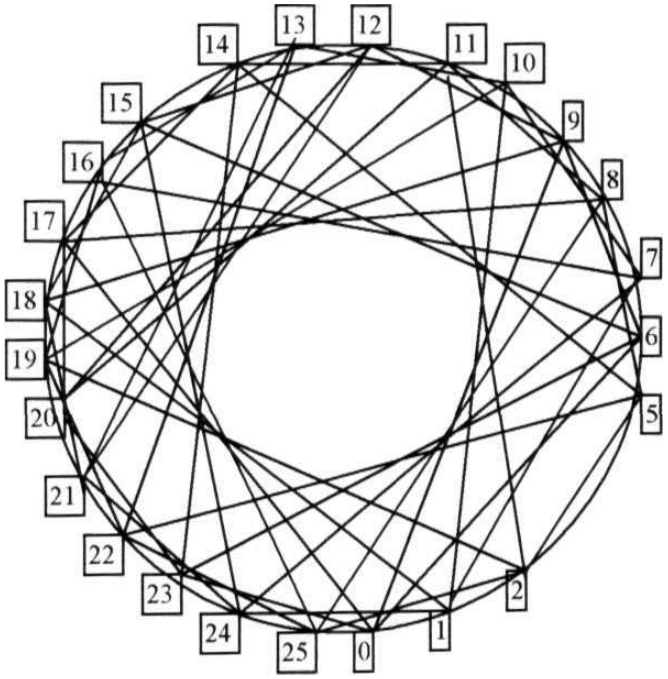


Figure 3.13 RCG-II with base topology $C_{p,S}$ *with* $p=25$ *and* $S=\{-9,-3,-1,+7,+13,+9\}$ *and failed nodes 3,4.*

Routing in RCG-II.

A distributed routing method in RCG-II $R_{q,S}$ with $q \leq p = k^n - 1$ and $S=\{-k^{n-1},-k^{n-2},..,-1,+1,..,+k^{n-2},+k^{n-1}\}$ is given in Algorithm 3.8. According to this

algorithm, a packet is routed along a bi-directional ring of jumps $+k$ and $-k$ for which the reminder of distance divided by k is not zero and there exists no other bi-directional ring of jumps $+k$ and $-k^j$, $j<i$, for which the reminder is not zero.

Algorithm 3.8. Send a message from the current node C to the destination node 1) which is originated at source node S in a RCG-I1, $R_{q,s}$ *with* $q \leq p\ k^n-1$ *and* $S=\{-k^{n-1}, -k^{n-2}, \ldots, -1, +1, \ldots, +k^{n-2}, +k^{n-1}\}$.

```
if (C == D) then
        Send message to local node C
else
        // Compute the distance and jump to move on.
        Distance = (D-C+p) %p
        jump = 1
        while (Distance % (jump*k) =~ 0)
                jump = jump * k
        endwhile.
        if ((Distance % (jump*k) (jump*k/2)) && (⌊Distance/(jump*k)⌋/0) )
                jump = (-1) * jump
                if((p-Distance) magnitude of length of jump)
                        Send negative acknowledgement to the source S indicating
                        that the destination node D is failed
                else
                        Move along jump.
                endif.
        else
                if((Distance magnitude of length of jump)
                        Send negative acknowledgement to the source S indicating
                        that the destination node D is failed
                else
                        Move along jump.
                endif
        endif.
endif.
```

Routing is explained by considering the RCG-II with 25 active nodes and one failed node with id 4, shown in Figure 3.12. For example, assume a packet is originated at node 3 to node 7. Since the distance is 4, the packet is routed along the Hamiltonian circuit of jump +1, to node 5. From there it continues on the Hamiltonian circuit of jump +1 to node 6 and then finally to the destination node 7. It may be observed that there exists a shorter route 3 to 6 to 7.

The routing algorithm doesn't guarantee to route a packet through shortest path. However, packets are guaranteed to reach the destination node, provided it is active. This is because the packet continues to move along the Hamiltonian circuit of jump *k'* as long as the reminder of the distance on the Hamiltonian circuit of jump +1 divided by $k^{(i+1)\%n}$ is not zero.

Diameter of RCG-II.

The maximum diameter of RCG-II is less than $\lceil (n-1)(k-1) \rceil$. Theorem 3.8 gives the proof.

Theorem 3.8. The maximum diameter of RCG-II, $R_{q,S}$ *with* $q \le k^n - 1$ *and connection set* $S=\{-k^{n-1}, -k^{n-2}, \ldots, -1, +1, \ldots, +k^{n-2}, +k^{n-1}\}$ *is less than* $\lceil (n-1)(k-1) \rceil$.

Proof: Let d = $(Y = X + k'' - 1)\%(k^n - 1)$ be the distance between two arbitrary nodes X and Y on the Hamiltonian circuit of jump +1. This distance can be expressed in *k*-ary number system as

$$d = d_{n-1} d_{n-2} \ldots d_0 \qquad (3.8)$$

where $d_i < k$. We can, without loss of generality, presume $d_i < k/2$ because if $d_i > k/2$, the distance $d_i k^i$ can be reached using one hop along jump $k^{(i+1)\%n}$ and *(k-d,)* hops using jump $-k^i$.

Ideally this distance can be reached, according to theorem 3.7, in a maximum of n*k/2 hops. However, as the nodes between X and Y keep failing, some of the nodes along jumps of magnitude k^i, i>0, may not be accessible. This leads to movement in the bi-directional ring of jumps of magnitude k^j, j<i.

Let us assume that Y is reached from X, by making hops according to Algorithm 3.8, in *m* hops. Then Y can be expressed as sum of hops from X as

$$Y = (X + h_1 + h_2 + \ldots + h_m)\% (k^n - 1) \qquad (3.9)$$

where h_i is an integer multiple of k^j, $0 \le j \le (k-1)$.

Without loss of generality, we assume X < Y. Then,

$$Y-(X+h_1+h_2+\ \ldots\ +h_m) \qquad (3.10)$$

Let h_j h_{j+1} are the hops along jumps k^a and k^b in the sequence of m hops. Then $a<b$. This is because, according to Algorithm 3.8, a packet is routed along jump k^l, where l is the lowest dimension in the binary representation of distance such that d_l

Out of these m hops, let the hops along jump k^l, r, is greater than or equal to k. Then, there exists c, d and e such that,

$$\sum_{j=c}^{d} h_j = ek^{(i+1)\%n} \qquad (3.11)$$

According to Hypothesis 3.4, there exists an image of a node along jumps of magnitude k^l in every k hops. Thus the hops along jump k^l, r, can be reduced to $(r-k)$ hops. By applying Hypothesis 3.4 repeatedly, to all jumps k^l for which the hops are greater than k, it reduces the number of hops to less than k.

However, the maximum hops along the bi-directional ring of jumps $+k^{n-1}$ and $-k^{n-1}$ is less than or equal to k/2, because Algorithm 3.8 makes use of the bi-directional ring of jumps $+k^{n-1}$ and $-k^{n-1}$.

Therefore, an arbitrary node Y can be reached from an arbitrary node X in a maximum of $\frac{k}{2}+(n-1)(k-1)$ hops, that is, the sum of maximum of hops along bi-directional ring of jumps $+k^{n-1}$ and $-k^{n-1}$ and the maximum hops along the Hamiltonian circuits of jumps $+k$, V/, $0\le i<n-1$.

Thus the maximum diameter of RCG-I1, $R_{q,S}$ with $q<k^n-1$ and $S=\{-k^{n-1},-k^{n-2},..,-1,+1,..,+k^{n-2},+k^{n-1}\}$ which is the maximum of hop-distance between all pair of nodes is less than — + (A? - 1)$(k-1)$ ■

Though it is proved that the maximum diameter of RCG-1I is less than - + $(n-1)(-1)$, it is observed empirically that the maximum diameter is $kn/2$ (refer Chapter 4). It is

probably because $r>kl2$ hops along bi-directional ring $+k$ and $-k$ can be replaced with one jump along the bi-directional ring $+k^{l+1}$ and $-k^{l+1}$ and $(k-r)$ jumps along the bi-directional ring $+k$ and $-k$.

The value k is selected such that it is n^{th}-root of the number of nodes in the base Circulant Graph-II, $p=k^n-1$.

This way of selecting k, n and p is motivated by the fact that the links of a node in a n-D Torus are identified by jumps of length $+1$, -1, $+k$, $-k$, ... , $+k^{n-1}$ and $-k^{n-1}$, assuming that the nodes are represented as integers instead of their position in the n-dimensional Euclidean space.

The successful design of RCG-II provided an insight to construct a base Circulant graph whose performance is equivalent to that of Binary Hypercube. Following section presents the design of a dynamically reconfigurable topology derived from this base topology.

3.3.3 Reconfigurable Circulant Graph - III

This section presents another reconfigurable topology derived from the base Circulant graph $C_{p,S}$ with $p=2^n-1$ and $S=\{1,2,\ldots,2^{n-2},2^{n-1}\}$. We refer this base Circulant graph as ***Circulant Graph-Ill*** or ***CG-III***. Let us first see the properties of the base CG-I topology in order to prove that the resultant reconfigurable topology retains the structural properties of the base topology. CG-I1I is different from the other two Circulant graphs, CG-I and CG-II, in terms of connection set, S and the number of nodes, p.

An example of the base Circulant Graph-III, $C_{p,S}$ with $p=7$ and connection set $S=\{1,2,4\}$ is shown in Figure 3.14.

It may be observed from the Figure 3.14 that there exists three edge-disjoint Hamiltonian circuits, each one being associated with a jump in the connection set S. Corollary 3.3 proves the existence of such Hamiltonian circuits in the base Circulant graph.

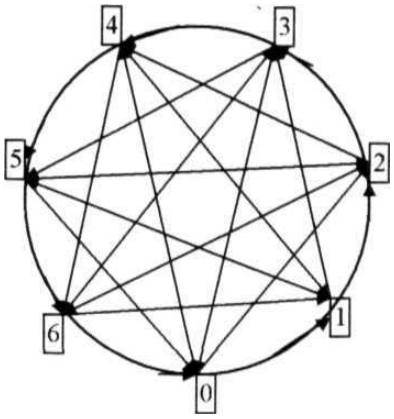


Figure 3.14 Circulant Graph-Ill $C_{p,S}$ *with p=7 and S {1,2,4}.*

Corollary 3.3. There are n edge-disjoint Hamiltonian circuits in $C_{p,S}$ *with p 2"-l and S* $\{1,2,\ldots,2^{n-2},2^{n-1}\}$.

Proof: According to Hypothesis 3.2, since all the jumps in the connection set S are distinct, it is sufficient to show that the paths associated with jumps are Hamiltonian circuits. Further, according to Theorem 3.4, it is sufficient to prove that all these jumps are relatively prime with respect to *p*.

1 is relatively prime with respect to *p*. Suppose 2', $0 \le i < n-1$, is not relatively prime with respect to *p*. Then *p/2'* should be an integer. But *p/2'=(2"-])/2'* is not an integer. This contradicts our assumption. Therefore, 2' has to be relatively prime with respect to *p*.

Thus all the jumps of connection set *S* are relatively prime with respect to *p*. Hence, the Hamiltonian circuits associated with *n* different jumps are edge disjoint. •

Routing in base Circulant graph-III.

A distributed shortest path routing method in the CG-II1 topology is described in Algorithm 3.9. According to this algorithm, packets are routed by making use of the Hamiltonian circuits present in the Circulant graph. To reach the destination node, a packet first moves along the Hamiltonian circuit of jump 1, then along the Hamiltonian circuit of jump 2, then along the Hamiltonian circuit of jump 2^2, and so on.

Algorithm 3.9. Send a message from (he current node C to the destination node D which is originated at source node S in a Circulant Graph-II, $C_{p,S}$ *with* $p=2^n-1$ *and connection set* $S=\{1,2,\ldots,2^{n-2},2^{n-1}\}$.

```
if (C = D) then
        Send message to local node C
else
        // Compute the distance and jump to move on.
        Distance (D-C + p) % p
        jump  1
        while (Distance % (jump* 2) 0)
                jump = jump * 2
        endwhile.
endif.
```

Diameter of base Circulant Graph-Ill.

The diameter of Circulant Graph-Ill is found to be *n.* The following theorem proves this.

Theorem 3.9. The diameter of Circulant graph $C_{p,S}$ *with* $p=2^n-1$ *and connection set* $S=\{1,2,\ldots,2^{n-2},2^{n-1}\}$ *is* n.

Proof: The maximum hop-distance between any two arbitrary nodes along the Hamiltonian circuit associated with jump 1 is *(2"-I)/2.* This hop-distance is further reduced by making use of jumps of higher magnitude, i.e., $2,2^2,\ldots,2^{n-1}$.

Let *h* be the hop-distance between two arbitrary nodes X and Y along the Hamiltonian circuit associated with jump 1. Let h be expressed in binary number system as

$$h=h_{n-1}h_{n-2}\ldots h_0$$

where $h_i=0$ or 1, Vi<n. Each bit $h_i\neq 0$, $i<n$ in *h* can be reduced to zero by moving along the jump 2'. Thus, all the bits of h can be reduced to zero, i.e., the distance *h* can be reached in a maximum of// hops.

Hence, the diameter of $C_{p,S}$ with $p=2^n-1$ and $S=\{1,2,\ldots,2^{n-2},2^{n-1}\}$, the maximum hop-distance between any pair of nodes is *n.* •

It may be observed that the Circulant graph $C_{p,S}$ with $p=2^n-1$ and $S=\{1,2,\ldots,2^{n-2},2^{n-1}\}$ can be considered equivalent to binary Hypercube. This is because the Circulant graph $C_{p,S}$

also has a node degree of n and a diameter of//, where $p=2^n-1$ is the number of nodes in the network.

Having seen the properties of base Circulant Graph-Ill, we will now define the Reconfigurable Circulant Graph-Ill (RCG-III). This reconfigurable topology is designed to preserve the properties of the base Circulant Graph-Ill, $C_{p,S}$ with $p=2^n-1$ and $S=\{1,2,\ldots,2^{n-2},2^{n-1}\}$.

Reconfigurable Circulant Graph-Ill. Let the Circulant Graph-Ill, $C_{p,S}$ with $p=2^n-1$ and $S=\{1,2,\ldots,2^n,2^{""}\}$ be the base topology and q be any positive integer such that $q < p$. The *q-node Reconfigurable Circulant Graph-Ill (RCG-III)* with connection set $S=\{1,2,\ldots,2^{n-2},2^{n-1}\}$, denoted by $R_{q,S}$, consists of q nodes. Each node in $R_{q,S}$ has unique label in the range 0 through p-\. Each node i is connected to $(i+j*s)\%p$ where $s \in S$ and j be a positive integer such that for any positive integer $k < j$, nodes with labels $(i+k*s)\%p$ are failed.

Figure 3.15 shows an example of a 6-node RCG-III derived from a base CG-III topology, $C_{p,S}$ with p=7 and $S=\{1,2,4\}$ and with a failed node 4.

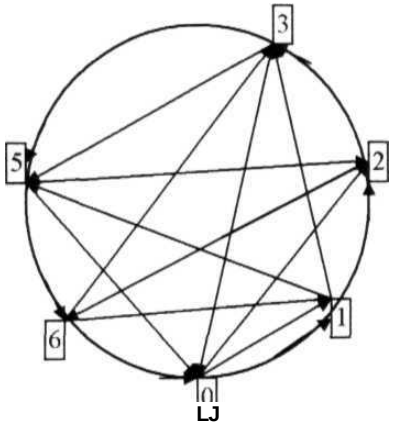


Figure 3.15 RCG-III with base topology $C_{p,S}$, p 7 and $S=\{1,2,4\}$ and failed node 4.

Reconfiguration in RCG.

The reconfiguration of RCG-III is similar to that of the RCGs described earlier using the Algorithm 3.5. Whenever a node fails, the neighboring nodes connect to the next node on the corresponding Hamiltonian circuit of jump s through which the failed node is

connected. Similarly, when a node recovers from failure, it occupies its position in the base circulant graph.

To explain the process of reconfiguration in case of node failure, consider a base topology shown in Figure 3.14. Assume that node 4 is failed. By connecting the neighbors of failed node to next nodes on the corresponding Hamiltonian circuits, we get the resultant topology as shown in Figure 3.15. Continuing the reconfiguration with failure of node 3 results in the structure shown in Figure 3.16.

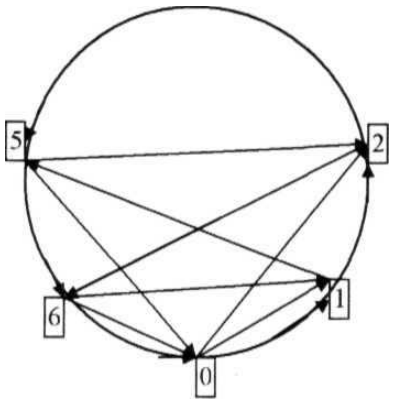


Figure 3.16 RCG-III derived from base topology $C_{p,S}$ with p 7 and S= {1,2,4} and failed nodes 3,4.

To explain the process of node addition, consider RCG-III with six active nodes and two failed nodes 3 and 4 as shown in Figure 3.16. If node 3 wishes to enter the network, it informs the network and occupies its position in the base circulant graph by forming links with its neighbors, i.e., with nodes 5 and 7. This results in the structure shown in Figure 3.15.

Routing in RCG-III.

A distributed routing method in RCG-III $R_{q,S}$ with $q \leq p=2^n-1$ and $S=\{1,2,\ldots,2^{n-2},2^{n-1}\}$ is described in the Algorithm 3.10. According to this algorithm, a packet is routed by making use of the Hamiltonian circuits present in the base Circulant Graph-Ill, each time reducing the distance of the destination along the Hamiltonian circuit of jump 1 To reach the destination node, a packet moves, starting from jump 1, along the jump 2^1 where the /-bit in binary representation of distance is 1

Algorithm 3.10. Send a message from the current node C to the destination node D which is originated at source node S in a RCG-III, $R_{q,s}$ with $q \leq p$ 2"-l and $S = \{1,2,\ldots,2^{n-2},2^{n-1}\}$.

```
if (C == D) then
        Send message to local node C
else
        // Compute the distance and jump to move on.
        Distance = (D-C+p) %p
        jump - 1
        while(Distance % (jump*2) = - 0)
                jump - jump * 2
        endwhile.
        if( (Distance  Length of jump)
                        Send negative acknowledgement to the source S indicating
                        that the destination node D is failed
        else
                        Move along jump.
        endif.
endif.
```

For better understanding of routing in RCG-III, consider the topology shown in Figure 3.15. The topology has 6 active nodes and one failed node with id 4. Assume that node 3 wants to send a packet to node 1. The distance 5 = (1-3+7)%7 is reached by making hop along jump1 to node 5. From node 5, where the distance is calculated as 3, the packet is moved along jump 1 to node 6. At node 6, the distance is calculated as 2 and the packet is routed along jump 2 to the destination node 1. Though the packet is reached the destination, this is not the shortest path from node 3 to node 1. However, this routing algorithm guarantees that the destination node can be reached, provided it is active. This is because the algorithm reduces the distance along the Hamiltonian circuit of jump 1, with each hop.

Diameter of RCG-III.

The maximum diameter of **RCG-III** is found to be *n.* Theorem 3.10 gives the proof.

Theorem 3.10. The maximum diameter of RCG-III, $R_{q,s}$ with $q \leq 2^n$-*1 and connection set* $S = \{1,2,\ldots,2^{n-2},2^{n-1}\}$ *is* n.

Proof: According to theorem 3.9, the maximum hop-distance between any two arbitrary nodes X and *Y,h(X,Y)* is *n.*

As the nodes between X and Y keep failing, some of the nodes along the Hamiltonian circuit of jump 2', />0, may not be accessible. This leads to movement along the Hamiltonian circuit of jump 2^{i-1}. Let us assume that Y is reached from X in *k* hops, each hop reducing the distance on the Hamiltonian circuit using Algorithm 3.10 Then Y can be expressed as sum of hops from X as

$$Y = (X + h_1 + h_2 + \ldots + h_k) \% (2^n - 1) \qquad (3.12)$$

Without loss of generality, we assume X < Y. Then,

$$Y = X + h_1 + h_2 + \ldots + h_k \qquad (3.13)$$

where h_i is an integer multiple of 1,2,..., 2^{n-2}, or 2^{n-1}. Let h, and h_{i+1} are two adjacent hops in the above sequence along the jumps 2^a and 2^b. Then, according to Algorithm 3.10, $a<b$.

If k is greater than n, then there exist two or more hops along jump 2', for some $0 < i < n$. Let h_c and h_d are the starting and ending hop along the jump 2', in the above sequence of *k* hops. Then, hops between h_c and h_d can be represented as

$$h_j = e_j 2^i \qquad (3.14)$$

where e_j is called as the coefficient of hop h_j.

The hops from h_c to h_d can be replaced with/hops of jump $2^{(i+1)\%n}$ such that

$$f < (d - c)$$

This is because the distance $2^{(i+1)\%n}$ can be reached in at most two hops of jump 2' and there exists at least *(d-c)* hops whose sum of the coefficients is an even number.

Thus, by replacing the hops from h_c to h_d with f hops of, there may exist at most one hop of jump 2'.

By repeating this procedure for all jumps, starting with jump 1, we can find a path with a maximum of// hops.

Hence, the number of hops required to reach an arbitrary node Y from an arbitrary node X, k < n. In other words, the maximum diameter of RCG-11I, $R_{q,s}$ with $q \leq 2^n - 1$, the maximum of hop-distance between all pair of nodes is *n.* •

So far we have studied the design of dynamically reconfigurable topologies whose base topologies are constructed with edge-disjoint Hamiltonian circuits. It is observed that these topologies have comparable performance with 2-D Torus, n-D Torus and Binary Hypercube.

Though the base Circulant Graphs are constructed using a set of edge-disjoint Hamiltonian circuits, it is enough if the base topology has one Hamiltonian circuit for maintaining connectivity.

3.4 Discussion

The reconfigurable logical topologies proposed in this chapter ~ Perturbed Torus, Reconfigurable Circulant Graphs that are equivalent to 2-D Torus, n-D Torus and Binary Hypercube — consider reconfiguration as part of the topology design issue. *These topologies reconfigure using Local Perturbation paradigm proposed as part of this thesis.* For accommodating changes (failure and addition of nodes) in network, these topologies assume an initial well defined regular structure called base topology. Addition or deletion of node is considered as assigning or un-assigning a position in the base topology. Hence, a reconfigurable topology moves closer or away from the base topology depending upon whether a node is added or deleted from the network.

Applicability of the reconfiguration method based on Local Perturbation to the traditional topologies such as Torus and Hypercube is investigated in Section 3.2. From the results obtained, it is shown that successive application of reconfiguration based on Local Perturbations for node failures disconnects the topology. To retain connectivity, additional steps must be devised into the reconfiguration process. Perturbed Torus is designed by maintaining a base row (column) in the base topology. However, it is not always possible to define such special methods to retain connectivity.

By adopting the concept of Hamiltonian circuit, wherein every node in the network is visited exactly once, a set of reconfigurable topologies is proposed. The idea is to define reconfiguration along Hamiltonian circuits so that the network is always connected. In Section 3.3, we designed a set of topologies, called Reconfigurable Circulant Graphs, derived from a subset of Circulant graphs. Incidentally, all these topologies have edge disjoint Hamiltonian circuits. However, it is enough if the base topology has one Hamiltonian circuit for maintaining connectivity.

As per the framework proposed earlier, for evaluation of topologies, here, we will evaluate the reconfigurable topologies proposed in this thesis. Table 3.1 summarizes the node degree and diameter properties of the dynamically reconfigurable topologies. It is observed that Perturbed Torus and Reconfigurable Circulant Graph - 1 (RCG-I) have same node degree and number of nodes as that of 2-D Torus and Circulant Graph - 1. Similarly, RCG-T1 has the same node degree and number of nodes as that of Circulant Graph - II and Multidimensional Torus. RCG-III has the same node degree and number of nodes as that of Circulant Graph - III and Binary Hypercube.

As the reconfigurable topologies change their structure with every reconfiguration, diameter is measured as the maximum of diameter for all different structures that can be derived from the base topology. Since the reconfiguration is done to retain routing properties, usually the maximum diameter of a reconfigurable topology is limited to the diameter of the base topology. We proved the maximum diameter of Perturbed Torus is almost same as 2-D Torus. Though the maximum diameter obtained for RCG-I and RCG-II is more than that of corresponding base topologies, the empirical results obtained in the next chapter supports our argument.

// *may not be possible to formulate the computation of average internode distance for these topologies as the structure of reconfigurable topologies change with every reconfiguration.* In chapter 4, we evaluate average internode distance empirically by assuming different structures and compare them with corresponding base structures and their counterparts in traditional topologies.

Topology	No. of Nodes	Node Degree	Diameter
Binary Hypercube	$N=2^n$	n	n
Multidimensional Torus	$N= k^n$	2*n	k*n/2
Perturbed Torus	$N=n^2$	4	$2*\lfloor \lceil \sqrt{N} \rceil /2 \rfloor +1$
2-D Torus	$N=n^2$	4	n
RCG-I	$N=n^2-1$	4	3n/2
Circulant Graph - I	$N=n^2-1$	4	n
RCG-II	$N= k^n-1$	2*n	$\frac{k}{2}+(n-1)(k-1)$
Circulant Graph – II	$N= k^n-1$	2*n	k*n/2
RCG-II1	$N= 2^n-1$	n	n
Circulant Graph - III	$N= 2^n-1$	n	n

Table 3.1 Comparison of Reconfigurable topologies with their counterparts.

Designing shortest path routing algorithms is also not possible for reconfigurable topologies because the nodes in these networks don 7 have the idea about the structure of the topology at any given point of time. In other words, because the nodes wouldn't have the knowledge of which positions in the base topological structure are vacant, it is not possible to design a shortest path routing algorithm. It is also observed that a little overhead will be introduced into the routing algorithms of reconfigurable topologies when compared to the routing in corresponding base topologies.

The main advantage of the proposed reconfigurable topologies when compared with traditional topologies such as deBruijn graph, MS Net and Binary Hypercube is the dynamic reconfiguration. *Dynamic reconfiguration is achieved with minimum disturbance to the network.*

Chapter

4. Evaluation of Dynamically Reconfigurable Logical Topologies

In this chapter, the performance of newly proposed dynamically reconfigurable topologies are evaluated for both linear and non-linear traffic conditions. Section 4.1 briefly discusses how the logical topologies are evaluated for linear and non-linear traffic conditions. Section 4.2 presents heuristic algorithms for optimal node placement. These algorithms are used in the evaluation of reconfigurable topologies under non-uniform traffic conditions. Section 4.3 discusses the performance of each topology designed in Chapter 3 for both linear and non-linear traffic conditions and compares them with respective base topologies. Section 4.4 summarizes the results obtained.

4.1 Introduction

Regular topologies are usually designed to have fixed node degree, smaller diameter and low average internode distance. In addition, regular topologies are designed to possess simple routing algorithms and hence, they are well suited for uniform traffic patterns. *The performance of regular topologies under uniform traffic conditions is measured in terms of the topological properties such as diameter and average internode distance.* In section 4.3, the empirical results of diameter and average internode distance of the proposed reconfigurable topologies are tabulated and compared with respective base topologies.

Traffic in real life is non-uniform; hence performance study of regular topologies under non-uniform traffic conditions is also discussed. Topologies are designed to meet non-uniform traffic demands in [Bannister et. al., 1990; Labourdette and Acampora, 1991]. These solutions treat topology design as combinatorial optimization problem and provide irregular topologies as heuristic solutions.

For regular topologies, optimization is studied as placement of nodes in a given regular topology. [Banerjee and Mukherjee, 1993] studied optimal node placement for regular topologies and provided heuristic solutions. The following section presents the optimal node placement problem as a combinatorial optimization problem and proposes heuristic solutions.

4.2 Optimal Node Placements

The optimal node placement problem may be stated as follows: Given *N* nodes in a network must be connected according to some regular logical topological structure and the positions that these nodes occupy in the logical structure may be adjusted by properly tuning their transmitters and receivers, what is the best way for placing them? In other words, *given the flexibility of interconnecting the nodes, one needs to construct a regular topology that satisfies an optimality criterion such as minimization of the flow-weighted average (hop) internode distance and minimizing the maximum load over any link.* This work presents the minimization of the flow-weighted average internode distance. The minimization problem can be stated mathematically as follows;

Given:

TV = number of nodes in the network.

$d(i,j)$ = hop distance between position i and position j in the logical topology.

D_{NxN} = distance matrix measured for all pairs of node positions in a given logical topology.

$t(m,n)$ = the average rate of traffic flow (in packets per second) between node m and node n.

$T_{N\ N}$ = traffic matrix indicating traffic between all pairs of nodes in the network.

Variables:

Position of node m, $P(m)$ - $i<N$, for all $m<N$.

Objective: Optimality Criterion

$$\textbf{Minimize: } \sum_{m=0}^{N-1}\sum_{n=0}^{N-1} t(m,n) * d(i,j)$$

where /= $P(m)$ **and**/= $P(n)$.

Constraints:

$$P(m) \neq P(n), \forall m \neq n$$

In general, there are as many as $N!$ ways in which N nodes can be placed in N positions. That is the problem domain space grows exponentially with the number of nodes in the network. Hence, identifying the optimal structure among these $N!$ possibilities is computationally intensive. So, we investigate efficient heuristic algorithms for constructing near-optimal structures. The following section describes such heuristic algorithms.

4.2.1 Heuristic Solutions

Two heuristic algorithms proposed here uses (hop) distance, D_{NxN} and traffic T_{NxN} matrices as input and computes the places that the nodes should occupy in the logical structure to minimize the weighted average number of hops. These algorithms are based on greedy approach and assume that the logical structure derived from the base topology gives minimum mean hop distance. *Note that the number of nodes N represents only active nodes in the network.*

The first greedy algorithm given in Algorithm 4.1 maximizes one-hop traffic in the network. The algorithm sorts the N^2 elements of traffic matrix in descending order as *Ts* and the N^2 elements of distance matrix in ascending order as *Ds*. Then, the element on the top of list *Ts* is taken out of the list and the nodes corresponding to this element are placed on a link in the logical topology (an element of *Ds* with value 1). In successive steps, the next highest element of the sorted list *Ts* is assigned to one of the available links in the logical topology. During this process, if one of the nodes corresponding to the highest element of *Ts* is placed, the other node takes a neighboring position. Otherwise, the nodes are placed on a link corresponding to unoccupied places. It may be

observed that the elements of T and D are sorted in $O(N^2 \log N^2)$ time. Then, in order to assign all the nodes, $O(N^2)$ elements of Ts have to be considered. Assuming one of the nodes corresponding to the highest element of Ts is already placed, d neighbor locations directly connected to the already placed node are to be searched in $O(d)$ time for an available location. Thus, the worst-case time complexity of this algorithm is the maximum of $(O(N^2 \log N^2), O(N^2 d))$.

Algorithm 4.1 Find place i for node m in the logical topology, given the traffic matrix T and distance matrix D for an N node network.

```
Sort the Distance matrix D in ascending order as Ds and Traffic matrix T in
    descending order as Ts.
while (all nodes are not placed) do
    Let the highest element in Ts corresponds to traffic from node m to node n.
    if (both nodes m and n are not placed) then
        if (link is available from position i to j such that both positions i
            and j are unoccupied) then
            Place nodes m and n at two unoccupied positions i and j.
        endif.
    else if (node m is placed at position i) then
        if (link is available from position i to position j such that position j
            is unoccupied)
            Place node n al position j.
        endif.
    else if (node n is placed at position j) then
        if (link is available from position i to position j such that position i
            is unoccupied)
            Place node m at position i.
        endif.
    endif.
    Delete the highest element in Ts.
endwhile.
```

The second greedy algorithm is described in Algorithm 4.2. This algorithm also maximizes one-hop traffic. The algorithm sorts the N^2 elements of traffic matrix in descending order as Ts. Then, the element on the top of list Ts is taken out of the list and the node in the corresponding element, from which traffic is originated, is assigned location 0. In successive steps, the unassigned neighbors of an already assigned location are allotted to nodes corresponding to the highest traffic. It may be observed that the elements of T are sorted in $O(N^2 \log N^2)$ time. Then, in order to assign all the nodes,

$O(Nd)$ links of the logical topology have to be considered To place a node in an unassigned location neighboring to an assigned location *p*, *N*-\ nodes are searched in $O(N)$ time to find the node pair (p,q) such that the traffic from/? to *q* is maximum among all node pairs originated from *p* and the node *q* is not placed Thus, the worst-case time complexity of this algorithm is the maximum of $(O(N^2 \log N^2), O(N^2 d))$.

Algorithm 4.2 Find place i for node m in the logical topology, given the traffic matrix T and distance matrix D for an N node network.

```
Sort the Traffic matrix T in descending order as Ts.
Assign position 0 to a node corresponding to first element in Ts.
Push position 0 onto a stack S.
while (all nodes are not placed) do
        Pop position i from stack S.
        Let m be the node placed at position i.
        for (each neighbor j of position i in the logical topology)
                Push position j onto stack S.
                Find node pair (m,n) in Ts that corresponds to highest traffic
                    originated from node m such that node n is not placed.
                Assign node n to position j.
        endfor.
endwhile.
```

It may be observed that both the algorithms wouldn 7 make use of the structure of the logical topology. The worst-case time complexity for both the algorithms is same. It may be possible to develop better heuristics by considering properties unique to a particular topology and also traffic patterns. However, algorithms proposed here serve as yardsticks for evaluating reconfigurable topologies because of their generic nature. In the following section, these algorithms are used for evaluating dynamically reconfigurable logical topologies.

4.3 Performance of Perturbed Topologies

In this section, reconfigurable topologies are evaluated under uniform and non-linear traffic conditions. Theoretical evaluation of reconfigurable topologies based on the framework proposed is discussed in Chapter 3. It was observed that the diameter and average internode distance of reconfigurable topologies couldn't be computed as the reconfigurable topologies change their structure with each reconfiguration. It is better to compare the diameter and average internode distance of reconfigurable topologies

empirically with respective base topologies of corresponding size. The following section presents the empirical results of diameter and average internode distance obtained for various structures of dynamically reconfigurable topologies of different sizes. In Section 4.3.2, the topologies are compared based on the optimality criterion - minimize the weighted average internode distance.

4.3.1 Evaluation Under Uniform Traffic Conditions

The diameter and mean internode distance of reconfigurable topologies for various sizes and for different structures are compared with the corresponding base topology and/or the counterpart in traditional topologies. Table 4.1 shows the diameter and average internode distance of Perturbed Torus in comparison with the base 2-D Torus topology. Table 4.2 compares the diameter and average internode distance of RCG-I topology with the base corresponding Circulant graph. In Table 4.3 the diameter and average internode distance of RCG-1I topology are compared with the corresponding base Circulant graph. Finally, in Table 4.4, the diameter and average internode distance of RCG-III topology are compared with the corresponding base Circulant graph. It is observed from these comparisons that, in most cases, the diameter and mean internode distance of reconfigurable topology are less than the base topology with size equivalent to that of reconfigurable topology. In very few cases, where the distribution of nodes is too sparse, the diameter and mean internode distance of reconfigurable topology are more than that of the base topology with size equivalent to that of reconfigurable topology. For example, in Table 4.1, the diameter and average internode distance for a Perturbed Torus with 9 active nodes are more when compared to that of a Torus structure of size 9. But in no case, the diameter of the reconfigurable topology is more than that of the base topology from which the reconfigurable topology is derived. Continuing with the earlier example, the diameter and average internode distance for a Perturbed Torus with 9 active nodes are less when compared to that of base Torus structures of sizes 16 and 25. *Thus in a reconfigurable topology, the upper bound on the delay associated with a packet transmission between any two arbitrary nodes is limited by the size of the base topology.*

No. of active nodes	Diameter and Mean Internode Distance		
	Perturbed Torus derived from graph of		Torus with connection set $\{\pm 1, \pm\lceil\sqrt{\text{Nodes}}\rceil\}$
	25 nodes with failed nodes as	16 nodes with failed nodes as	
25	4 and 2.5		4 and 2.5
22	{7,10,23} 4 and 2.2684		4 and 2.3939
	{10-12} 4 and 2.394		
20	{6,10,18,19,22} 4 and 2.2053		4 and 2.3158
	{20-24} 4 and 2.3158		
16	{6,10,12,14-16,18,20,22} 4 and 2.0833	4 and 2.1333	4 and 2.1333
	{16-24} 4 and 2.075		
15	{5,8,10,12,14-16,18,20,22} 3 and 2.0191	{5} 4 and 1.981	4 and 1.981
	{5-11,15,20,21} 4 and 2.0095	{14} 4 and 1.981	
14	{6,7,11,12,15,17-22} 4 and 2.1429	{5,8} 3 and 1.8791	4 and 1.989
	{5-9,13,16,18,22-24} 3 and 1.9341	{14,15} 4 and 1.989	
9	{6-10,12-16,18-23} 4 and 2.2222	{6,7,10-14} 4 and 1.9444	2 and 1.5
	{9-24} 3 and 1.8056	{5-8,10,12,14} 3 and 1.7778	
8	{6-9,11-23} 3 and 1.9643	{5,6,8,9,10,12-14} 4 and 1.9286	2 and 1.5
	{8-24} 3 and 1.75	{9-15} 3 and 1.7143	

Table 4.1 Diameter and mean internode distance of Perturbed Torus and 2-D Torus

No. of active nodes	Diameter and Mean Internode Distance		
	Reconfigurable Circulant Graph derived from graph of		Circulant Graph with connection set $\{\pm 1, \pm \lceil \sqrt{Nodes} \rceil\}$
	24 nodes with failed nodes as	15 nodes with failed nodes as	
24	4 and 2.4348		4 and 2.4348
23	{6} 4 and 2.3083		3 and 2.2727
	{19} 4 and 2.3083		
20	{6,10,18,19} 4 and 2.1105		4 and 2.3157
	{20-23} 4 and 2.1895		
15	{10,12,14-16,18,20,22,23} 3 and 1.8857	3 and 1.8571	3 and 1.8571
	{16-24} 3 and 2.0		
14	{5,8,10,12,14-16,18,20,22} 3 and 1.7912	{5} 3 and 1.7912	3 and 1.7692
	{5-11,15,20,21} 3 and 1.8642	{14} 3 and 1.7912	
13	{6,7,11,12,15,17-23} 3 and 1.8077	{5,8} 3 and 1.6923	3 and 1.8333
	{3,5-9,13,16,18,22,23} 3 and 1.7949	{2,14} 3 and 1.6923	
8	{6-10,12-16,18-23} 2 and 1.4642	{3,7,10-14} 2 and 1.4286	2 and 1.4286
	{8-23} 2 and 1.4643	{4-7,10,12,14} 2 and 1.4643	
7	{6-9,11,12-23} 3 and 1.6191	{4,6,8-10,12-14} 3 and 1.6191	2 and 1.3333
	{3,4,8,9,11-23} 2 and 1.33	{0,3,4,7,8,11,12,14} 2 and 1.3333	

Table 4.2 Diameter and mean internode distance of of RCG-land Circulant Graph with connection set {- √Nodes ,-J, + I, + √ Nodes }.

No. of active nodes	Diameter and Mean Internode Distance			
	Reconfigurable Circulant Graph derived from graph of		Circulant Graph with connection set $\{-k^{n-1},-k^{n-2},..,-1,+1,..,+k^{n-2},+k^{n-1}\}$, $n=\log_k^N$	
	63 nodes with failed nodes as	15 nodes with failed nodes as	K=4	k=3
63	4 and 2.7097		4 and 2.7097	4 and 2.5161
40	{6-15,20-30,44,62} 4 and 2.3756		4 and 2.359	4 and 2.1539
	{19-41} 4 and 2.3551			
30	{6,10,18,19,32-60} 4 and 2.1816		3 and 2.069	4 and 2.2414
	{20-52} 3 and 2.1058			
15	{10,12,14-16,18,20-62} 3 and 1.8095	3 and 1.8571	3 and 1.8571	2 and 1.5714
	{15-62} 3 and 1.8571			
14	{5,8,10,12,14,15,16,18, 20,22,24-62} 3 and 1.6593	{5} 3 and 1.7912	3 and 1.7692	3 and 1.6154
	{5-11,15,20,21,23-61} 3 and 1.7143	{14} 3 and 1.7912		
13	{6,7,11,12,15,17-61} 3 and 1.7949	{5,8} 3 and 1.6923	3 and 1.8333	2 and 1.5
	{3,5-9,13-16,18,22,23,25-61} 3 and 1.6282	{2,14} 3 and 1.6923		
8	{1-4,6-9,11-14,16-19,21-24,26-29,31-34,36-39,41-62} 2 and 1.3929	{3,7,10-14} 2 and 1.4286	2 and 1.5714	2 and 1.4286
	{8-62} 2 and 1.4643	{4-7,10,12,14} 2 and 1.4643		
7	{6-9,11-62} 2 and 1.3333	{4,6,8-10,12-14} 3 and 1.6191	2 and 1.3333	2 and 1.3333
	{3,4,8,9,11-62} 2 and 1.4762	{0,3,4,7,8,11,12,14} 2 and 1.3333		

Table 4.3 Diameter and mean internode distance of RCG-II and Circulant Graph with connection set $\{-k^{n-1},-k^{n-2},..,-1,+1,..,+k^{n-2},+k^{n-1}\}$, *k=3 and 4.*

No. of active nodes	Diameter and Mean lnternode Distance		
	Reconfigurable Circulant Graph derived from graph of		Circulant Graph with connection set $\{1,2,\ldots,2^{n-2},2^{n-1}\}$.
	31 nodes with failed nodes as	15 nodes with failed nodes as	
31	4 and 2.5		4 and 2.5
28	{6,20,24} 4 and 2.4762		4 and 2.3704
	{24-26} 4 and 2.4749		
20	{10-13,20-23,26,28,30} 4 and 2.3184		4 and 2.1053
	{0,2,4,6,8,10,11,13,15,17,19} 4 and 2.2526		
15	{1,3,5,7,9,11,13,15,17,19-25}4 and 2.2286	3 and 2.0	3 and 2.0
	{5-20} 3 and 2.0		
14	{5,8,10,12,14-16,18,20-28} 4 and 2.1813	{5} 3 and 2.0	3 and 1. 9231
	{5-11,15,20,21,24-30} 4 and 2.1374	{14} 3 and 2.0	
13	{6,7,11,12,15,17-29}4 and 2.1026	{5,8} 3 and 2.0064	3 and 1.8333
	{3,5-9,13,16,18,22-30} 4 and 2.0769	{2,14} 3 and 2.0064	
8	{6-10,12-16,18-30} 3 and 1.75	{3,7,10-14} 3 and 1.8214	3 and 1.7143
	{8-30} 3 and 1.7321	{4-7,10,12,14} 3 and 1.8393	
7	{6-9,11-30} 2 and 1.5	{4,6,8-10,12-14} 3 and 1.7857	2 and 1.5
	{3,4,8,9,11-30}3 and 1.6905	{0,3 4,7,8,11,12,14} 2 and 1.5	

Table 4.4 Diameter and mean inteniode distance of RCG-IIIand Circulant Graph with connection set $\{1,2,\ldots,2^{n-2},2^{n-1}\}$.

4.3.2 Performance Under Non-Linear Traffic Conditions

The performance of a logical topology under non-linear traffic conditions is measured using the optimal criterion - weighted average internode distance computed as the sum of product of traffic and distance between all pairs of active nodes. Assuming the nodes are placed using one of the node placement algorithms, described in Section 4.2.1, in a dynamically reconfigurable topology, in this section, we show that the performance of dynamically reconfigurable logical topology after a reconfiguration is almost same as the performance of newly computed optimal structure with nodes being placed using the node placement algorithms.

Let us assume a dynamically reconfigurable topology of size, N, equivalent to that of base topology. That is, all the nodes in the base topology are active. Now generate the traffic matrix T such that each element $t(m,n)$, which indicates the fraction of traffic generated from node m to node n, is a uniformly distributed random number between 0 and 1. (The random number generator uses a 48-bit seed, which is modified using a linear congruential formula.) Place the N nodes in the base topology using one of the greedy algorithms discussed in Section 4.2.1. Assume some nodes are failed from this structure and compute the objective function – weighted average internode distance, using all active nodes, in the resultant structure after reconfiguration, as Cr. Compute optimal structure by removing the farthest places, equivalent to the number of failed nodes, in the base topology. (Optimal structures are relevant to the reconfigurable topologies as the structure changes with each reconfiguration.) Perform the optimal node placement with the active nodes in this newly computed optimal structure, using the same greedy algorithm. Compute the objective function - weighted average internode distance, in this resultant topology, as *Co*. Repeating the procedure for different events such as node failures and node additions in the current reconfigurable topology, the results of *Cr* and *Co* are tabulated for dynamically reconfigurable topologies namely Perturbed Torus, RCG-I, RCG-II and RCG-III in Table 4.5, Table 4.6, Table 4.7, Table 4.8, Table 4.9, Table 4.10, Table 4.11 and Table 4.12.

Each table lists the results of weighted average internode distance in a reconfigurable topology under different events. From the results in each table, it can be said that the

values of Cr are almost same as *Co*, given the fact the values of Co computed for the two optimal node placement algorithms differ to some extent. *Hence, the performance of dynamically reconfigurable logical topology after a reconfiguration is on par with the performance of an optimal structure with the active nodes being optimally placed in it.*

Event	Value of weighted average internode distance is computed with uniformly distributed random traffic and by using			
	Algorithm 4.1		Algorithm 4.2	
	Re-placing, *Co*	Reconfiguring, *Cr*	Re-placing, *Co*	Reconfiguring, *Cr*
Initial configuration N=16	28.4658		33.3236	
Node 4 is deleted	24.3127	23.4546	27.7331	28.4605
Node 6 is deleted	25.1987	21.4653	26.3804	24.4336
Node 4 is added	25.3813	25.3642	28.2737	27.6503
Nodes 5,7-10 are deleted	16.0905	14.9919	17.8546	19.2565
Node 8 is added	17.5971	15.832	17.9057	19.6328

Table 4.5 Effect of reconfiguration over optimal node placement in Perturbed Torus derived from Base Torus topology of size is 16

Event	Value of weighted average internode distance is computed with uniformly distributed random traffic and by using			
	Algorithm 4.1		Algorithm 4.2	
	Re-placing, *Co*	Reconfiguring, *Cr*	Re-placing, *Co*	Reconfiguring, *Cr*
Initial configuration N=36	98.9001		109.8605	
Node 29 and 30 are deleted	86.5085	88.922	95.3007	95.9788
Nodes 20-24 are deleted	69.8376	66.3536	75.6439	76.1082
Nodes 22,23 and 30 are added	81.1332	77.6815	91.5607	85.9814
Node 4 is deleted	73.591	70.986	82.6919	77.0783
Node 6 is deleted	69 2377	66.165	79.7713	72.427
Node 4 is added	76 2844	72.6524	84.9657	80.8894
Nodes 5,7-10 are deleted	64.8752	54.8654	66.5775	61.2266
Node 8 is added	62 2224	59 3319	69.9359	64.294

Table 4.6 Effect of reconfiguration over optimal node placement in Perturbed Torus derived from Base Torus topology of size is 36

Event	Value of weighted average internode distance is computed with uniformly distributed random traffic and by using			
	Algorithm 4.1		Algorithm 4.2	
	Re-placing, *Co*	Reconfiguring, *Cr*	Re-placing, *Co*	Reconfiguring, (V
Initial configuration N=15	24.6058		26.5375	
Node 4 is deleted	20.9154	21.1336	23.8453	26.7073
Node 6 is deleted	18.8658	19.2347	19.9683	22.1514
Node 4 is added	21.6925	21.7034	25.5399	23.7094
Nodes 5,7-10 are deleted	11.1411	11.1547	11.893	10.8957
Node 8 is added	12.3627	12.6094	14.4196	13.3461

Table 4.7 Effect of reconfiguration over optimal node placement in RCG-I derived Base CG-I Topology of size is 15

Event	Value of weighted average internode distance is computed with uniformly distributed random traffic and by using			
	Algorithm 4.1		Algorithm 4.2	
	Re-placing, *Co*	Reconfiguring, *Cr*	Re-placing, *Co*	Reconfiguring, *Cr*
Initial configuration N=35	90.0137		104.3855	
Node 29 and 30 are deleted	83.7405	81.9446	91.9362	96.0792
Nodes 20-24 are deleted	63.938	65.372	72.465	72.4695
Nodes 22,23 and 30 are added	75.3341	73.5967	86.3008	83.6302
Node 4 is deleted	67.4715	66.3219	80.3846	76.2243
Node 6 is deleted	65.6802	62.281	73.6651	72.3947
Node 4 is added	74.4866	69.1322	83.627	79.8381
Nodes 5,7-10 are deleted	53.2446	51.0204	60.2332	60.2143
Node 8 is added	56 5462	54.2812	62.5398	63.5271

Table 4.8 Effect of reconfiguration over optimal node placement in RCG-I derived Base CG-I Topology of size is 35

Event	Value of weighted average internode distance is computed with uniformly distributed random traffic and by using			
	Algorithm 4.1		Algorithm 4.2	
	Re-placing, *Co*	Reconfiguring, *Cr*	Re-placing, *Co*	Reconfiguring, *Cr*
Initial configuration N=26	46.0506		51 387	
Node 4 is deleted	42.2395	42.1866	50 8498	47 2463
Node 6 is deleted	39.007	39.3143	44 7229	43 4728
Node 4 is added	42.0266	42.5406	51 2122	46 9985
Nodes 5,7-10 are deleted	31.4707	29.2172	34 3846	30 6797
Node 8 is added	31.6423	31.5686	37 1351	32 9335

Table 4.9 Effect of reconfiguration over optimal node placement in RCG-II derived Base CG-II Topology of size is 26

Event	Value of weighted average internode distance is computed with uniformly distributed random traffic and by using			
	Algorithm 4.1		Algorithm 4.2	
	Re-placing, *Co*	Reconfiguring, *Cr*	Re-placing, *Co*	Reconfiguring, *Cr*
Initial configuration N=63	147.0277		167.2691	
Node 29 and 30 are deleted	138.6884	140.0186	161.1473	157.3261
Nodes 20-24 are deleted	130.4936	122.891	146.8377	138.0057
Nodes 22,23 and 30 are added	138.8908	134.001	152.8549	150.8976
Node 4 is deleted	126.5256	127.0602	144.3783	141.3854
Node 6 is deleted	125.204	122.7523	147.2828	138.6784
Node 4 is added	128 9582	128.7305	146.6653	147.7384
Nodes 5,7-10 are deleted	112.6306	108.4309	133.5824	127.6864
Node 8 is added	114.8848	112.1649	134.6788	131.4723

Table 4.10 Effect of reconfiguration over optimal node placement in RCG-II derived Base CG-II Topology of size is 63

Event	Value of weighted average internode distance is computed with uniformly distributed random traffic and hy using			
	Algorithm 4.1		Algorithm 4.2	
	Re-placing, Co	Reconfiguring, *Cr*	Re-placing, *Co*	Reconfiguring, *Cr*
Initial configuration N=15	26.7536		30.869	
Node 4 is deleted	22.2559	23.085	26.5775	27.4465
Node 6 is deleted	20324	21.6256	23.7402	26.1049
Node 4 is added	23.0019	24.8989	27.4409	29.1322
Nodes 5,7-10 are deleted	15.258	12.1397	15.258	15.5397
Node 8 is added	15.1535	15.2841	15.1535	18.8935

Table 4.11 Effect of reconfiguration over optimal node placement in RCG-III derived Base CG-1II Topology of size is 15

Event	Value of weighted average internode distance is computed with uniformly distributed random traffic and by using			
	Algorithm 4 1		Algorithm 4.2	
	Re-placing, *Co*	Reconfiguring, *Cr*	Re-placing, *Co*	Reconfiguring, *Cr*
Initial configuration N=31	67.0525		74.7479	
Node 29 and 30 are deleted	60.0943	63.0874	72.8389	68.3697
Nodes 20-24 are deleted	52.8429	49.5823	50.8835	56.6345
Nodes 22,23 and 30 are added	55.7489	58.4106	65.6059	63.4094
Node 4 is deleted	56.0142	55.3843	65.3866	61.0455
Node 6 is deleted	48.993	52.317	59.4481	56.9291
Node 4 is added	55 5687	55.283	63.0624	59.2465
Nodes 5,7-10 are deleted	41.0782	39.8379	45.8872	46.368
Node 8 is added	40 3031	42.112	46.8701	48.5664

Table 4.12 Effect

Base CG-II1 Topology of size is 31

4.4 Discussion

The diameter and average internode distance of the dynamically reconfigurable topologies - Perturbed Torus, RCG-I, RCG-II and RCG-III, are found to be on par with that of base topological structures with size equals to that of reconfigurable topology. Also, the effect of reconfiguration based on Local Perturbations under non-linear traffic conditions is found to be unnoticeable as the value of objective function - flow weighted average internode distance, computed after reconfiguration is almost same as the value found using the optimal node placement algorithms described in Section 4.2.1. *Thus the reconfigurable topologies are better than traditional topologies such as Torus and Binary Hypercube as reconfigurable topologies give added advantage of dynamic reconfiguration with minimum disturbance to the network, which not only provides the highest reliability, but also accommodate arbitrary number of nodes.*

In the next chapter, we study the fiber optic LAN/MAN architectures using the dynamically reconfigurable topologies and also evaluate the networks using simulation.

Chapter

5. Fiber Optic LAN/MAN Architectures based on Dynamically Reconfigurable Logical Topologies

This chapter presents various implementation techniques for fiber optic LAN/MAN s using the dynamically reconfigurable topologies designed in proposed in this thesis. Section 5.1 provides an introduction to various issues addressed by different fiber optic architectures and also provides an understanding of these architectures. Section 5.2 describes the architectures based on broadcast physical topologies that use dynamically reconfigurable topologies as virtual topology. Section 5.3 discusses how network architectures could be designed by sharing WDM channels to realize dynamically reconfigurable virtual topologies. Section 5.4 provides the simulation results of the networks based on reconfigurable topologies. Finally, section 0 summarizes the results.

5.1 Introduction

Fiber optic network architectures can be classified into two broad categories: broadcast and select, and wavelength routing architectures. *In broadcast and select architectures, different nodes broadcast messages on different WDM channels, typically, over passive star, linear bus, or tree topologies. Access nodes in these networks route the messages.* In contrast, wavelength routing networks employ active switches to route messages based on the wavelengths and thus messages carried in optical domain "as far as possible". *Today, broadcast topologies are used for LAN/MAN solutions because they are simple to construct and cost effective.*

Figure 5.1 shows various alternatives for broadcast and select networks. These networks can be built using the optical couplers. For example, the star coupler shown in Figure 5. l(a) can be made out of the 2x2 couplers as shown in Figure 5.2.

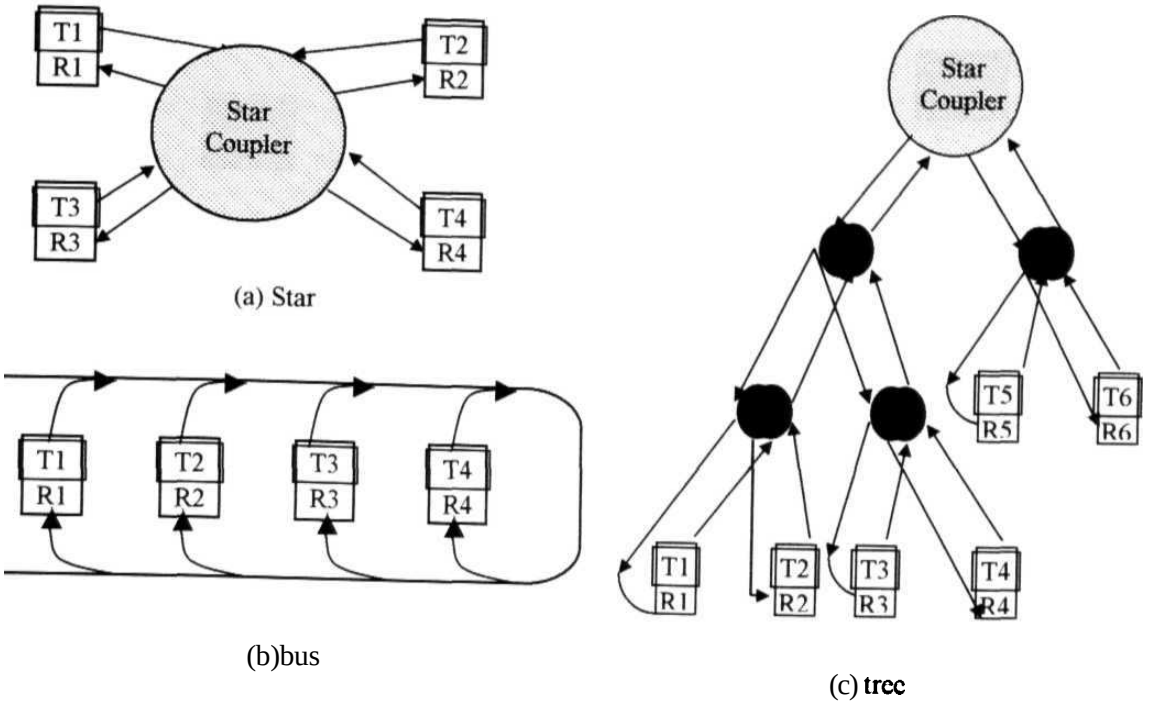


Figure 5.1 Broadcast and Select physical topologies

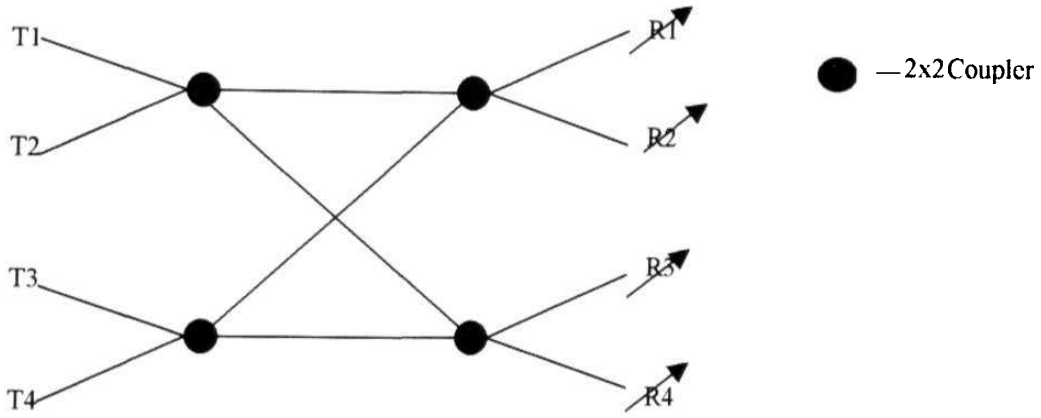


Figure 5.2 Internal structure of a Star Coupler

Each node transmits at different wavelengths, which are combined using 2x2 couplers. The node employs tunable optical filters to select the desired wavelengths for reception. *A node, in broadcast and select network, broadcasts messages on the wavelengths assigned to them.* Theoretically, any node can receive these messages by tuning their

receivers to the transmitted wavelengths. However, only selected nodes dictated by the logical topology receive these messages.

The number of nodes in the broadcast and select networks is limited. This is because the wavelengths cannot be reused in the network. Also, the transmitted power from a node must be split among all the receivers in the network. The design of these networks should also keep in mind the capabilities and limitations of the available optical components and technology such as

- the maximum number of channels attainable with current device technology, subject to tuning time requirements and cost constraints,
- the number of transceivers, and
- tuning range and tuning times of the transmitters and/ or receivers

In order to accommodate more number of nodes and also to overcome the limitations of optical components such as tuning range, WDM channels in a broadcast topology are shared by different links in logical topology. These architectures referred as shared channel multihop architectures. Generally, channel sharing advocates the use of Time Division Multiplexing (TDM) as multiple access mechanism for sharing a common channel. However, other channel arbitration strategies are studied by [Dowd 1991; Dowd 1992] in connection with a shared-channel Hypercube architecture.

The following sections discuss both broadcast and select architectures, and shared channel multihop architectures using dynamically reconfigurable topologies designed in this thesis.

5.2 Broadcast and Select Network Architectures

The broadcast and select architectures based on dynamically reconfigurable logical topologies designed earlier - Perturbed Torus, RCG-I, RCG-II and RCG-III can be constructed using any of the physical topologies in Figure 5.1. The bandwidth of these physical topologies is divided into a set of logical (WDM) channels. In order to realize the logical topology the transmitters and/or receivers at each node are tuned to different wavelengths of the WDM channels. Let c be the maximum number of channels that can

be realized based on technological and topological limits. Let us assume that the size of the base topology is *N* and the degree of a node in the base logical topology is *d*, then *c*, *N* and *d* are related using the formula

$$N = c/d$$

Dynamically reconfigurable logical topologies that use Local Perturbations change their structure with each reconfiguration and the size of the network is limited to the size of the base structure. In order to facilitate reconfiguration due to a change in the network such as addition of a node and deletion of a node, nodes must employ tunable components. *Depending on whether the tune-ability is incorporated into transmitters or receivers, the broadcast and select architectures can be implemented in two different ways - using fixed transmitters and tunable receivers (FT-TR), and using tunable transmitters and fixed receivers (TT-FR).* These implementations vary in terms of assigning transmitters to receivers. For example, consider RCG-I topology with a base Circulant graph topology of size 8 is implemented using fixed transmitters and tunable receivers (FT-TR) over linear bus topology as shown in Figure 5.3. Assuming node 4 is deleted from the network, the wavelengths of the receivers of neighboring nodes of the deleted node 4 are reconfigured resulting in the new wavelength assignment shown in Figure 5.4.

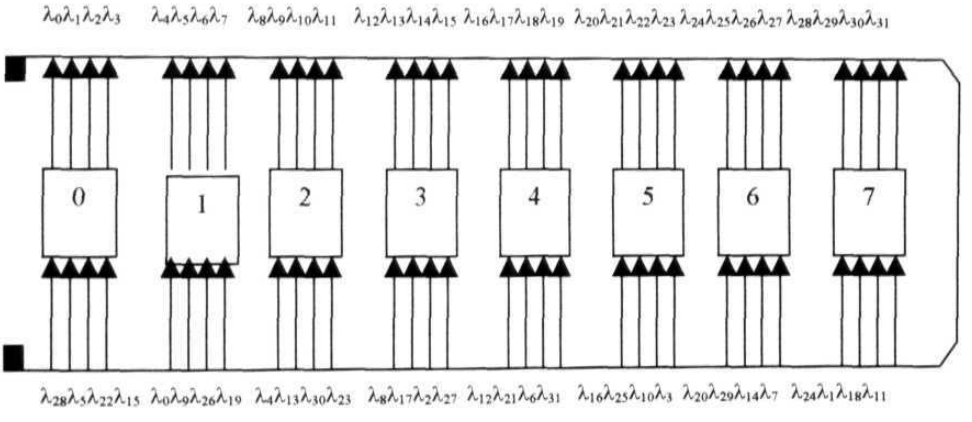


Figure 5.3 Wavelength assignment in FT-TR implementation of RCG-I topology with base Circulant graph of 8 nodes over linear bus physical topology

In general, for an FT-TR implementation, wavelengths of transmitters of node *X* are calculated as

$$T(X,s) = X*d + p(s) \qquad (5.1)$$

where *d* is node degree and *p(s)* takes the values 0,1,2,3,...,*d*-1, for various values of jump *s*. For example, the jumps +1, -1, +*n*, -*n* in Perturbed Torus and RCG-1 topologies take values 0,1,2 and 3, respectively.

Similarly, the receivers of the node *X* are tuned to wavelengths

$$R(X,s) = N(X,s)*d + p(s) \qquad (5.2)$$

where *N(X,s)* is the address of neighbor of node *X* in the direction of jump *s*.

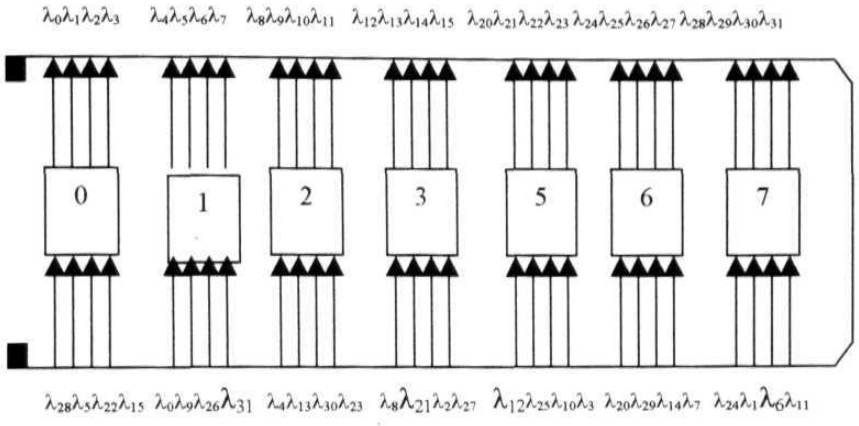


Figure 5.4 Wavelength assignment in FT-TR implementation when node 4 is deleted from the topology shown in Figure 5.3

In TT-FR implementation, the transmitter and receiver wavelengths are calculated using the following equations.

$$T(X,s) = N(X,s)*d + p(s) \qquad (5.3)$$

$$R(X,s) = X*d + p(s) \qquad (5.4)$$

where *d* is node degree and *p(s)* takes the values 0,1,2,3,...,*d*-1, for various values of jump *s*. *N(X,s)* is the address of neighbor of node *X* in the direction of jump *s*.

For example, in a TT-FR implementation of a RCG-1 topology with a base topology of size 8, nodes are connected as shown in Figure 5.5. When a node 4 is deleted from this network, the transmitters of neighbors of node 4 are reconfigured resulting in the network shown in Figure 5.6.

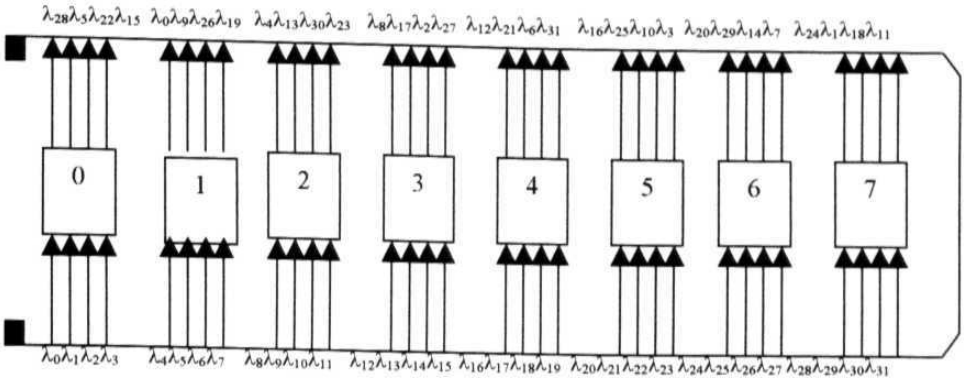


Figure 5.5 Wavelength assignment in TT-FR implementation of RCG-1 topology with base Circulant graph of 8 nodes over linear bus physical topology.

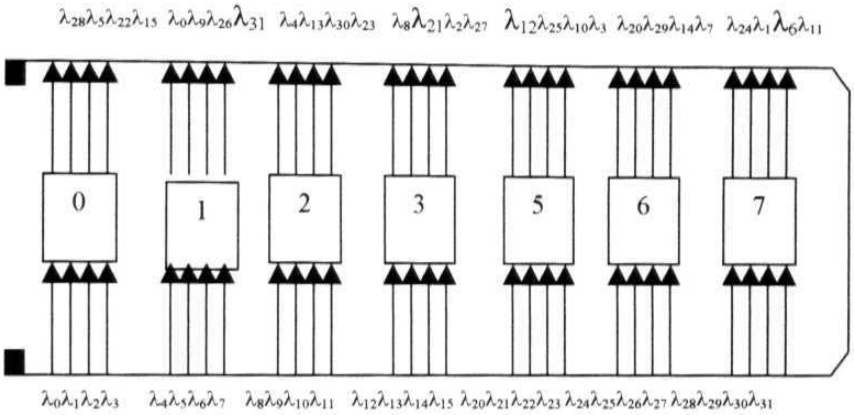


Figure 5.6 Wavelength assignment in TT-FR implementation when node 4 is deleted from the topology shown in Figure 5.5

Both TT-FR and FT-TR implementations must employ a *control channel* to which all nodes are connected. *The control channel is mainly used to facilitate reconfiguration due to a change in the network, i.e., addition or failure of a node.* Before the reconfiguration is initiated, neighboring nodes in the network must be aware of the change. For example, if a node wishes to enter the network, it should inform the neighbors so that they can tune transmitters and/or receivers to the wavelengths of the node being added. Also, the node being added should know which nodes in the network constitute as its neighbors. Similarly, when a node is found to be not responding by its neighbors; they confirm the failure of a node by exchanging the messages among themselves. Once a node is

identified as failed, the neighboring nodes tune their wavelengths according to the links established by reconfiguration.

Generally, the arbitration mechanism for the control channel is random. *The control channel may also be used for transferring messages other than the one used for reconfiguration.*

It may be observed that the number of WDM channels used by a reconfigurable topology in a broadcast topology implementation is limited by the limitations of WDM technology, tuning ranges of transmitters and/or receivers and characteristics of reconfigurable topology such as diameter and average internode distance. The following section discusses shared channel architectures that overcome the channel limitation of broadcast and select architectures.

5.3 Shared Channel Network Architectures

Channel sharing was introduced by [Acampora, 1987] with a goal of improving the utilization of a multihop link (wavelength). *This work uses "channel sharing" concept for increasing the size of the base topology of a reconfigurable logical topology.* In other words, a wavelength is used by more than one link in the logical topology. Traditionally, random access protocols are used for sharing channel under light loads and Time Division Multiplexed Accessing (TDMA) is used for sharing channel higher loads [Dowd, 1992]. *Random access protocols are suitable for reconfigurable topology networks to effectively utilize the channel, when the size of a network is much smaller than the size of the base topology.* This is because TDMA requires rearrangement of time slots with each reconfiguration and it involves the coordination of all nodes to reassign time slots, which is against the spirit of Local Perturbations paradigm.

Channel sharing is required when $c<N*d$, where c is the number of channels that can be realized due to technological limitations, N is the size of the base topology and d is node degree. Works in [Acampora, 1987; Dowd 1991; Dowd, 1992; Ganz and Li, 1992; Kovacevic et. al., 1995] address channel sharing for two cases; First one is when $c=N$ and the second case is for $d<c<N$. Based on the techniques used in [Acampora, 1987; Dowd

1991; Dowd, 1992; Ganz and Li, 1992; Kovacevic et. al., 1995] for specific regular topologies, we proposed general methods for channel sharing.

Let us assume $c=N$. In this case, $d*N$ logical links of reconfigurable topology are assigned to $c=N$ channels with each node assigned to a different channel to transmit packets to its neighbors. Since every node is assigned to one channel for transmitting packets, one transmitter is enough for transmitting and the receivers of neighbor nodes are tuned to the transmission wavelength. However, the number of receivers can be one or more. Using one receiver per node is not advisable because it requires switching of channels after each packet reception based on the logical topology. By assigning d tunable receivers to transmission channels of neighbor nodes, each node constantly listens for packets arriving on the channels of neighbor nodes and tunes the channels whenever reconfiguration occurs. *It is worth noticing that since each channel has been assigned to only one node, there is no contention among nodes, and hence no coordination required for channel sharing. Also, with each reconfiguration due to node failure one channel becomes unused.*

When $d<c<N$, the problem of channel sharing becomes assignment of $d*N$ links of base structure of reconfigurable topology to c channels. To provide simple solution, we assume N/c is an integer. Assuming each node transmits on one channel, N/c nodes compete for transmitting packets on a single channel. The receivers of the neighbor nodes are tuned to the transmission wavelengths to receive packets. Since each node transmits on a single channel, one transmitter is enough. The use of one receiver per node is not advisable because it requires switching of channels after each packet reception based on the logical topology. By using d tunable receivers to transmission channels of neighbor nodes, each node constantly listens for packets arriving on the channels of neighbor nodes and tunes the channels whenever reconfiguration occurs. // *may be worth noticing that the average hop distance can be reduced, by employing the knowledge of neighbors of all nodes assigned to a channel into routing algorithm. Also when N/c is large, node placement algorithms described in previous chapter will not provide any benefit.*

5.4 Simulation Results

This section presents the performance of the networks based on dynamically reconfigurable topologies and compares the same with the networks based on corresponding base topologies. Performance measures such as throughput, average queuing delay and utilization of bandwidth are studied through simulation by varying average packet arrival rate for different sizes of the network.

The simulators are based on stochastic discrete event models, written in the Java programming language. Network simulators assume that traffic is homogeneous and packets are generated based on Poisson distribution. Propagation delays are assumed to be same for all pairs of nodes. Packets are transmitted at fixed slots in time. Packet arrival rate is measured as the number of packets generated at each node in one packet slot time. We assume, each node has two queues -- a transmission queue and an insertion queue. Transmission queue is used for the packets generated at the node. Insertion queue is used for the packets being routed through the node. Both the queues are finite in size and have the same size. Simulation results are also studied to verify the effect of buffer size on the network performance.

For a multihop network, efficient routing algorithm is important in determining the performance of the network. For networks based on reconfigurable topologies, defining shortest path routing is found to be difficult (refer Chapter 3). Hence, we used routing algorithms proposed in Chapter 3. These routing algorithms reduce the absolute distance of the destination node with each hop.

In the following sections, we analyze the throughput, average queuing delay, and channel utilization against packet arrival rate for Perturbed Torus, RCG-I, RCG-II and RCG-III based on broadcast and select networks.

5.4.1 Throughput Analysis

Throughput of reconfigurable topologies is studied against packet arrival rate. Normalized throughput is measured as the average number of packets reached a destination node in one packet time slot. Network throughput is measured as the Normalized throughput times the size of the network. In this study, we refer normalized

throughput as throughput. Figure 5.7, Figure 5.8 and Figure 5.9 analyze the throughput of the networks based on reconfigurable topologies for different packet arrival rates by varying the size of the network, configurations of logical topology and the number of packet buffers in transmission and insertion queues. From these figures, it can be observed that as the arrival rate increases, the throughput will first increase as more number of packets becomes available for transmission, and then eventually stabilizes because each node has limited number of transmitters.

Figure 5.7 depicts normalized throughput for various sizes of reconfigurable topologies under different packet arrival rates. It is observed that the normalized throughput decreases for larger networks. This is because as the size of network increases, the average number of hops a packet makes also increases. However, the network throughput increases with the increase of the size of the network.

Figure 5.8 plots the throughput for two different structures of reconfigurable topologies of same size. It is observed that the configuration (connectivity of nodes) of a reconfigurable topology sometimes affects the throughput. This is evident from the topological characteristics studied, earlier, in Chapter 3 and Chapter 4 that diameter and average internode distance vary from one configuration to another. However, the throughput of a network based on reconfigurable topology is usually limited by the throughput of network with the base topology.

Figure 5.9 illustrates the effect of buffer size on throughput. From the figure it is clear that buffers play limited role on throughput. This is because, the packets in the insertion buffer are given priority over the packets generated at the node and the minimum size of insertion buffer is taken as the number of transmitters (degree of node). However, if one uses a single buffer for the packets generated at the node as well as the packets being routed, the buffer size plays vital role in deciding the throughput.

Figure 5.10 provides comparison of throughput of the proposed reconfigurable topology networks (Perturbed Torus, RCG-I, RCG-I1 and RCG-III) and their counterparts in traditional logical topologies (Torus and Hypercube). The networks are compared with almost same size. The buffer sizes of transmission queue and insertion queue is taken as 12. It is observed that the throughput of the network based on RCG-II topology is better

than all other topologies. The throughput of RCG-III and Perturbed Torus is comparable to Hypercube and 2-D Torus topologies.

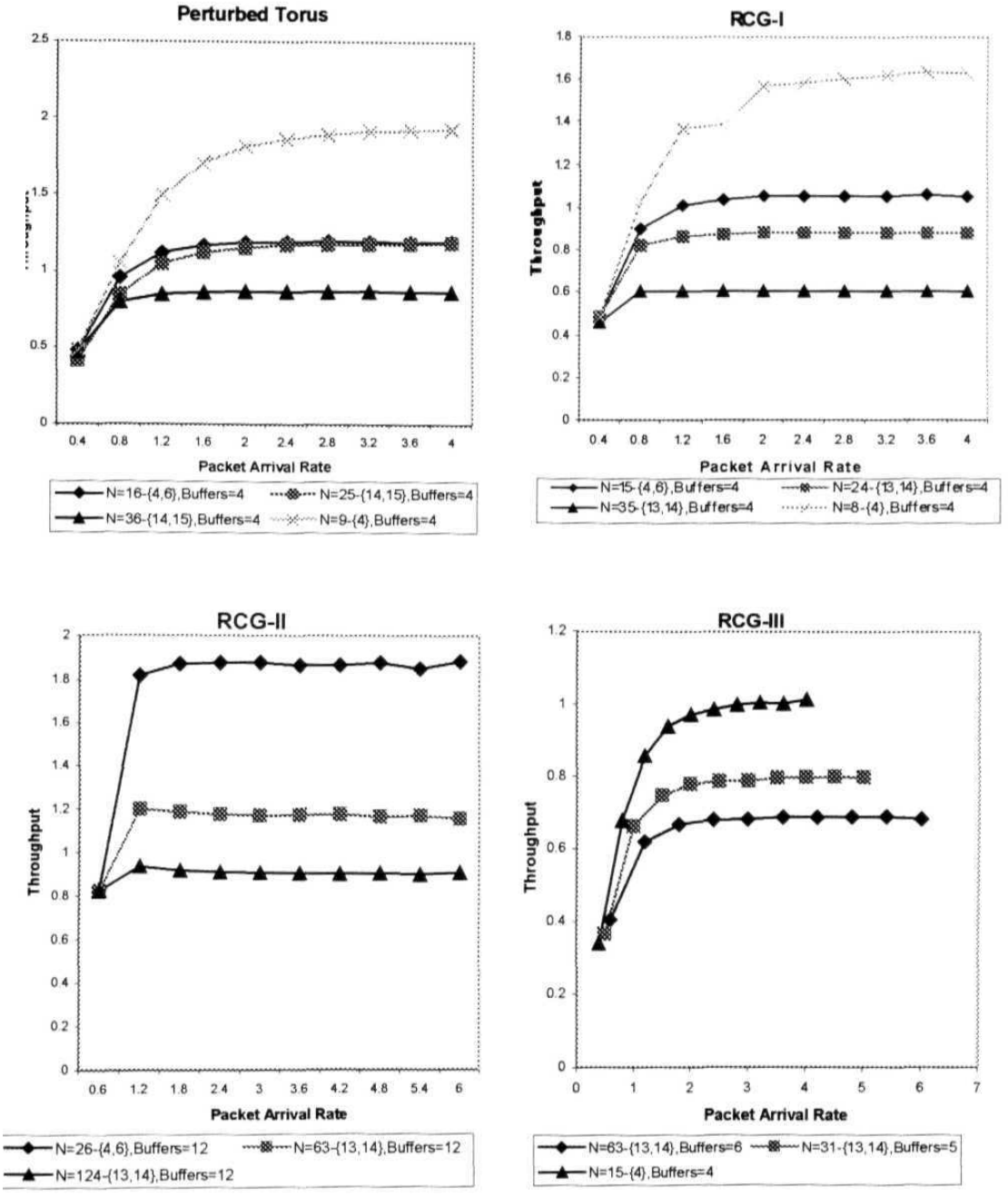


Figure 5.7 Throughput Vs. Packet arrival rate for different sizes of reconfigurable topologies

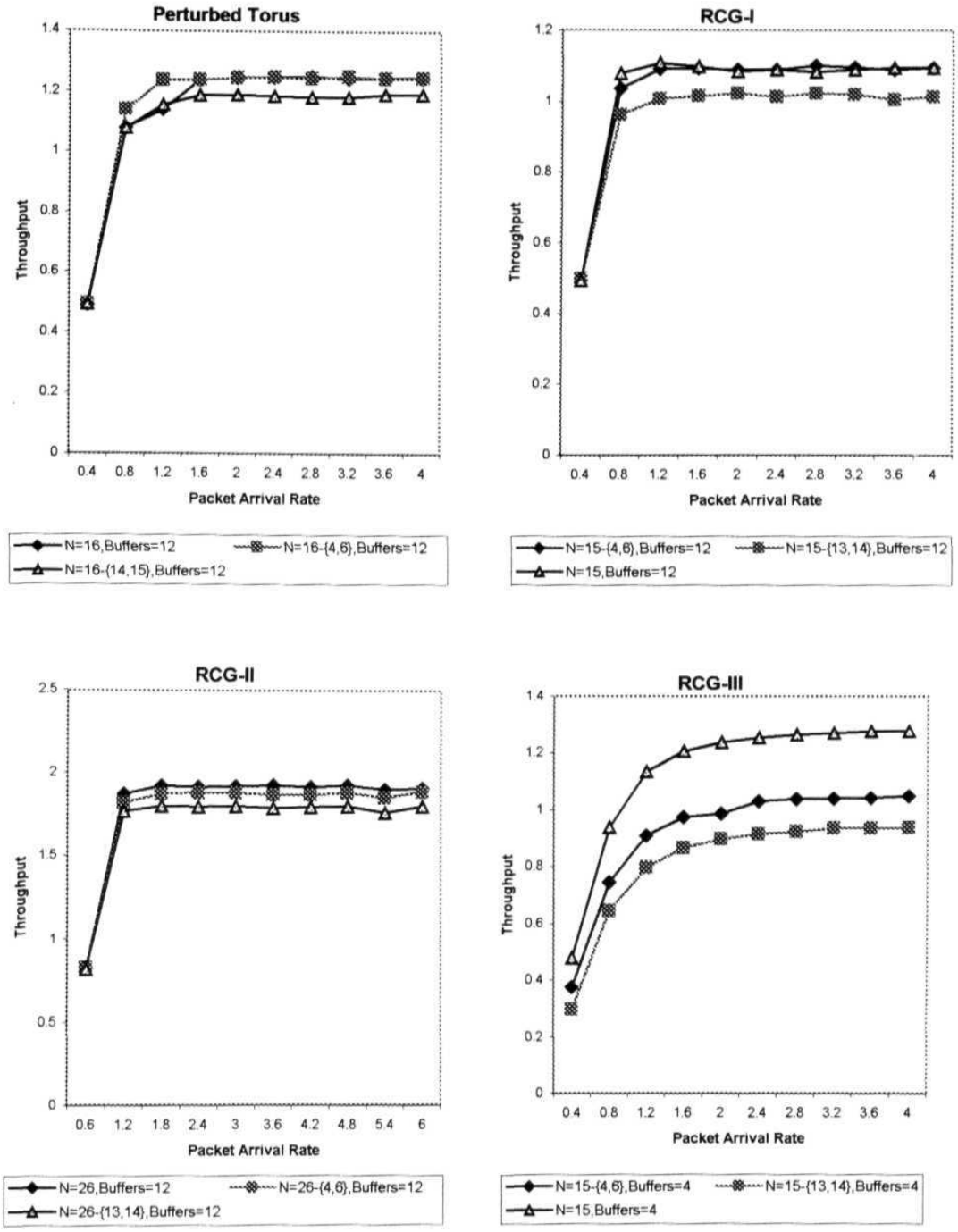


Figure 5.8 Throughput Vs. Packet arrival rate for reconfigurable topologies of same size, but different structures

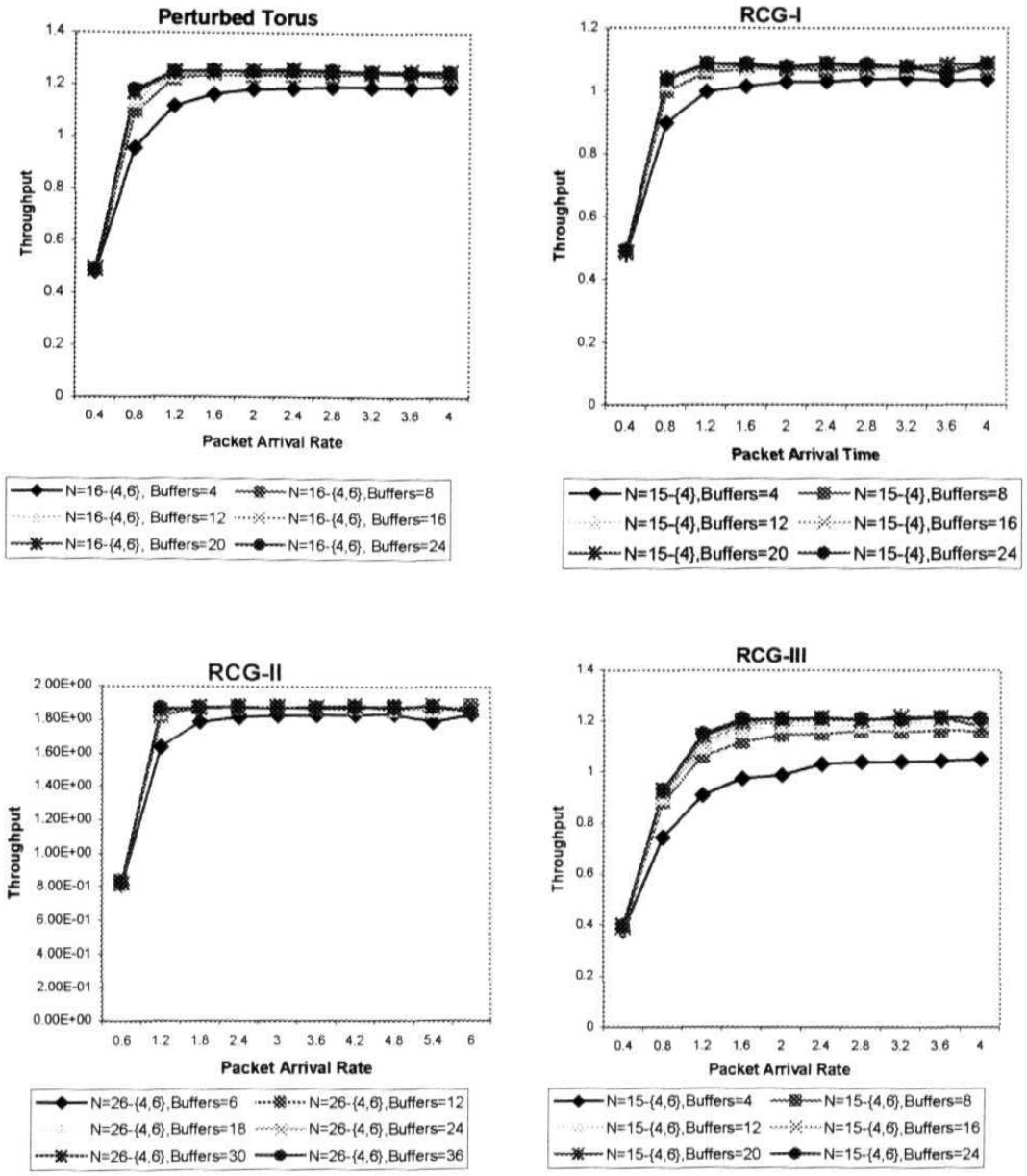


Figure 5.9 Throughput Vs. Packet arrival rate for different buffer sizes in a configuration of reconfigurable topologies.

E.V.R.C. Mohan Reddy and K.C. Reddy, Reconfigurable Circulant Graph: A Logical Topology for Parallel Architectures and Optical LANs, Proc. 6th National Seminar on Theoretical Computer Science, Banasthali, India, pp. 143-156, 1996b.

E.V.R.C. Mohan Reddy and K.C. Reddy, A Dynamically Reconfigurable WDM LAN Based On Reconfigurable Circulant Graph, Proc. IEEE MILCOM, Virginia, Vol. 3, pp. 786-790, 1996c.

B.Mukherjee, WDM-Based Local Lightwave Networks Part I: Single-Hop Systems, IEEE Networks, pp. 12-27, May 1992a.

B.Mukherjee, WDM-Based Local Lightwave Networks Part II: Multihop Systems, IEEE Networks, pp. 20-32, July 1992b.

B.Mukherjee, Optical Communication Networks, McGraw-Hill, 1997.

M.M.Nassehi, CRMA: An Access Scheme for High-Speed LANs and MANs, Proceedings of IEEE ICC'90, pp. 1697-1702, Apr. 1990.

Y.Ofek and M.Yung, Metanet: Principles of an Arbitrary Topology LAN, IEEE/ACM Transactions on Networking, Vol. 3, No. 2, pp. 169-180, April 1995.

J.Opatrny and D. Sotteau, Linear Congruential Graphs, Graph Theory, Combinatorics, Algorithms and Applications, SIAM Proc. Series, pp.404-426, 1991.

J.Opatrny, D. Sotteau, N. Srinivasan and K. Thulasiraman, DCC Linear Congruential Graphs: A New Class of Interconnection Networks, IEEE Transactions on Computers, Vol. 45, No. 2, pp. 156-164, February 1996.

G.Panchapakesan and A. Sengupta, On Multihop Optical Network Topology Using Kautz Digraphs, Proceedings of IEEE INFOCOM, pp. 675-682, April 1995.

R.Ramaswami and K.N. Sivarajan, A Packet-Swithched Multihop Lightwave Network Using Subcarrier and Wavelentgh Division Multiplexing, IEEE Transactions Communications, Vol. 42, No. 2/3/4, pp. 1198-1211, April 1994.

R Ramaswami and K.N. Sivarajan, Design of Logical Topologies for Wavelength-Routed Optical Networks, Proceedings of IEEE TNFOCOM, 1995.

E.M. Reddy and K.C. Reddy, Local Perturbations: A paradigm for logical topology reconfiguration, Communicated to ACM/IEEE Transactions on Networking, 2001a.

K.C. Reddy and E.M. Reddy, A Framework for evaluation of Logical Topologies, Communicated to IEEE Transaction of Computers, 2001b.

D.A Reed and H.D. Schwetman, Cost-Performance Bounds for Multicomputer Networks, IEEE Transactions on Computers, Vol. C-32, pp.83-95, January 1983.

5.4.2 Delay Analysis

This section presents average queuing delays in reconfigurable topologies against packet arrival rate. Queuing delay is measured as the time spent by a packet at each intermediate node in the insertion queue before reaching destination node. Figure 5.11, Figure 5.12 and Figure 5.13 analyze the average queuing delay of the networks based on reconfigurable topologies for different packet arrival rates by varying the size of the network, configurations of logical topology and the number of packet buffers in transmission and insertion queues. From these figures, it can be observed that as the arrival rate increases, the queuing delay will increases exponentially because only limited number of packets will be transmitted by each node in one time slot. However as queuing delays are shown to be stabilized because packets generated by the node are discarded, when the transmission queue is full. If the size of transmission queue is unlimited, the queuing delays will always increase. However, in practice, buffers are always limited. This is clear from the study of buffer size on queuing delay shown in Figure 5.13. Unlike throughput, average queuing delay will be same irrespective of whether a single buffer is used or not, for the packets generated at the node and for the packets being routed through the node.

Figure 5.11 depicts average queuing delay for various sizes of reconfigurable topologies under different packet arrival rates. It is observed that the average queuing delay is more for larger networks. This is because the average internode distance is more for larger networks.

Figure 5.12 plots average queuing delays for two different structures of reconfigurable topologies of same size. It is observed that the configuration (connectivity of nodes) of a reconfigurable topology sometimes affects the queuing delays.

Figure 5.14 provides comparison of queuing delays in the proposed reconfigurable topology networks (Perturbed Torus, RCG-1, RCG-II and RCG-III) and their counterparts in traditional logical topologies (Torus and Hypercube). The networks are compared with almost same size. The buffer sizes of transmission queue and insertion queue is taken as 12. It is observed that the queuing delays of the networks based on RCG-II, RCG-III and Hypercube topologies are less than rest of topologies compared. The

queuing delays of the network based on Perturbed Torus topology are same as that of network based on 2-D Torus topology.

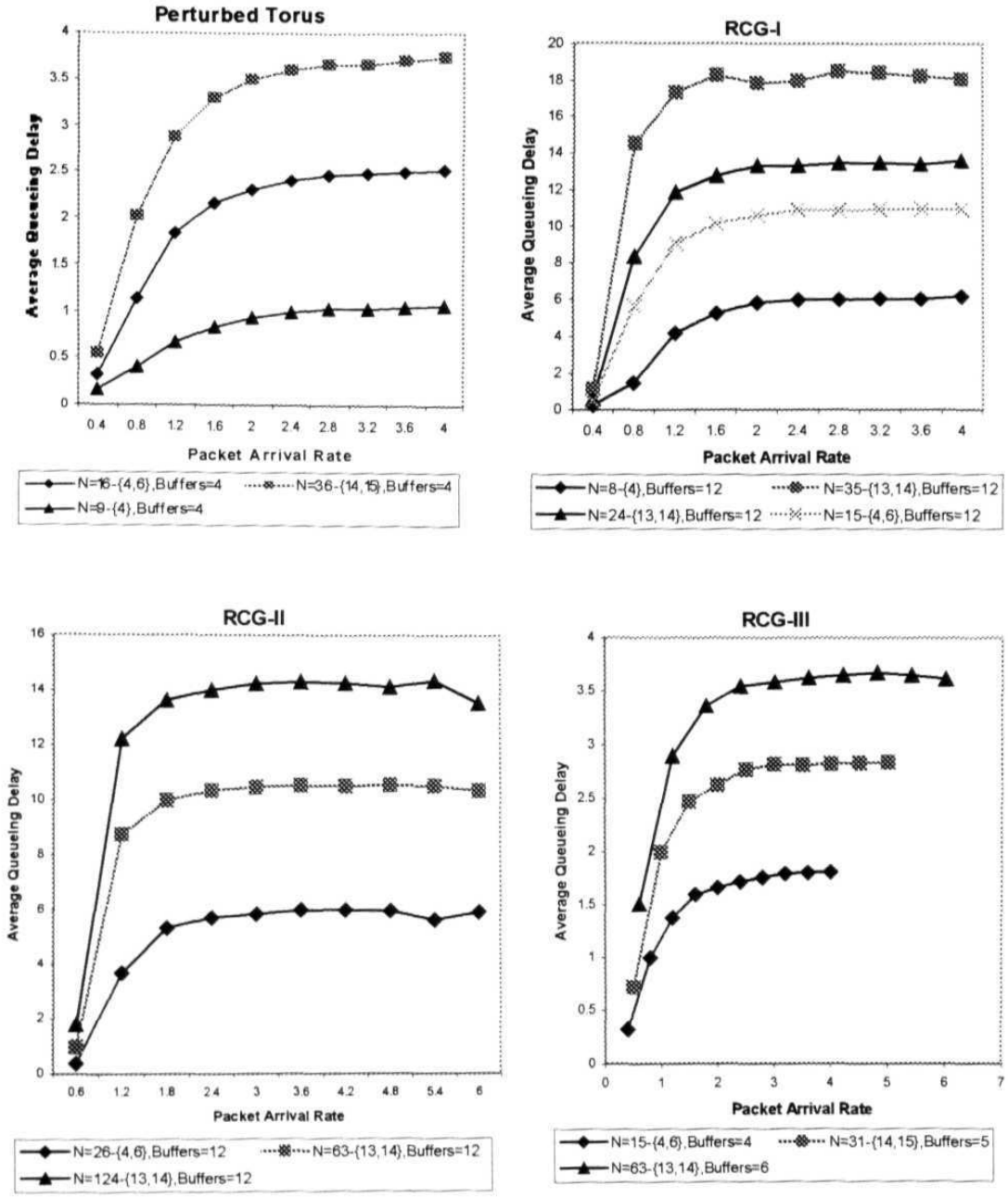


Figure 5.11 Average Queueing Delay Vs. Packet arrival rate for different sizes of reconfigurable topologies

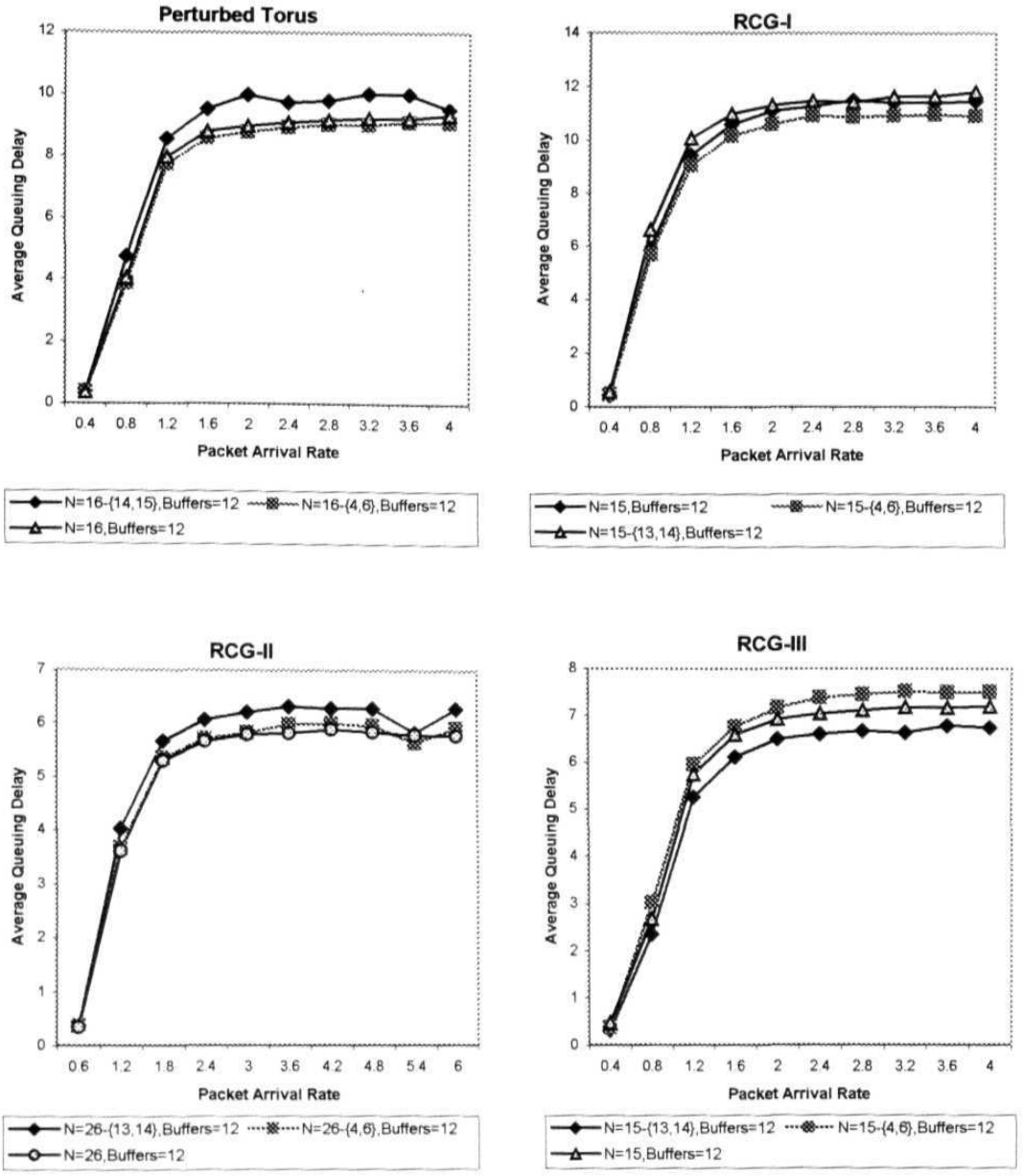


Figure 5.12 Average Queueing Delay Vs. Packet arrival rate for reconfigurable topologies of same size, but different structures

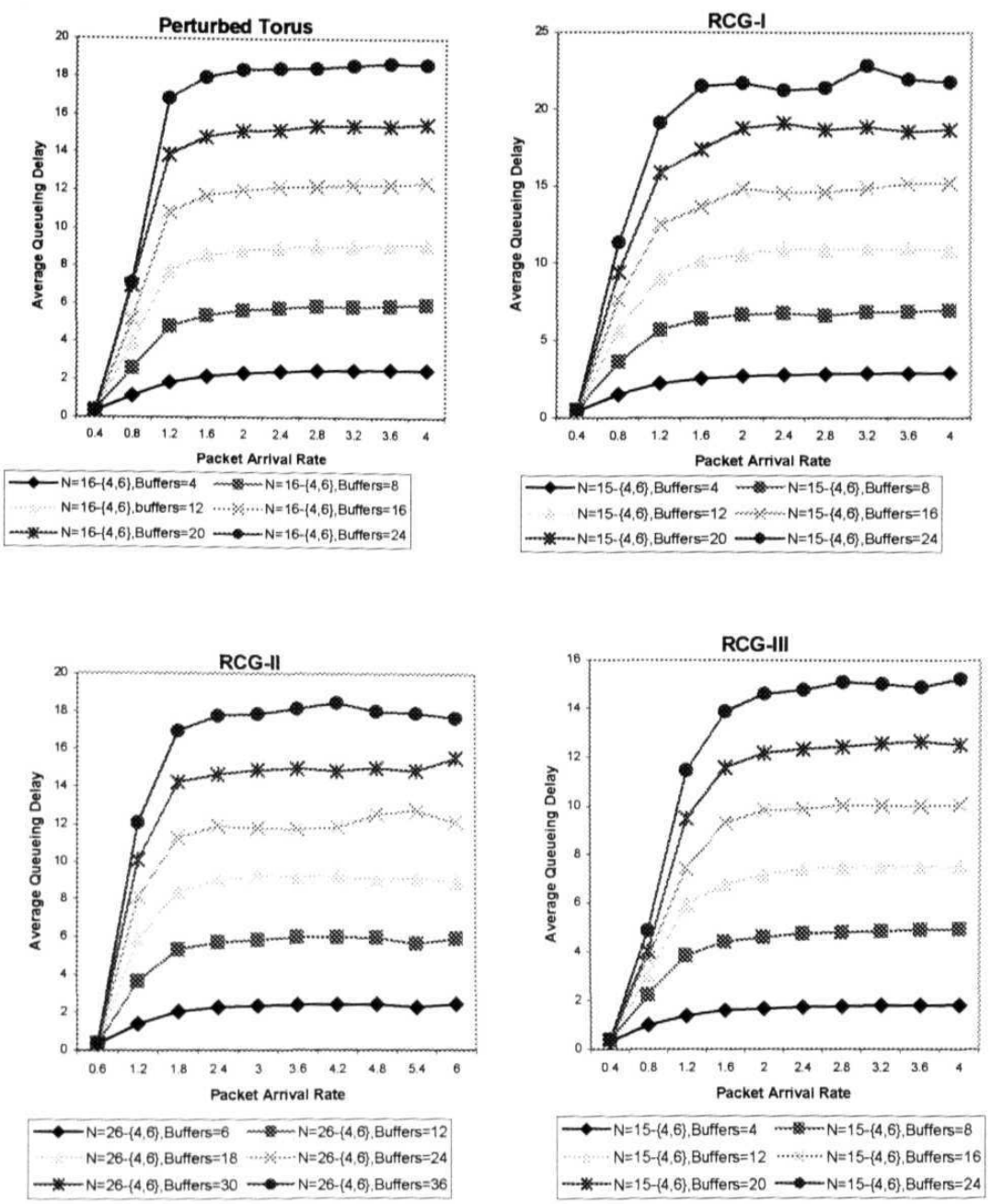


Figure 5.13 Average Queue ing Delay Vs. Packet arrival rate for different buffer sizes in a configuration of reconfigurable topologies.

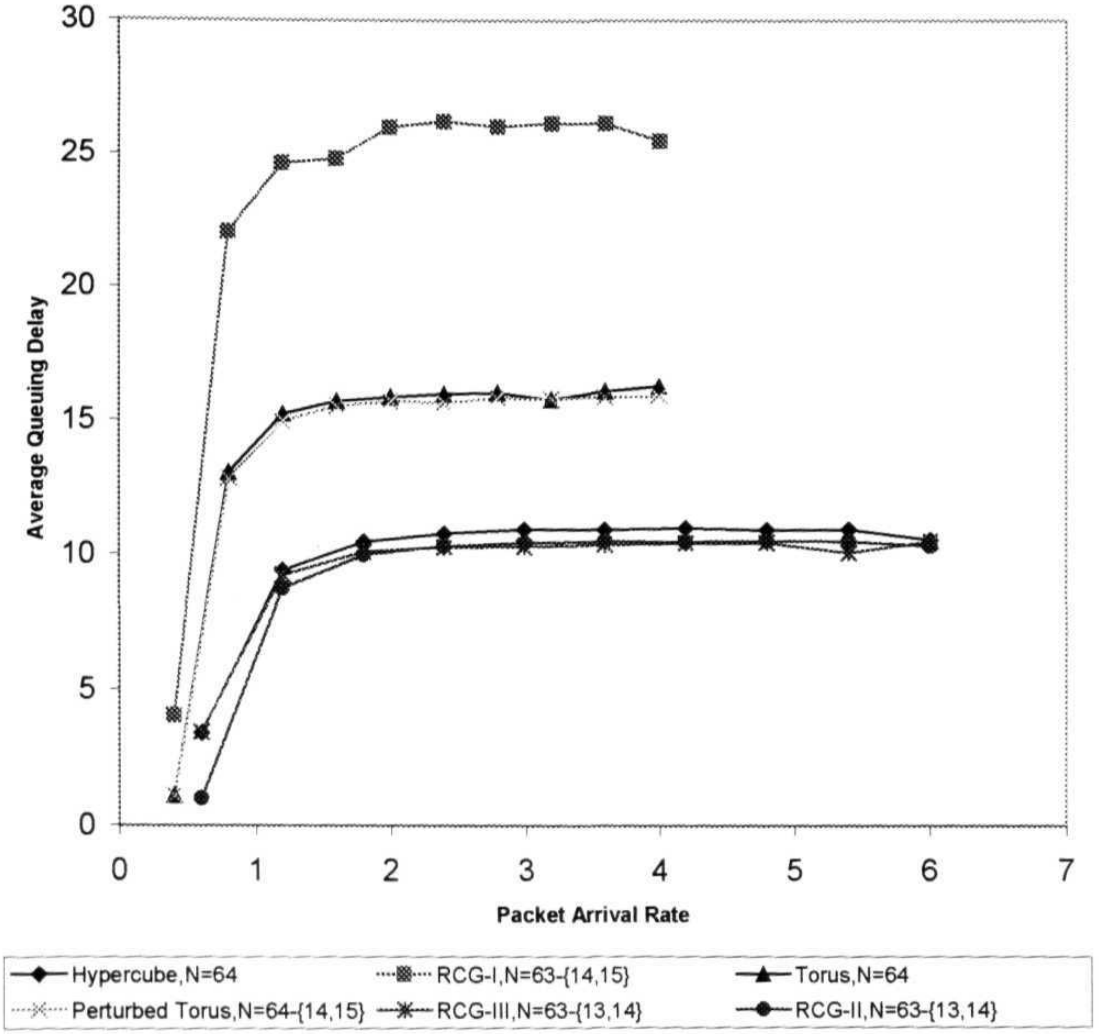


Figure 5.14 Average Queuing Delay vs. Packet Arrival Rate for different networks of same size

5.4.3 Utilization Analysis

In this section, utilization of channels in reconfigurable topologies is studied against packet arrival rate. Channel utilization is measured as the fraction of channels used for transmission out of the channels available. Figure 5.15, Figure 5.16 and Figure 5.17 analyze the utilization of channels in the networks based on reconfigurable topologies for different packet arrival rates by varying the size of the network, configurations of logical topology and the number of packet buffers in transmission and insertion queues. From these figures, it can be observed that as the arrival rate increases, the channel utilization will first increase as more number of packets becomes available for transmission, and then eventually stabilizes because each node has limited number of transmitters.

Figure 5.15 depicts channel utilization for various sizes of reconfigurable topologies under different packet arrival rates. It is observed that the size of network doesn't affect channel utilization. This is because packets are routed based on the routing algorithm and hence channel utilization depends mostly on routing algorithm.

Figure 5.16 plots the channel utilization for two different structures of reconfigurable topologies of same size. It is observed that the configuration (connectivity of nodes) of a reconfigurable topology doesn't affect channel utilization substantiating our earlier argument that channel utilization depends mostly on routing algorithm. This is further substantiated with the study of the effect of buffer size on channel utilization shown in Figure 5.17.

Figure 5.18 shows the comparison of utilization of channels in the proposed reconfigurable topology networks (Perturbed Torus, RCG-I, RCG-II and RCG-III) and their counterparts in traditional logical topologies (Torus and Hypercube). The networks are compared with almost same size. The buffer sizes of transmission queue and insertion queue is taken as 12. It is observed that the channel utilization in the networks based on Torus and Perturbed Torus topologies is better than rest of topologies compared. However, channel utilization in RCG-I, RCG-II and RCG-III topology networks is better than Hypercube network.

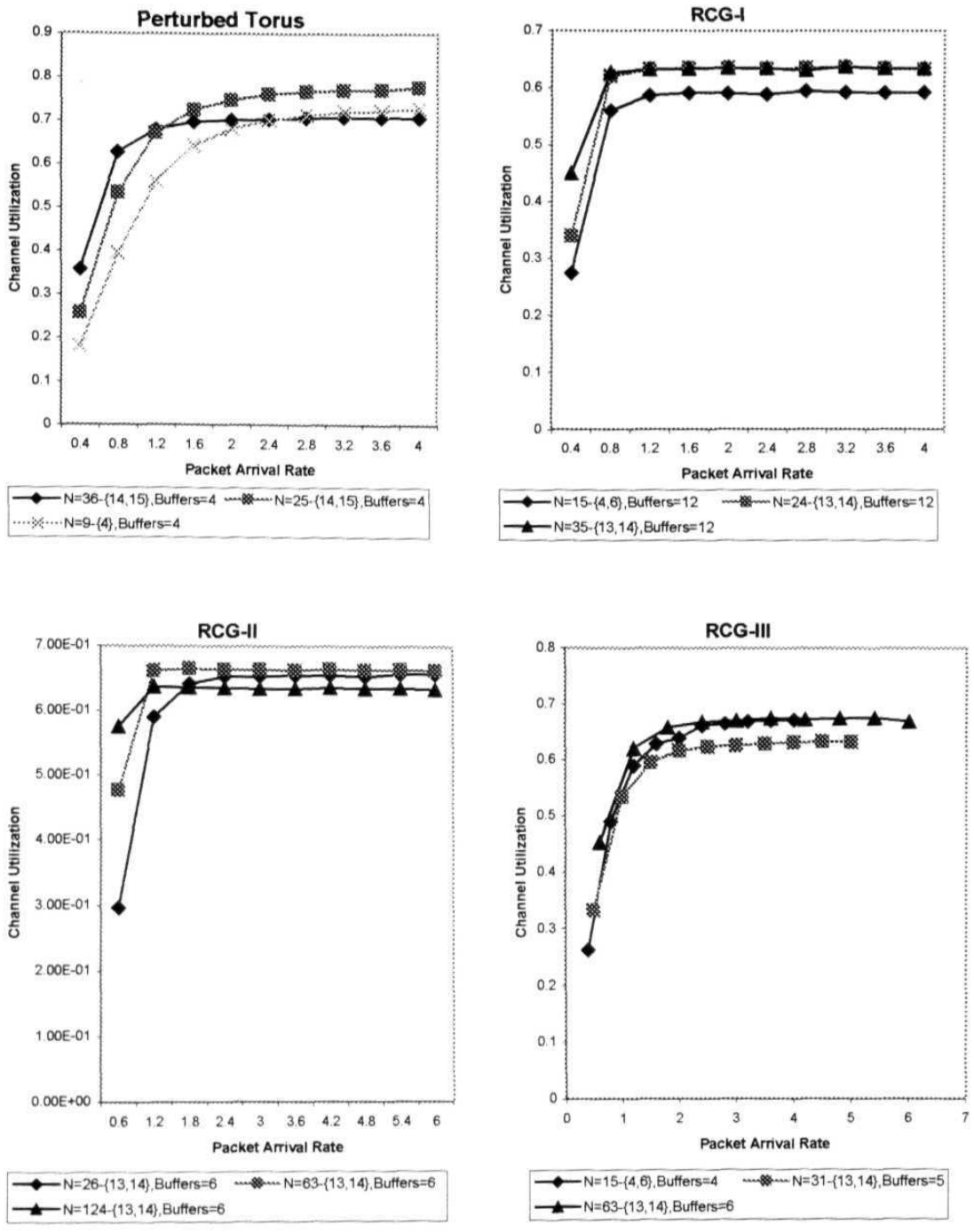


Figure 5.15 Channel Utilization Vs. Packet arrival rate for different sizes of reconfigurable topologies

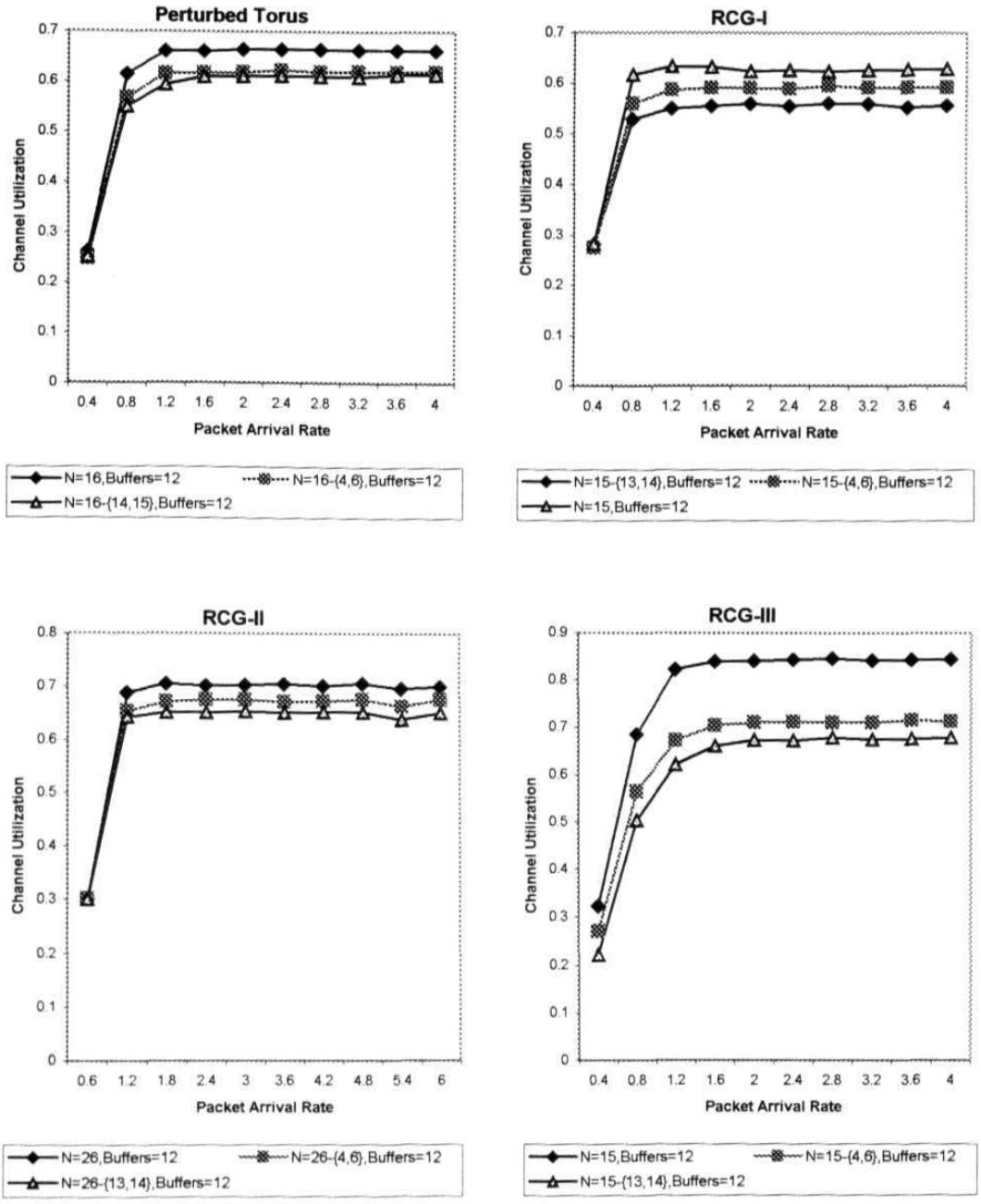


Figure 5.16 Channel Utilization Vs. Packet arrival rate for reconfigurable topologies of same size, but different structures

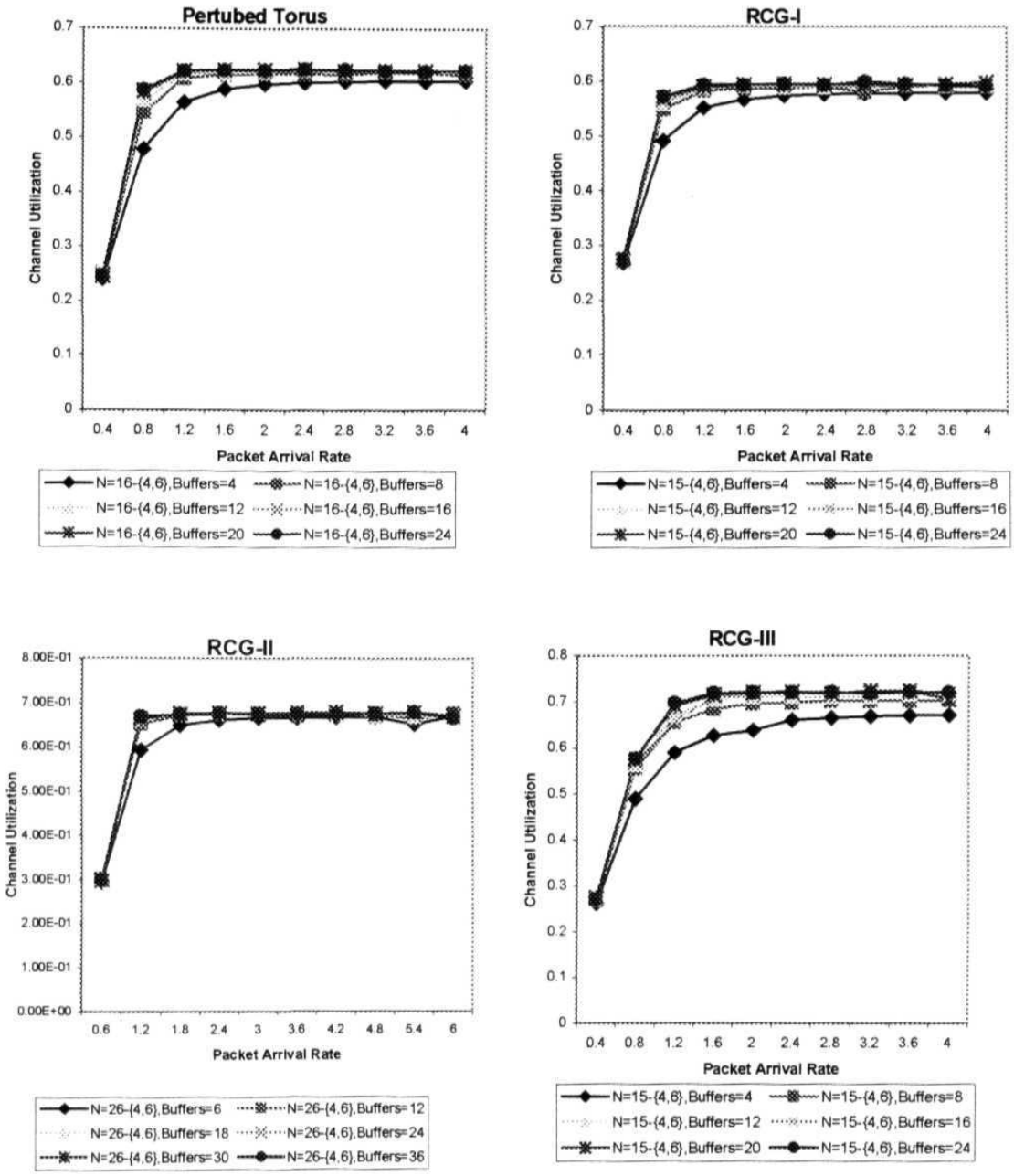


Figure 5.17 Channel Utilization Vs. Packet arrival rate for different buffer sizes in a configuration of reconfigurable topologies.

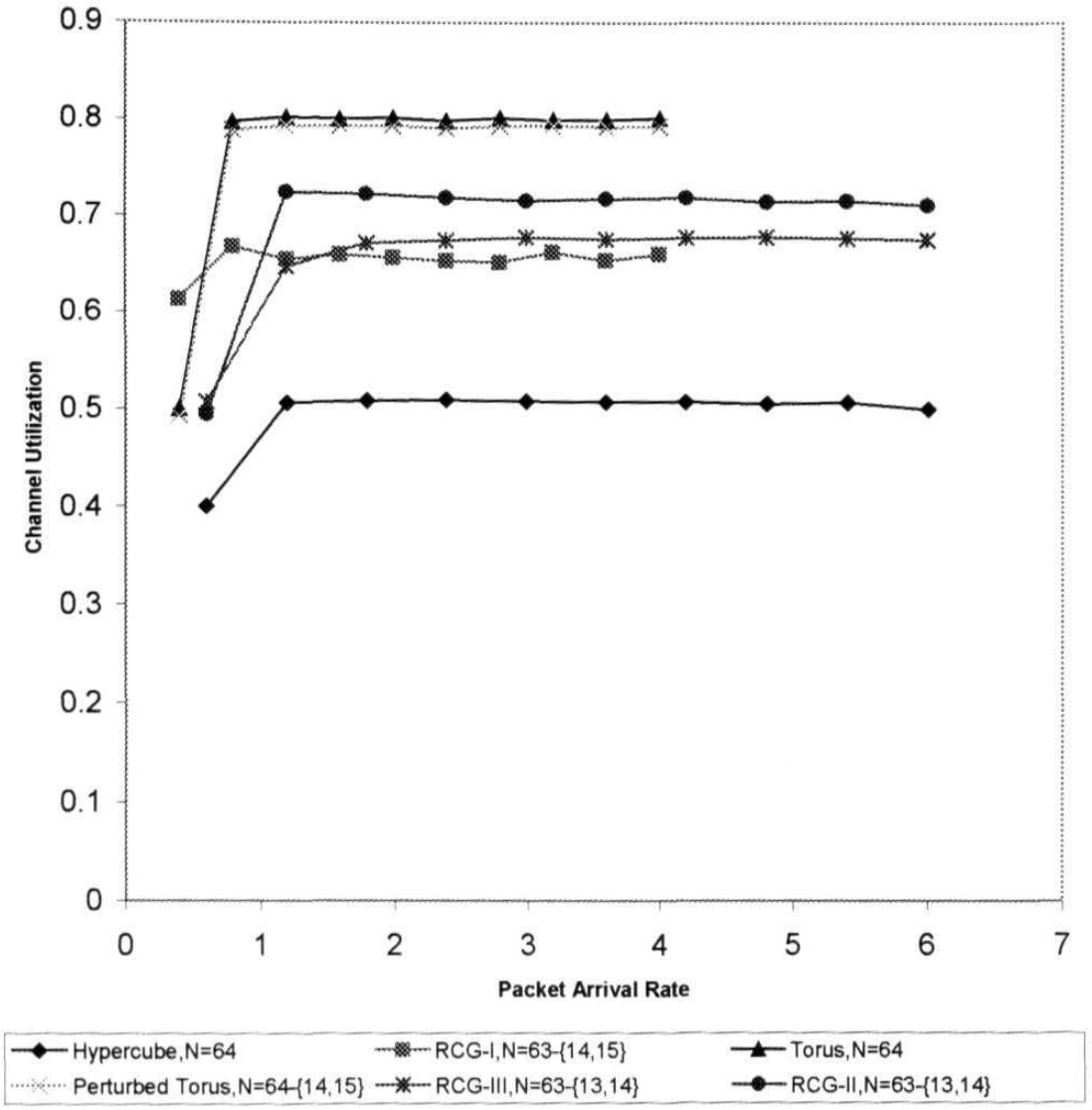


Figure 5.18 Channel Utilization vs. Packet Arrival Rate for different networks of same size

5.5 Discussion

Broadcast and select architectures and shared channel architectures discuss the implementation details of fiber optic networks based on newly proposed dynamically reconfigurable topologies. Broadcast and select architectures are simple in nature and hence are easy to implement. *However, the size of the base topology for a reconfigurable topology in the broadcast and select architectures is restricted by the limitations of optical components characteristics such as tuning range of a transmitter/ receiver.* Also, with each reconfiguration due to node failure, channels equivalent to the degree of node become unusable. *Shared channel architectures with the number of channels less than the size of base topology of reconfigurable topology overcome these problems as more nodes share a channel for transmission as well for reception.*

Simulation results of reconfigurable topology networks show that the throughput, queuing delay and channel utilization will first increase along with the increase in packet arrival rate, and then eventually stabilizes because each node has limited transmitters and packet buffers. It is also observed that network throughput and queuing delay increase as the size of the network increases. The number of packet buffers in transmission and insertion queues play limited role in deciding throughput and channel utilization.

The results also show that the performance of reconfigurable topology networks vary for different configurations of reconfigurable topologies. *However, the performance of reconfigurable topologies is limited to the performance of networks of respective base topologies.*

It is also observed that the performance of reconfigurable topology based networks proposed (Perturbed Torus, RCG-I, RCG-II, and RCG-III) is comparable to their traditional topology counterparts (Torus and Hyper cube).

Chapter

6. Conclusions

This chapter summarizes the results of the research and discusses scope for future directions. Section 6.1 summarizes the results and Section 6.2 discusses future direction to this research.

6.1 Summary

This research has been concerned with new architectures for local and metropolitan area networks that try to fully exploit the properties of lightwave technology. The mismatch between operating speeds of access nodes and potential bandwidth of optical fiber led to the research into the new generation of networks wherein parallel transmission among access nodes is done through multiple channels obtained by dividing the optical bandwidth using Wavelength Division Multiplexing.

Among multichannel networks, in multihop architectures nodes communicate using a predefined connectivity pattern, called as logical topology, created by the assignment of wavelengths to transmitter/ receiver pairs of nodes. The independence of logical topology and broadcast physical topologies used by LAN/MANs, and technological advances in slowly tunable lasers and optical filters allow the logical topology to be dynamically reconfigured according to changes in the network such as node failures and recovery from failures (or addition of node). Since network changes occur randomly, reconfiguration should be done in a transparent way so that the network traffic is minimally disrupted. Also, the reconfiguration process should be fast enough to tolerate even high frequency of changes to the network.

It is observed that reconfiguration affects the network performance parameters such as reliability and expandability similar to the way the other properties of logical topology such as diameter and average internode distance affects network throughput and packet delays. It is also proved that reconfiguration and properties of topology such as node

degree are interdependent. *The primary contribution of this research has thus been the consideration of reconfiguration as a design issue of the logical topology.*

A second contribution was made in the area of reconfiguration by proposing local perturbations paradigm. According to this paradigm, each node in the network occupies a unique position in a regular structure called as base topology. Reconfiguration is defined by changing the links of the neighbors of failed/ added node so that the resultant structure remains connected, and most importantly preserves all the properties of base topology. The resultant topology is called as dynamically reconfigurable topology, because it is resilient to node changes.

The third one was the study of the applicability of Local Perturbations paradigm to traditional topologies such as Torus and Hypercube. A reconfigurable topology called Perturbed Torus has been designed based on 2-D Torus. The diameter and average internode distance of Perturbed Torus was found to be on par with that of 2-D base Torus structure. However, it was found that applying Local Perturbations to traditional topologies is a difficult task.

The fourth one was the design of reconfiguration topologies that reconfigure using Local Perturbations paradigm. The topologies RCG-I, RCG-II and RCG-III were designed with base structures constructed using a set of edge-disjoint Hamiltonian circuits. These topologies reconfigure along the Hamiltonian circuits to retain the connectivity. By the theoretical evaluation of these topologies, the upper limits of the diameter were found to be slightly more than the diameter of respective base topologies. However, the empirical results showed that the diameter is always closer to that of base topology of corresponding size. *Most interestingly, to out surprise, RCG-I, RCG-II and RCG-III are found to have structural properties comparable to the traditional topologies 2-D Torus, n-D Torus and Binary Hypercube, respectively.*

In order to evaluate the logical topologies, this work proposed a framework for evaluating logical topologies. Through this framework, all the topologies discussed in this thesis are evaluated. It was noticed that reconfigurable topologies have edge over their traditional topologies such as Torus and Binary Hypercube because of their ability to reconfigure dynamically.

The other contribution was made in the area of optimal node placements in regular logical topologies. Two heuristic algorithms based on greedy approach were developed. These algorithms served as yardstick for comparing the values of flow-weighted averages of internode distances, under non-linear traffic conditions.

The final contribution was the design of fiber optic architecture for LAN/MANs based on the reconfigurable topologies. It was proved that simple broadcast and select architectures and shared channel architectures were feasible to construct. The simulation results confirmed that their performance is comparable to their counterparts.

6.2 Future Directions

The successful design of reconfigurable logical topologies, i.e., by considering reconfiguration as a design issue opens up more interesting and challenging problems for future research in this area.

The work presented in this thesis designed logical topologies that have properties similar to 2-D Torus, n-D Torus and Binary Hypercube. These topologies use the simple concept of Hamiltonian circuit to maintain connectivity and reduce the distance to the destination with each hop. *This work could be extended by making use of Hamiltonian circuit connecting all nodes sequentially and devising other links so that the properties such as node degree, diameter and average internode distance would be on par with traditional topologies such as deBruijn graph and Star graph.*

Another extension of this work could investigate the use of reconfigurable topologies for parallel computer architectures as processor interconnection networks. One could easily observe that reconfigurable topologies provide simple alternative for fault tolerance that would facilitate the effective use of all active processors for assigning task graphs. Hence, no redundancy of time or processor is required.

Yet another extension of this work is to provide better lower bounds for the diameter of reconfigurable topologies designed in this work. It is observed from the empirical results that the diameter of logical topology is always lower than that of base topology.

It is known that the routing algorithms play critical role in the network performance. Though, it may not possible to design shortest path routing without having the global knowledge of the network, probably one could investigate better algorithms than those proposed in this thesis.

Another interesting extension of this is to develop analytical model for the broadcast and select architectures and shared channel architectures.

References

A.S.Acampora, A Multichannel Multihop Lightwave Network, Proceedings of IEEE GLOBECOM, pp.1459-1467, November 1987.

S.B.Akers and B. Krishnamurthy, A Group-theoretic Model for Symmetric Interconnection Networks, IEEE Transactions on Computers, Vol. 38, No. 4, pp.555-566, April 1989.

F.Ayadi, J.F. Hayes and M. Kavehrad, Bilayered ShuffleNet: A New Logical Configuration for Multihop Lightwave Networks, Proceedings of IEEE GLOBECOM, Vol. 2, pp. 1159-1163, 1993.

D.Banerjee, B. Mukherjee and S. Ramamurthy, The Multidimensionsl Torus: Analysis of Average Hop Distance and Application as a Multihop Lightwave Network, IEEE Intl. Conf. on Commun. Vol. 3. pp. 1675-1680, 1994.

S.Banerjee and B. Mukherjee, Algorithms for Optimized Node Arrangements in ShuffleNet Based Lightwave Networks, IEEE, pp. 557-564, 1993.

S.Banerjee, V Jain and S. Shah, Regular Multihop Logical Topologies for Lightwave Networks, IEEE Communication Surveys, pp.2-18, First Quarter 1999.

S.Banerjee and D. Sarkar, Heuristic Algorithms for Constructing Near-Optimal Structures of Linear Multihop Lightwave Networks, Proc. INFOCOM, pp. 671-680, 1992.

J.A.Bannister, L. Fratta and M. Gerla, Topological Design of the Wavelength-Division Optical Network, Proceedings of IEEE INFOCOM, pp. 1005-1013, 1990.

L.N.Bhuyan and DP. Agarwal, Generalized Hypercube and Hyperbus Strctures for a Computer Network, IEEE Transactions on Computers, Vol. C-33, No. 4, pp. 323-333, Apr. 1984.

J.Brassil, A.K. Choudhury and N.F. Maxemchuk, The Manhattan Street Network: a high performance, highly reliable metropolitan area network, Computer Networks and ISDN Systems, Vol. 26, pp. 841-858, 1994.

J.Bruck, R Cypher and C-T. Ho, Fault-Tolerant Meshes and Hypercubes with Minimal Number of Spares, IEEE Transcations on Computers, Vol. 42, No. 9, pp. 1089-1094, September 1993.

J.Bruck, R Cypher and C-T. Ho, Fault-Tolerant de Bruijn and Shuffle-Exchange Networks, IEEE Transcations on Parallel and Distributed Systems, Vol. 5, No. 5, pp. 548-553, May 1994.

J.Bruck, R. Cypher and C-T Ho, On the Construction of Fault-Tolerant Cube-Connected Cycles Networks, Journal of Parallel and Distributed Computing, Vol. 25, pp. 98-106, 1995.

P.Camarda and M Cocolicchio, Virtual Topology Design of Multichannel Multihop Lightwave Networks, Computer Communications, Vol. 15, No. 7, pp. 458-466, September 1992.

F.B.Chedid, On the Generalized Twisted Cube, Information Processing Letters, Vol.55, pp. 49-52, 1995.

I.Chlamtac and A. Farago, Making Transmission Schedule Immune to Topology Changes in Multi-Hop Packet Radio Netwroks, IEEE/ACM Transactions on Networking, Vol. 2 No. 1, pp. 23-29, February 1994.

I.Chlamtac and A. Ganz, Toward Alternative High-Speed Network Concepts: The SWIFT Architecture, IEEE Transactions on Communications, Vol. 38, No. 4, pp. 431-439, April 1990.

S.P.Dandamudi and D.L. Eager, On Hypercube-based Hierarchical Interconnection Network Design, Journal of Parallel and Distributed Computing, Vol. 12, pp. 283-289, 1991.

S.K.Das and A.K Banerjee, Hyper Petersen Network: Yet Another Hypercube-Like Topology, IEEE, pp. 270-277, 1992.

K Day and A. Thripathi, A Comparitive Study of Topological Properties of Hypercubes and Star Graphs, IEEE Transactions on Parallel and Distributed Systems, Vol. 5, No. 1, pp.31-38, January 1994.

N.J.Dimopoulos, D. Radvan and K.F. Li, Performance Evaluation of the Backtrack-To-Origin-And-Retry Routing for Hypercycle-Based Interconnection Networks, Proceedings of IEEE GLOBECOM, pp. 278-284, 1990.

P.W.Dowd, Random Access Protocols for High Speed Inter-Processor Communication Based on an Optical Passive Star Topology, IEEE/OSA Journal of Lightwave Technology, Vol. 9, pp.799-808, June 1991.

P.W.Dowd, Wavelength Division Multiple Access Channel Hypercube Processor Interconnection, IEEE Transactions on Computers, 1992.

D.Z.Du and F.K. Hwang, Generalized de Bruijn Digraphs, Networks, Vol. 18, pp.27-38, 1988.

T-y.Feng, A Survey of Interconnection Networks, IEEE Computer, pp. 12-27, December 1981.

Z.Feng and L.W.W. Yang, Routing Algorithms in the Bidirectional De Bruijn Graph Metropolitan Area Networks, IEEE, pp. 957-961, 1994.

E.Ganesan and D.K. Pradhan, The Hyper-deBruijn Networks: Scalable Versatile Architecture, IEEE Transactions on Parallel and Distributed Systems, Vol. 4, No. 9, pp. 962-978, Sep. 1993.

A.Ganz, W. Gong and X. Wang, Wavelength Assignment in Multihop Lightwave Networks, IEEE Transactions on Communications, Vol. 42, No. 7, pp. 2460-2469, July 1994.

A. Ganz and B. Li, Broadcast-Wavelength Architecture for a WDM Passive Star-Based Local Area Network, Proceedings of IEEE International Conference on Communications, pp.837-842, June 1992.

A.Ganz and Z. Koren, WDM Passive Star-Protocols and Performence Analysis, IEEE, pp. 991-1000, 1991.

P.T.Gaughan and S Yalamanchili, Adaptive Routing Protocols for Hypercube Interconnection Networks, IEEE Computer, pp. 12-22, May 1993.

P.E.Green, Fiber Optic Networks, Prentice Hall, 1993.

M.Hamdi, Topological Properties of the Directional Hypercube, Information Processing Letters, Vol. 53, pp. 277-286, 1995.

IEEE Std 802.6/D15, Distributed Queue Qual Bus (DQDB) Subnetwork of a Metropolitan Area Network (MAN), IEEE Std 802.6/D15, Dec. 1990.

J.Iness, S Banerjee and B. Mukherjee, GEMNET: A Generalized, Shuffle-Exchange-Based, Regular, Scalable, Modular, Multihop, WDM Lightwave Network, IEEE/ACM Transactions on Networking, Vol. 3, No. 4, pp.470-476, August 1995.

ISO 9312-2/1989, FDD1 Token Ring Media Access Control (MAC), ANSI X3T9.5, Doc. X3.139/1987 (ISO 9312-2/1989).

H.P.Katseff, Incomplete Hypercubes, IEEE Transaction on Computers, Vol. C-37, pp.604-608, May 1988.

W.H. Kautz, Design of Optimal Interconnection Networks for Multiprocessors, Architecture and Design of Digital Computers, NATO Advanced Summer Institute, pp. 249-272, 1969.

L.G.Kazovsky, C. Barry, M. Hickey, C.A. Noronha and P. Poggiolini, WDM Local Area Networks, IEEE LTS, pp. 8-15, May, 1992.

M. Kovacevic, M. Gerla and J.A. Bannister, On the Performance of Shared-Channel Multihop Lightwave Networks, Proceedings of IEEE INFOCOM, pp. 544-551, April 1995.

A.Khurshid, D.M. Rouse and R.A. Widlicka, Photonic Switching Ring-Based Optic Networks, IEEE, pp. 867-876, 1989.

J-F.P.Labourdette, Rearrangeability Techniques for Multihop Lightwave Networks and Applicaion to Distributed ATM Switching Systems, Ph.D. Dissertation, Columbia University, 1991.

J-F.P.Labourdette and A.S. Acampora, Logically Rearrangeable Multihop Lightwave Networks, IEEE Transactions on Communications, Vol. 39, No. 8, August 1991.

S.Lakshmivarahan, J.S. Jwo and S.K. Dhall, Symmetry in Interconnection Networks Based on Cayley Graphs of Permutation Groups: A Survey, Parallel Computing, Vol. 19, pp.361-407, 1993.

S.Latifi, On the Fault-Diameter of the Star Graph, Information Processing Letters, Vol. 46, pp. 143-150, 1993.

S.Latifi and P. K. Srimani, Transposition Networks as a Class of Fault-Topelant Robust Networks, IEEE Transactions on Computers, Vol. 45, No. 2, pp. 230-238, 1996.

B.Li and A. Ganz, Virtual Topologies for WDM Star LANs: The Regular Structure Approach, Proceedings of IEEE INFOCOM, pp.2134-2143, May 1992.

G.Liu, K.Y. Lee and H.F. Jordan, Hierarchical Networks for Optical Communications, IEEE Intl. Conf. on Commun., Vol. 3, pp. 1664-1668, 1994a.

G.Liu, K.Y. Lee and H.F. Jordan, TDM Hypercube and TWDM Mesh Optical Interconnections, Proceedings of IEEE GLOBECOM, Vol. 3, pp. 1953-1957, 1994b.

Q.M.Malluhi and M.A. Bayoumi, The Hierarchical Hypercube. A New Interconnection Topology for Massively Parallel Systems, IEEE Transctions on Parallel and Distributed Systems, Vol. 5, No.1, pp. 17-30, Jan. 1994.

N.F.Maxemchuk, Compasion of Deflection and Store-and-Forward Techniques in the Manhattan Street and Shuffle-Exchange Networks, IEEE, pp. 800-809, 1989.

E.V.R.C. Mohan Reddy and K.C. Reddy, Perturbed Torus: A Dynamic Topology for Multichannel Multihop Lightwave Networks, Software Bulletin, pp. 2- 10, June 1996a.

Y.Saad and M.H. Schultz, Topological Properties of Hypercubes, IEEE Transactions on Computers, Vol. C-37, pp.867-872, July 1988.

A. Sen and P. Maitra, A Comparative Study of ShuffleExchange, Manhattan Street and Supercube Network for Lightwave Applications, Computer Networks and ISDN Systems, Vol. 26, 00. 1007-1022, 1994.

K.Sivarajan and R Ramaswami, Multihop Lightwave Networks Based on deBruijn Graphs, IEEE, pp. 1001-1011, 1991.

K.N.Sivarajan and R. Ramaswami, Lightwave Networks Based on de Bruijn Graphs, IEEE/ACM Transactions on Networkings, Vol. 2, No. 1, pp. 70-79, February 1994.

S.Sur and P.K. Srimani, Incrementally Extensible Hyptercube (1EH) graphs, Proc. of IEEE IPCCC, pp 1-7, 1992.

T.Szymanski, "Hypermeshes": Optical Interconnection Networks for Parallel Computing, Journal of Parallel and Distributed Computing, Vol. 26, pp. 1-23, 1995.

S.T.Tan and D.H.C Du, Embedded Unidirectional Incomplete Hypercubes for Optical Networks, IEEE Transactions on Computers, Vol. 41, No. 9, pp. 1284-1289, Sep. 1993.

K.W.Tang and B.W. Arden, Representations of Borel Cayley Graphs, IEEE, pp. 194-201, 1992.

F.A.Tobagi, F. Borgonovo and L. Fratta, Expressnet: A High Performance lntegrated-Services Local Area Networks, IEEE Journal on Selected Areas in Communications, Vol. 1, No.5, pp. 898-913, 1983.

S.R.Tong and D.H.C. Du, Designing Multi-Hop Optical Networks Using Fixed Wavelength Transceivers, Proceedings of IEEE GLOBECOM, Vol. 3, pp. 1538-1543, 1994.

N.F.Tzeng, Structural Properties of Incomplete Hypercube Computers, Proceedings of IEEE GLOBECOM, Vol. 1, pp. 262-269, 1990

P.Vadapalli and P.K. Srimani, A New Family of Cayley Graph Interconnection Netwokrs of Constant Degree Four, IEEE Transactions on Parallel and Distributed Systems, Vol. 7, No. 1, pp. 26-32, January 1996.

H.R.van As, Media access techniques: The evolution towards terabit/s LANs and MANs, Computer Networks systems and ISDN systems, Vol 26, pp. 603-656, 1994.

L.Wang and K-W. Hung, Augmented Shuffle Net Multihop Lightwave Networks, IEEE, 13th Annual International Phoenix Conference on Communications, pp. 465-471, 1994.

L.D. Wittie, Communication Structures for Large Networks of Microcomputers, IEEE Transactions on Computers, Vol. C-30, No. 4, pp. 264-273, Apr. 1981.

K.L. Yeung and T-S.P. Yum, Node Placement Optimization in ShuffleNets, IEEE/ACM Transactions on Networking, Vol. 6, Mo. 3, pp. 319-324, June 1998.

Z.Zhang and A. Acampora, A Heuristic Wavelength Assignment Algorithm for Multihop WDM Networks with Wavelentgh Routing and Wavelength Reuse, Proceedings of IEEE INFOCOM, Vol. 2, pp. 534-543, 1994.