

D 65-92
PAW

FAULT-TOLERANT MINs FOR PARALLEL PROCESSING

A THESIS

submitted in fulfilment of the
requirements for the award of the degree

of

DOCTOR OF PHILOSOPHY

in

ELECTRONICS AND COMPUTER ENGINEERING

By

PAWAN KUMAR



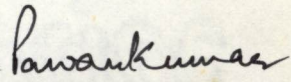
DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING
UNIVERSITY OF ROORKEE
ROORKEE-247 667 (INDIA)

APRIL, 1992

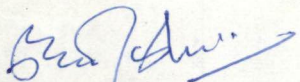
CANDIDATE'S DECLARATION

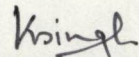
I hereby certify that the work which is being presented in the thesis entitled **Fault-Tolerant MINs for Parallel Processing** in fulfilment of the requirement for the award of the Degree of Doctor of Philosophy and submitted in the Department of Electronics and Computer Engineering of the University is an authentic record of my own work carried out during a period from October 1988 to April 1992 under the supervision of Dr.R.C.Joshi and Dr.Kuldip Singh.

The matter presented in this thesis has not been submitted by me for the award of any other degree of this or any other University.


(Pawan Kumar)

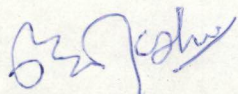
This is to certify that the above statement made by the candidate is correct to the best of our knowledge.

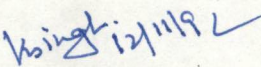

(R.C.Joshi)
Professor and Head,
Dept. of Electronics and Computer
Engineering

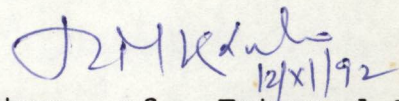

(Kuldip Singh)
Reader
Dept. of Continuing
Education


Date: 27. 4. 1992

The Ph.D Viva-Voce examination of Sri Pawan Kumar Research Scholar, has been held on _____.


Signature of


Supervisors


Signature of External Examiner


Signature of H.O.D.

ABSTRACT

This thesis addresses the techniques for the design of reliable fault-tolerant multistage interconnection networks (MINs) used in multiprocessor systems. Statically as well as dynamically reroutable MINs are studied. Methods for the construction of regular fault-tolerant MINs are described. Their characteristics pertaining to performance and reliability are analyzed and compared with the previously proposed networks. It is shown that these multipath regular MINs are of higher reliability than other MINs having similar fault-tolerant capabilities and give better performance. The effect of component failures on the performance of these networks is also evaluated. It is observed that although faults do not significantly affect the overall network performance, they degrade the performance of some parts of the system resulting in an increase in network cycle time. New type of irregular fault-tolerant multistage networks are introduced and analyzed. Various algorithms are developed to study their characteristics. Compared to regular networks, these irregular networks have lesser hardware complexity and more computational speed because of their shorter path lengths between a processor and its favourite memory modules. The results of analysis show that it is worthwhile to employ the proposed irregular techniques in designing fault-tolerant MINs. Modular implementation has been proposed to simplify the design of statically reroutable networks. The proposed fault-tolerant networks have many attractive features for use in multiprocessor systems.

ACKNOWLEDGEMENTS

I wish to express my heartfelt gratitude for the inspiration, encouragement and the expert guidance that Dr.R.C.Joshi and Dr.Kuldip Singh have given me throughout my stay at the University of Roorkee, Roorkee. I consider myself very fortunate for having been associated with them.

The co-operation and help extended by the Head, Department of Electronics and Computer Engineering, is gratefully acknowledged.

I am grateful to Dr. Arun Kumar, Reader E and CE Department for his help and friendly advice during my stay at Roorkee. I am also grateful to all my fellow researchers and friends, who have provided me encouragement and timely support.

I am thankful to my wife Neeta and children Neha and Rohit, not only for their patience during my research work, but also for the inspiration they have given me.

Last but not the least, I am thankful to my parent Institute i.e Thapar Institute of Engineering and Technology, Patiala, for sponsoring me under Quality Improvement Programme. In particular, I wish to record my gratitude to Prof. A.K. Sawhney for his co-operation during the period.

CONTENTS

	Page
ABSTRACT	(i)
ACKNOWLEDGEMENTS	(ii)
CONTENTS	(iii)
1. INTRODUCTION	1
1.1 Introduction	1
1.2 Statement of the Problem	3
1.3 Organization of the Thesis	5
2. REVIEW OF FAULT-TOLERANCE TECHNIQUES	6
2.1 Introduction	6
2.2 Interconnection Methods	6
2.3 Multistage Interconnection Networks	8
2.4 Fault-Tolerant MINs	14
2.5 Fault-Tolerance Techniques	15
2.5.1 Redundancy at the Network Level	16
2.5.2 Redundant Stages in the Network	20
2.5.3 Chaining Switches within a Stage of the Network	24
2.5.4 Other Fault-Tolerant designs for Full Access	26
2.5.5 Time Redundancy for Dynamic Full Access	28
2.6 Conclusion	31

3.	MODULAR NETWORK	32
3.1	Introduction	32
3.2	Construction	32
3.3	Routing	33
3.4	Fault-Tolerance and Repair	37
3.5	Reliability Analysis	38
3.6	Comparison with other Networks	41
3.6.1	Baseline Network	41
3.6.2	Extra Stage Cube (ESC) Network	42
3.6.3	3-Replicated Network	43
3.6.4	INDRA Network	43
3.7	Cost Analysis	44
3.8	Reliability/Cost Ratio	46
3.9	Performance Model	46
3.9.1	Assumptions	49
3.9.2	Performance Analysis	50
3.10	Conclusion	58
4.	AUGMENTED BASELINE NETWORK	60
4.1	Introduction	60
4.2	Construction	60
4.3	Routing Scheme	63
4.4	Fault-Tolerance	67
4.5	Reliability Analysis	70
4.5.1	Augmented Baseline Network (ABN)	70
4.5.2	Unique-Path Baseline Network	73

4.6	Cost-Effectiveness	74
4.7	Performance Analysis	76
4.8	Conclusion	86
5.	MODIFIED FAULT-TOLERANT DOUBLE TREE (MFDOT) NETWORK	88
5.1	Introduction	88
5.2	Modified Double Tree (MDOT) Network	88
5.3	Fault-Tolerant Double Tree (FDOT) Network	91
5.4	Modified Fault-Tolerant Double Tree (MFDOT) Network	92
5.5	Routing Scheme of MFDOT Network	95
5.5.1	Path Length Algorithm	95
5.5.2	Routing Tag Algorithm	96
5.5.3	Routing Procedure	98
5.6	Fault-Tolerance of MFDOT Network	99
5.7	Permutation Capability of MFDOT Network	100
5.8	Reliability and Repairability of MFDOT Network	100
5.9	Communication Delay in MFDOT Network	101
5.10	Conclusion	101
6.	QUAD TREE (QT) NETWORK	103
6.1	Introduction	103
6.2	Construction	103
6.3	Routing Scheme	105
6.3.1	Path Length Algorithm	105
6.3.2	Routing Tag Algorithm	107
6.3.3	Routing Procedure	110

6.4	Fault-Tolerance	112
6.5	Cost Analysis and Comparison with other Networks	114
6.6	Conclusion	116
7.	CONCLUSIONS	117
7.1	Conclusions	117
7.2	Suggestion for Future Investigations	120
	REFERENCES	122
	APPENDIX	131
	RESEARCH PAPERS OUT OF THE WORK	134

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

Quest for higher and higher computational speeds has driven the hardware technology almost to the physical limits of response time of electronic circuits, but the demand for more processing power continues to grow. As a result the computer scientists are now exploring multiprocessor architectures where several processors can work concurrently in a co-operative manner. In such a multiprocessing system, processors may spend a considerable amount of time just communicating among themselves unless an efficient interconnection network (IN) connects them. A suitable IN, thus, inevitably becomes a critical system component since a communication vehicle is needed to synchronize processes and to co-ordinate processor resources.

Multistage interconnection networks (MINs) are recognized as cost-effective means to provide high-bandwidth communication in multiprocessor systems, as opposed to crossbar switches which are highly efficient but are prohibitively expensive, with $O(N^2)$ cost (where N is the number of input and output terminals) and the time shared buses which are inexpensive but have unacceptable throughput when the number of processors connected to them is large.

A MIN consists of multiple stages of switching elements (SEs) and is categorised as regular or irregular, based upon the

number of SEs in each stage of the network. If the number of SEs are same in each stage then the network is said to be regular otherwise irregular. A majority of the proposed regular MINs belong to a class of networks which in their basic form, consist of $\log_m N$ stages of $m \times m$ SEs connecting N input terminals to N output terminals. The SEs in adjacent stages are connected by an interconnection pattern that allows any input to be connected to any output. Such MINs have $O(N \log N)$ hardware cost, $O(\log N)$ path length, and the ability to provide up to N simultaneous connections. In addition, they can employ simple and distributed routing algorithms which eliminate the need for a central controller and reduce the connection set up time. These properties make MINs attractive for multiprocessor applications.

The irregular double tree (DOT) network [58,63,81] has many desirable properties for use in multiprocessor systems, but lacks fault-tolerant capability. The useful properties of DOT network are: full access capability and shorter path lengths for a connection between a processor and its favourite memory modules as compared to regular multistage networks.

An important characteristic of these MINs is that there is a unique-path between any input-output pair. This leads to two serious disadvantages: a) poor performance and b) lack of fault-tolerance and reliability. Performance of unique-path is low because of possible blocking of an input-output connection by a previously established connection and the inability to find an alternate path. There is no fault-tolerance because even a single link or switch failure can disconnect paths between several

input-output pairs. This lack of fault-tolerant capability has received considerable attention, and many ways of providing fault-tolerance to the regular multistage interconnection networks have been proposed. An exhaustive survey has been presented by Adams et al [2].

The basic idea for fault-tolerance is to provide multiple paths for an input-output pair so that alternate paths can be used in case of faults. The methods include increasing the number of stages [1], using multiple links between stages [20,61,69], increasing the size of switches [65], partitioning a unique-path network into several subnetworks [43], and incorporating multiple copies of a basic network [51,74]. For example, an Extra Stage Cube (ESC) network [1] has a redundant stage to provide two paths for every input-output pair, while a d-replicated network [51] (or layered banyan network [34]) provides redundant paths by replicating d identical basic networks. Compared to unique-path networks, these multiple path networks certainly have higher reliability but with increased hardware complexity, which not only increases cost, but also puts wrinkle on the claim of enhanced reliability.

1.2 STATEMENT OF THE PROBLEM

The general goals for the design of fault-tolerant MINs are high reliability, good performance even in the presence of faults, high computational speed, low cost and simple control. However, most fault-tolerant MINs proposed in the literature cannot achieve all of these goals at the same time. Most of the

regular MINs proposed in the literature have path length $O(\log N)$, which puts a limitation on the computational speed of a MIN. Some of the networks fail to tolerate faults in the first and or the last stages. Some others can tolerate faults in any stage, but they cannot maintain permutation capability when any single fault occurs. A few exceptions such as replicated networks and INDRA [75] network can maintain permutation capability in case of faults, but they are, in general, too costly.

The main objective of the present research work is to explore the techniques for the design of reliable, computationally faster, fault-tolerant and cost-effective MIN having good performance. The problem of improving the reliability and performance of the existing regular fault-tolerant MINs have been examined. This thesis also investigates the possibility of constructing irregular fault-tolerant MINs having shorter path length compared to regular fault-tolerant MINs. Specifically, the problems considered in this thesis can be stated as follows:

- . To propose newer and or augmenting the existing regular networks, in order to improve their reliability and performance.
- . To develop irregular fault-tolerant MINs which are cost-effective and have shorter path length compared to regular fault-tolerant MINs.

The efforts in exploring the above objectives result in the development of the following networks:

- . Modular Network (MN)
- . Augmented Baseline Network (ABN)
- . Modified Fault-Tolerant Double Tree (MFDOT) Network, and
- . Quad Tree (QT) Network.

1.3 ORGANIZATION OF THE THESIS

The background and the related research reported by others, are outlined in chapter 2, which is helpful for the design of reliable MINs. Chapter 3 contains the performance analysis of the modular network (MN). The improvement obtained over the performance of unique-path networks is quantified and comparison of the analyses of MTTF and cost-effectiveness with other multipath networks is given. Performance under faulty conditions is also described. Chapter 4 deals with the reliability and performance analyses of augmented baseline network (ABN). The effect of component failures on the performance of ABN are also studied.

Statically re-routable irregular modified fault-tolerant double tree (MFDOT) network is described in chapter 5. The path length and routing algorithms are proposed and the fault-tolerance properties are examined. Chapter 6 covers the dynamically reroutable irregular quad tree (QT) network. The routing algorithms are formulated and a comparison with the hardware cost of other fault-tolerant MINs is presented. Finally, conclusions and proposals for further research needed in this area are given in chapter 7.

CHAPTER 2

REVIEW

2.1 INTRODUCTION

Fault-tolerance is an important factor in the study of multistage interconnection networks used in large multiprocessor systems for their continuous operation over a relatively long period of time. In such networks, the failure of a switching element or connecting link destroys the communication capability between one or more pair(s) of source and destination terminals. Many techniques exist for designing multistage interconnection networks that tolerate switch and/or link failures without losing connectivity. In this chapter, interconnection methods, multistage interconnection networks and the fault-tolerance techniques are reviewed.

2.2 INTERCONNECTION METHODS

The interconnection of N processors to N memory modules in multiprocessor systems is an important and challenging problem for large values of N . A crossbar network provides a fast means of interconnecting the resources in a multiprocessor system. Unfortunately, the complexity and the cost of such networks grows with the square of the network size N .

Several network topologies have been designed which require much less hardware than crossbars and they can be broadly classified as static and dynamic [5]. In a static scheme, each

processor is connected by dedicated links to a subset of processors. Several static structures with regular topology have been proposed - these include ring [60], tree [35], near neighbor mesh [8], hypercubes [79], systolic arrays [42], and pyramids [95]. A comparative study by Wittie [108] describes these static networks. In these networks, each processor can communicate only with a small number of processors directly and communication with others involves the transfer of information through one or more intermediate processors.

Dynamic networks allow different connections to be set up between processing elements by changing their internal states. Multistage interconnection networks, which consist of multiple stages of switching elements, are networks with a dynamic topology. Several MINs with N inputs and N outputs have been reported in the literature [31,90]. These networks have hardware complexity $O(N \log N)$ compared to $O(N^2)$ required by crossbars.

At the other end of the spectrum, the linear bus offers a simple and inexpensive means of communication, but they do not admit simultaneous communication between distinct processor/processor or processor/memory pairs. Thus, they are inefficient and becomes unacceptably slow for even a moderate number of processors. Use of multiple buses is helpful, but to achieve significant improvement in performance, a large number of buses ($\approx N^2/2$) must be employed [22,64] which makes them no longer economically attractive.

2.3 MULTISTAGE INTERCONNECTION NETWORKS

A multistage interconnection network consists of multiple stages of switching elements. Popular among them is a class of regular networks which, in their basic form, consist of $\log_m N$ stages of $m \times m$ SEs connecting N input terminals to N output terminals. If the number of switches in each stage of the network are equal then the network is said to be regular otherwise the network is irregular. They can also be built using larger SEs and correspondingly have less number of stages, with similar properties. There exist many different topologies for MINs which are characterized by the pattern of the connecting links between stages. The omega network shown in Figure 2.1, maintains a uniform connection pattern between stages, known as perfect shuffle [54,93]; many other multistage networks have non-uniform connection patterns between stages. The minimum requirement of any of these networks is to provide full access capability, which means that any input terminal of the network should be able to access any output terminal in one pass through the network.

Multistage networks differ in the interconnection pattern between stages, the type and operation of individual SEs, and the control scheme for setting up the SEs. Examples include the baseline [109], omega [54], banyan [34], the indirect binary n -cube [71], flip [10], and delta [70] networks. The topological equivalence of several of these networks has been established [4,67,109]. In multistage networks, data must flow through several switching stages. Hence, these networks have a longer internal delay as compared to a crossbar.

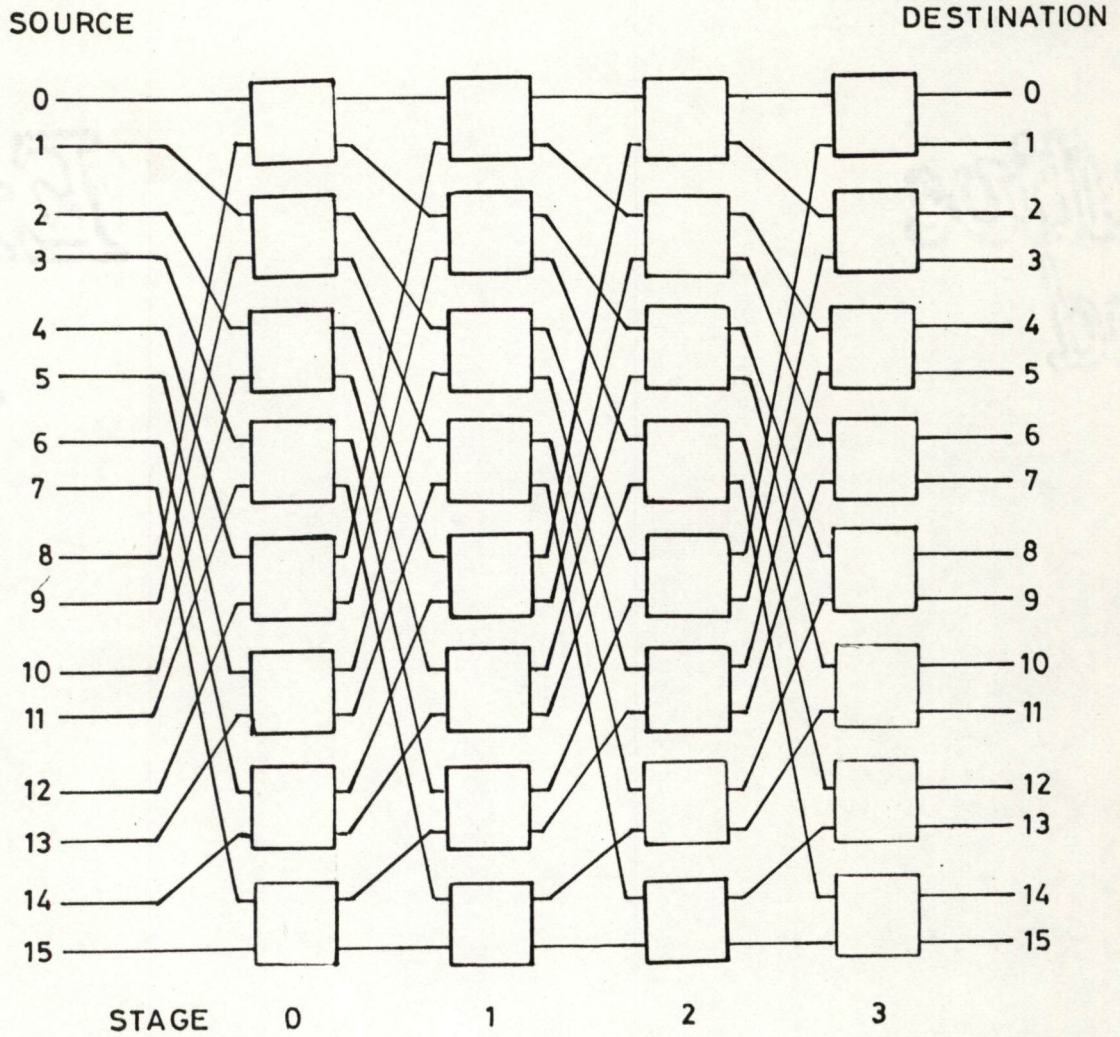


FIG. 2.1 Omega network with $N = 16$

An important characteristic of these MINs is the unique-path property. This means that each source has exactly one path through the network to reach any particular destination. Moreover, the routing of data from source to destination can be performed in a distributed manner using the destination address as routing tag. For example, Figure 2.2 represents a baseline network in which the route taken by a source with binary address $s_{n-1} s_{n-2} \dots s_0$ to destination $d_{n-1} d_{n-2} \dots d_0$ will be uniquely specified by the path code as $s_{n-1} s_{n-2} \dots s_0 d_{n-1} d_{n-2} \dots d_0$. The position of the path in any intermediate stage of the network can be found by observing a n bit long window in the path code (where $n = \log_2 N$). Conflicts can arise while routing multiple connections (or permutations) through such networks when two inputs request the same output link of a switch in some intermediate stage.

This class of networks does not realize all the possible permutations of data between inputs and outputs. Networks capable of realizing all the $N!$ permutations between the N inputs and outputs are known as rearrangeable networks. The baseline network does not possess this property since many permutations cause conflicts in one or more of the switching stages. The number of passable permutations can be increased by increasing the number of stages. This also includes multiple paths between source and destination terminals. However, the control algorithm to find non-conflicting paths for arbitrary permutations in such networks can be quite complex. A well studied multistage rearrangeable network is the benes network [11] which consists of $2 \log_2 N - 1$

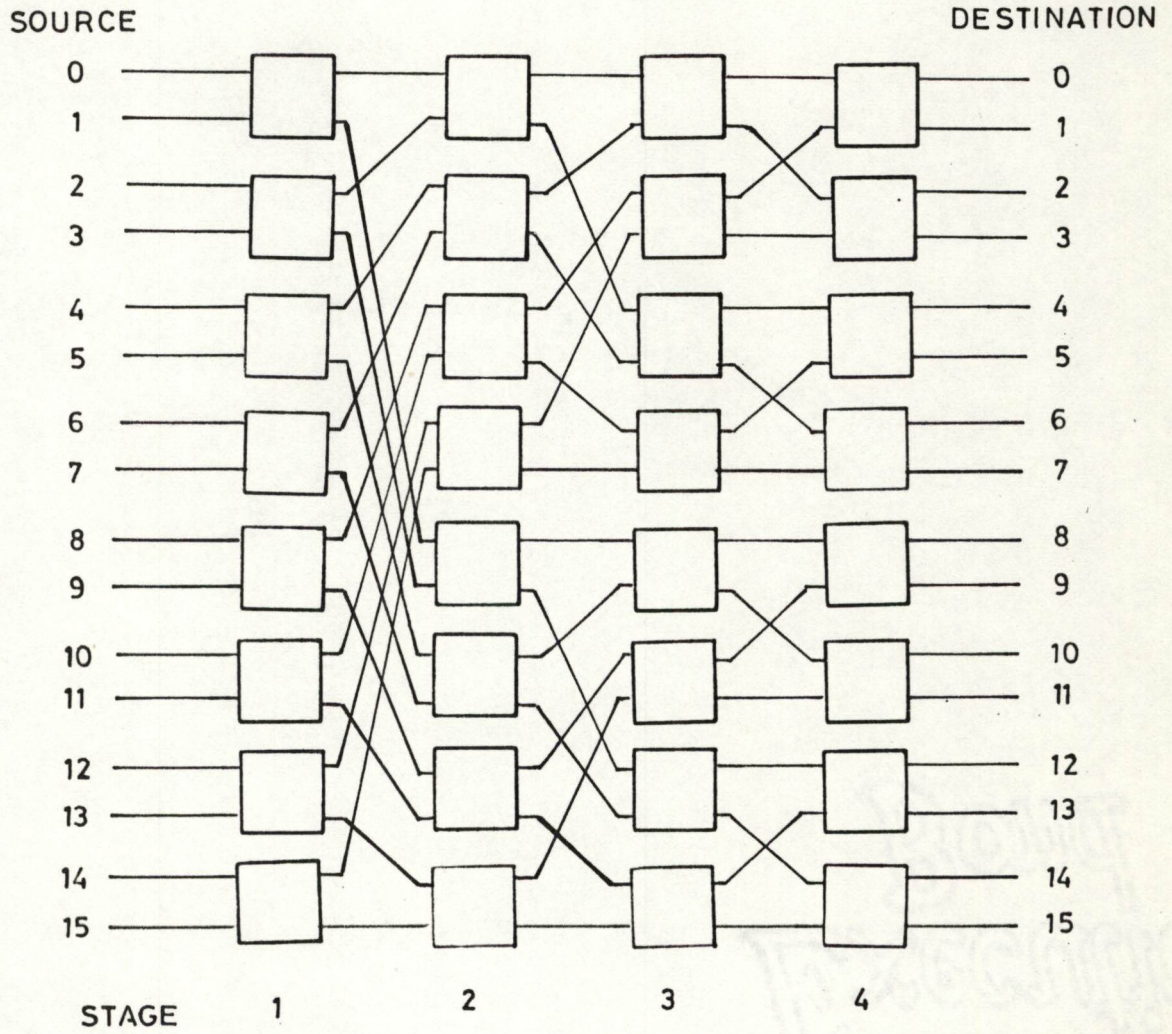


FIG. 2.2 Baseline network with $N = 16$

stages of 2×2 SEs. A serial cascade of the omega network with its inverse network also displays this property.

The irregular double tree (DOT) multistage network was originally proposed by Levitt et al [58]. It has found an application in the design of MIT data flow processor [57]. A DOT network shown in Figure 2.3 consists of a right and a left half. Each half of the network resembles a binary tree. The left and right trees are mirror images of each other. A DOT network of size $N \times N$ has N source and N destination terminals and $2n-1$ number of stages (where $n = \log_2 N$). Further, it has $(2^{n+1}-3)$ SEs. An i th and $(2n-i)$ th stage has 2^{n-i} SEs of size 2×2 for $i = 1, 2, 3, \dots, n$. DOT network possesses the property of full access. Compared to regular networks, the DOT network has shorter path length for a connection between a processor and its favourite memory modules. Thus, DOT network has many attractive properties for use in multiprocessor systems, but lacks in fault-tolerant capability.

Only a limited set of permutations are required in some computational environments, and multistage unique-path networks are usually adequate in such cases. A control algorithm is required to set up the SEs for performing various permutations. It is necessary to have small set up times in an environment where the switching permutations change rapidly. While the control of a crossbar is relatively straightforward, the algorithm for finding the switch settings in some multistage networks is more complex.

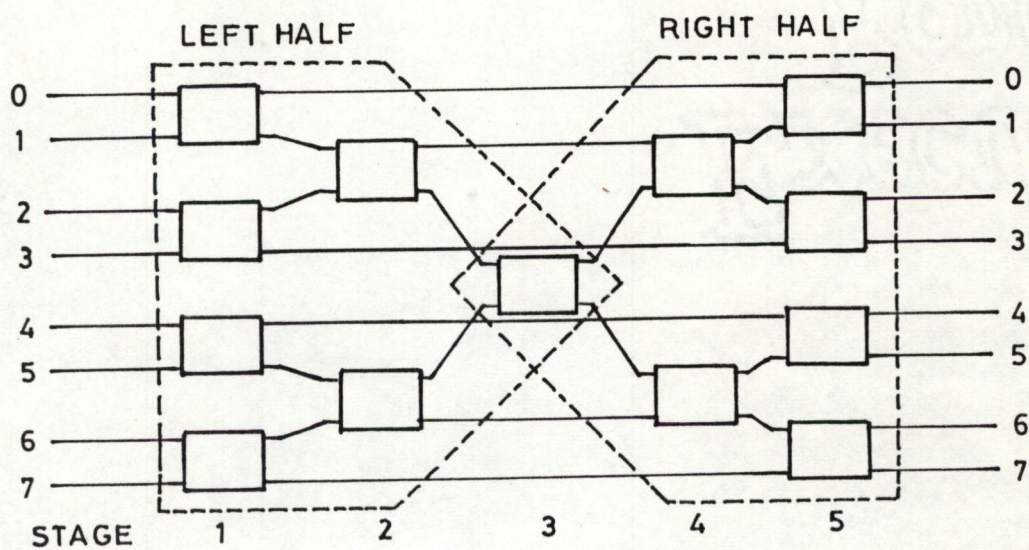


FIG. 2.3 DOT NETWORK WITH $N=8$

2.4 FAULT-TOLERANT MINS

A number of multistage interconnection networks have been proposed in the literature and majority of them belong to a class of networks, which in their basic form, consist of $\log N$ stages connecting N inputs to N outputs. These networks possess the full access property; in addition, a unique-path exists from any input terminal of the network to any output terminal. The unique-path property facilitates the use of simple and efficient routing algorithms for setting up connections through the network. However, unique-path MINS have the following two major problems:

- i) Lack of fault-tolerance and poor reliability. Since there is a unique-path between any input-output pair, the presence of a single failure among the SEs or the connecting links destroys the full access property. Even though the failure of individual components in the network are usually very small, the failure rate of the entire network can be quite high in systems of large size.
- ii) Poor performance in random access environment. Since there are no alternate paths between input-output pairs, blocked requests can cause significant deterioration in performance.

Thus, fault-tolerance is a necessary attribute for the continued operation of such systems. The basic idea for fault-tolerance is to provide multiple paths for a input-output pair so that alternate paths can be used in case of faults. Availability of multiple paths results in increased reliability and performance, because an alternate path can be chosen in the event

of a request being blocked or a faulty link or switch being encountered.

Rerouting or selection of alternate path in a MIN may be either static or dynamic. If a multipath MIN allows rerouting to be made only at the source or at some fixed points in the network, where the alternate path fork exists, then it is statically reroutable. In such MINs, if blocking occurs or faults are encountered, it may be necessary to backtrack to the stage where a fork occurs and attempt an alternate path from there. In dynamically reroutable multipath MIN, the paths between any given source-destination pair have a fork available at every stage. Thus, rerouting decisions can be made by the switches at any stage, on the fly, as faulty switches or links are encountered or as blocking occurs.

A convenient way of determining the degree of fault-tolerance of multipath MIN, and to find out whether it allows static or dynamic reroutability, is by the use of its redundancy graph. The redundancy graph in a MIN shows all the available paths between a input-output pair [65,68].

2.5 FAULT-TOLERANCE TECHNIQUES

Many ways of providing fault-tolerance to the multistage interconnection networks have been proposed. A survey is found in [2]. Most of the proposed techniques fall somewhere between 0 and 100 percent redundancy. These techniques are briefly reviewed in this section.

2.5.1 Redundancy at the Network Level

An approach to design fault-tolerant networks is to use multiple copies of the basic network such as the omega network. Examples include INDRA network [75], Merged Delta Network (MDN) [76], Augmented C-Network (ACN) [76], and d-replicated network [51]. INDRA network can be viewed as the union of R parallel layers (or copies) of a basic network with an initial distribution stage. INDRA network is constructed using $R \times R$ switches for $N = R^n$ inputs and outputs, and provides R^2 different paths between any input-output pair. The network can be visualized as a shuffle exchange MIN of size $N \times R$, with each of the N sources connected to a specified set of R switches in the first stage, and each destination receiving connections from R specified switches in the last stage. The network is shown in Figure 2.4. Up to $(R-1)$ failures can be tolerated and only static rerouting is possible.

The Merged Delta Network (MDN) [76] is constructed by combining d identical layers (copies) of $N/d \times N/d$ delta network to form a network with N inputs and outputs. The basic difference between an MDN and a INDRA network is that the MDN allows connections to cross layers between stages of the network, whereas a connection is confined to a single layer in the case of INDRA network. Given d copies of delta networks, each with N/d inputs and outputs and consisting of $R \times R$ SEs, an MDN is constructed as follows:

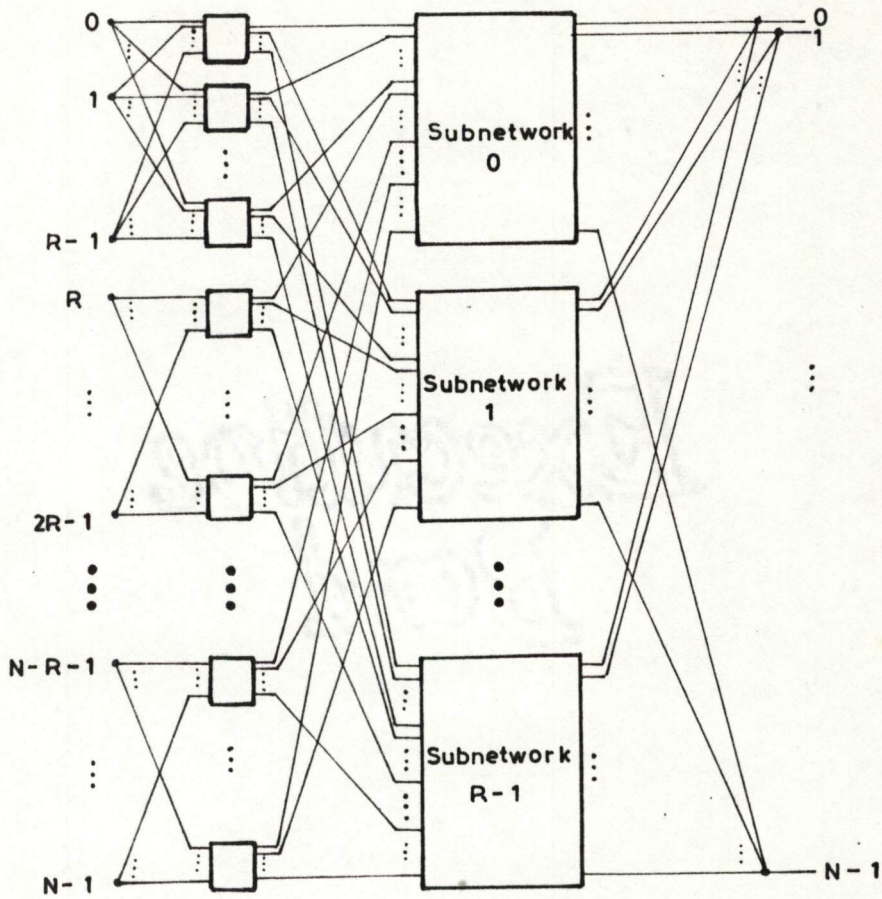


FIG. 2.4 INDRA network shown as the union of R subnetworks

1. Increase the size of all SEs to $R.d \times R.d$.
2. If an input (output) terminal i of the network was originally connected to switch j in the input (output) stage of one of the delta networks, connect it to switch j in the input (output) stage of all the delta networks.
3. If a switch j in some stage i of the delta network was originally connected to switch k of the stage $(i + 1)$, connect it to switch k in stage $(i + 1)$ of all the delta networks.

An MDN constructed from d copies of delta networks is referred to as a d -MDN. Figure 2.5 shows a 2-MDN consisting of 8 inputs and outputs constructed from two identical copies of delta networks of size 4×4 .

Another example of combining multiple layer of networks to obtain fault-tolerance is the class of Augmented C-Network (ACN). The omega network has the property that for each SE in every stage, except the last, another SE exists in the same stage such that they are connected to a common pair of switches in the next stage. Such a pair of switches is called "conjugates" [76] or "output buddies" [4]. Networks satisfying this property are called C-networks and can be used to construct Augmented C-Network (ACN) [76]. An ACN with $N = 2^n$ inputs and outputs is constructed from an omega network of the same size as follows:

1. Replace each 2×2 switch by a 4×4 switch.
2. If an input terminal in the original network is connected to a switch j in the input stage of the network, connect the input terminal to the conjugate of j also.

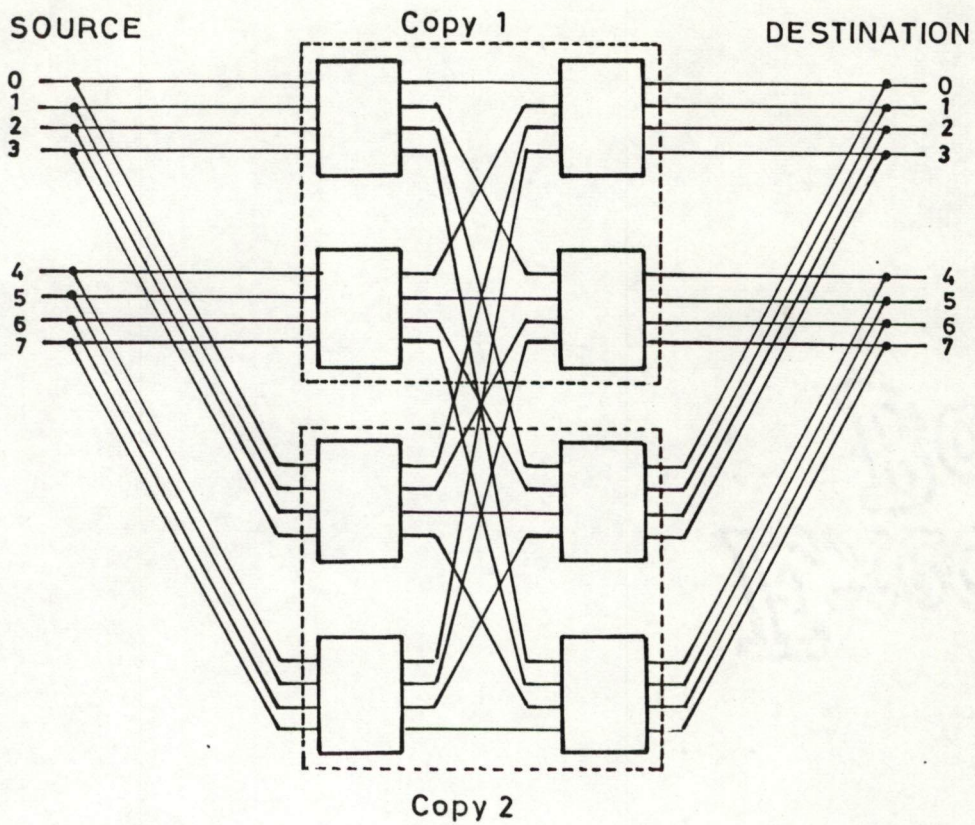


FIG. 2.5 A 2-MDN with 8 inputs and outputs

3. If an input terminal of the network was initially connected to switch j of the network, connect it also to the switch $(j + N/4) \bmod N/2$.
4. For every stage i except the last, if a switch j is connected to switches k, k' of stage $(i + 1)$, connect its two new outputs to the conjugates of k and k' in stage $(i + 1)$.

Figure 2.6 shows an ACN with $N = 8$ inputs and outputs constructed from an omega network of the same size. The added links are shown by broken lines.

2.5.2 Redundant Stages in the Network

Adding redundant copies of networks is an expensive solution, but allows routing of classes of permutations even when there are multiple switch failures. If only the full access capability is required, then extra switching stages can be used to provide multiple disjoint paths between source and destination pairs. This technique provides fault-tolerance at a modest overhead. Examples include the Extra-Stage Cube (ESC) network [1], extra-stage omega network [104] and banyan networks [18]. The Extra Stage Cube network is constructed from the generalized cube network [1] by adding an extra stage at the front end of the network. The switches in this extra stage and in the final stage are equipped with demultiplexers at the inputs and multiplexers at the outputs, respectively. This arrangement makes it possible to tolerate a single switch failure in any stage of the network. Figure 2.7 shows an ESC network. Figure 2.8 shows an extra-stage

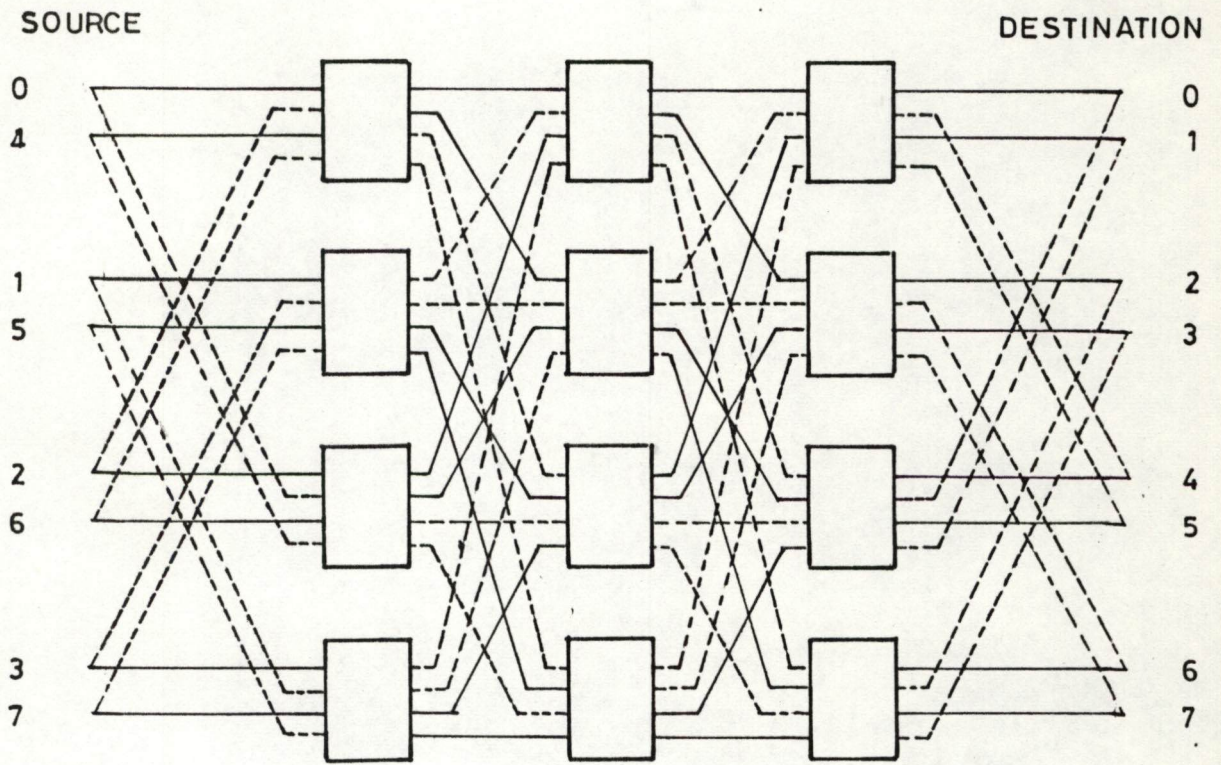


FIG. 2.6 An ACN with $N = 8$

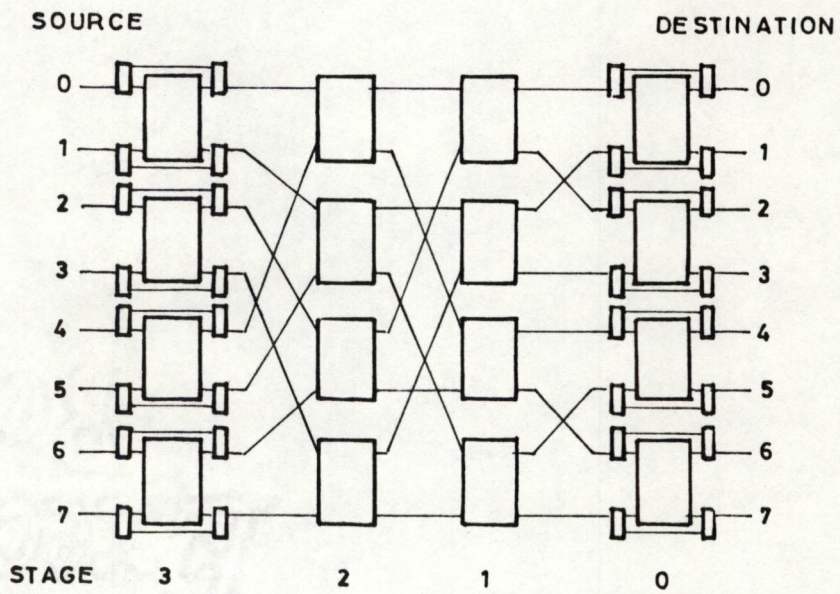


FIG.2.7 The extra - stage cube (E SC) with $N=8$

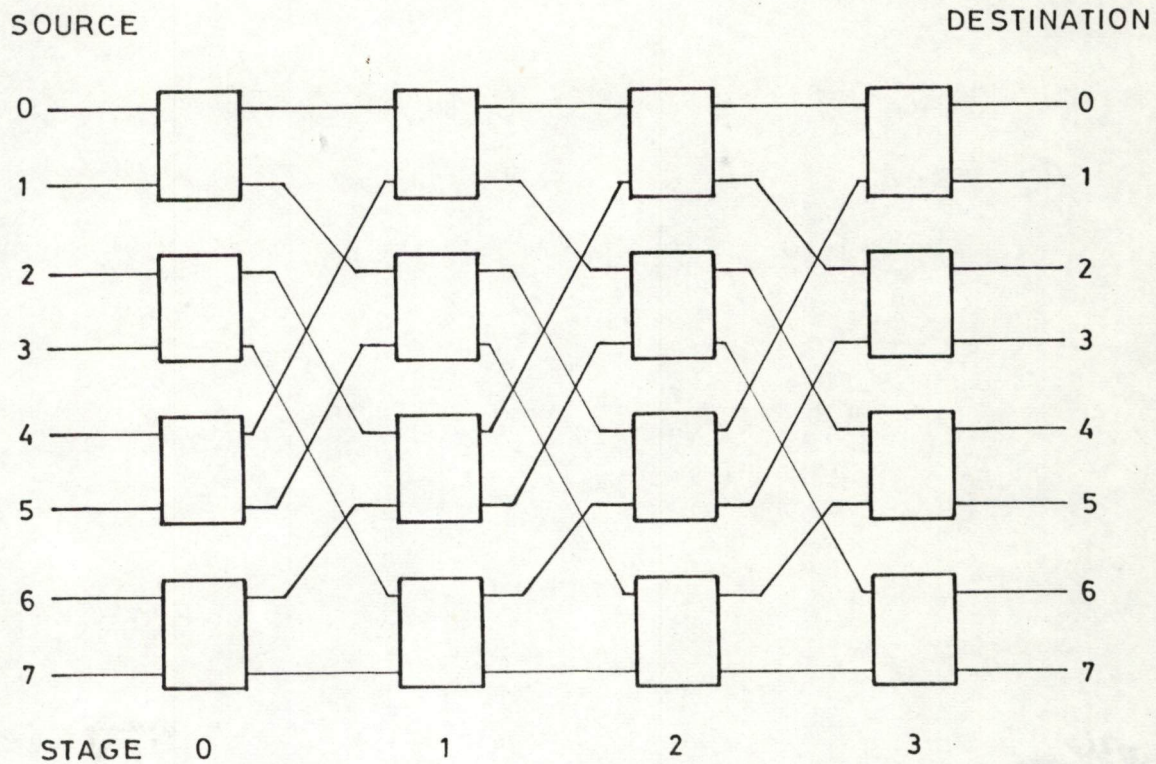


FIG. 2.8 The extra-stage omega network with $N = 8$

omega network obtained by adding a redundant switching stage at the input of the network. The extra stage allows the network to retain its full-access capability in the presence of single switch failures in any stage except the first and the last. For all single failures in the intermediate stages, the network can also route permutations by performing multiple passes, each pass realizing a submap of the permutation [104].

2.5.3 Chaining Switches within a Stage of the Network

The technique achieves fault-tolerance by providing additional links between switches belonging to the same stage, so that data can sidestep a faulty switch. Examples include Augmented Shuffle Exchange Network (ASEN) [46]. Figure 2.9 shows an ASEN-2. ASEN is a multipath MIN which features links among switches belonging to the same stage. It has the property that for every loop of switches, there exists another loop which is connected to the same set of switches in the next stage. An ASEN is self routing. Each source has a primary multiplexer and SE and a secondary multiplexer and SE. Each source attempts entry into the ASEN via its primary multiplexer and SE. If either primary component is faulty, the request is sent to the secondary multiplexer. If the secondary multiplexer is also faulty, the ASEN fails. For stages 1 through $\log_2 N - 2$, requests are first routed through the usual output link, if it is busy or if the successor SE (in the next stage) is faulty, routing is attempted via the auxiliary link. A faulty demultiplexer at the output of the ASEN is regarded as a failure of its associated SE in stage

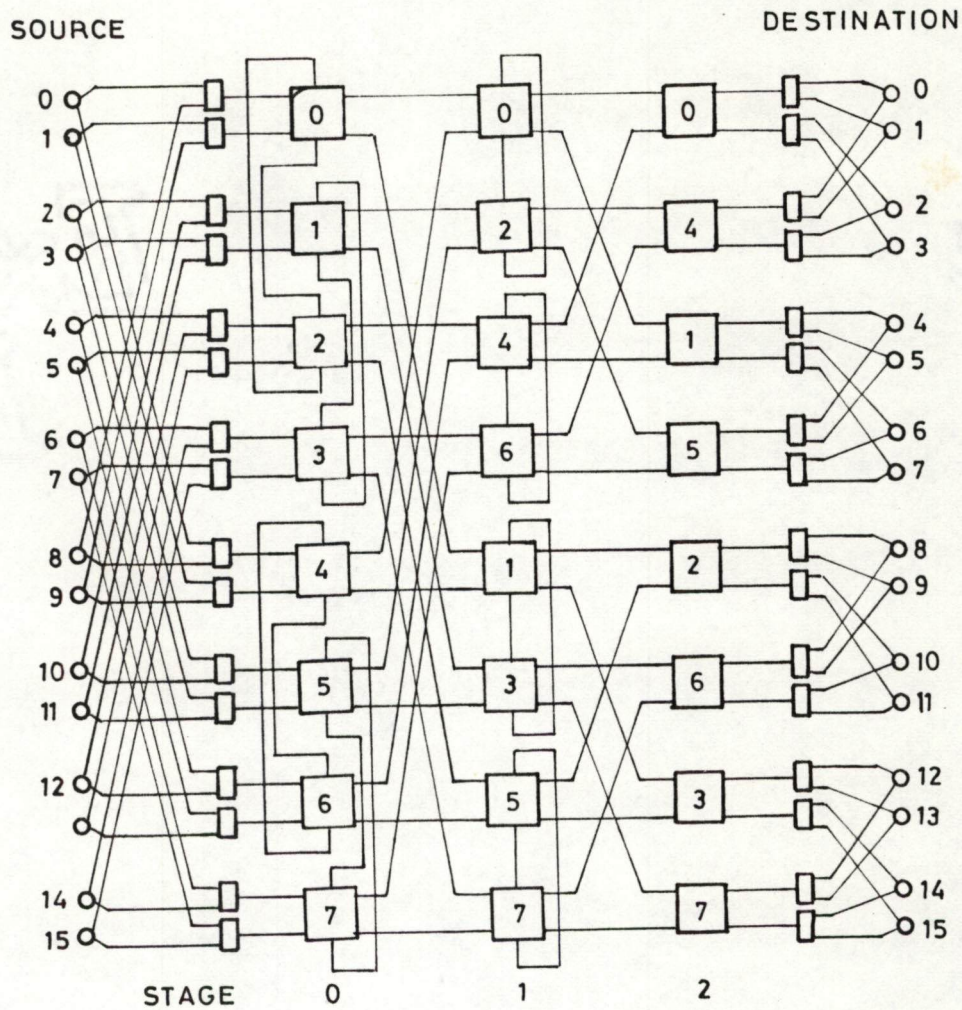


FIG. 2.9 A augmented shuffle-exchange network (ASEN-2) with $N = 16$

$\log_2 N - 1$. This strategy essentially enables a SE to detect a failure of its successor SE and re-route the request whenever possible. The ASEN is failed if a request that is not blocked does not find a path to its destination.

This technique has two benefits. First, the network can tolerate the failure of both switches in a conjugate loop. Second, the network provides a topology which lends itself to on-line repair and maintainability. That is, a loop can be removed from the ASEN without disrupting the operation of the network.

2.5.4 Other Fault-Tolerant Designs for Full Access

Multistage networks designed to provide full access in the presence of faults include the Augmented Data Manipulator (ADM) [62], Inverse Augmented Data Manipulator (IADM) [61], the Gamma network [69], the F-network [19], and the modified omega network [65,66]. These networks use enhanced SEs and/or additional links between stages to provide multiple paths between input/output pairs, thereby achieving fault-tolerance.

The ADM, Inverse ADM (IADM) as well as the Gamma network are based on $\pm 2^i$ interconnection between stages. They use SEs with three inputs and outputs, except in the input and output stages, where the switches are 1×3 and 3×1 respectively. Each of these networks has $(n + 1)$ switching stages for $N = 2^n$ inputs and outputs. At stage i of the IADM, the outputs of a switch j are connected to the switches j , $(j + 2^i) \bmod N$, and $(j - 2^i) \bmod N$, of the $(i + 1)$ th stage. Figure 2.10 shows the ADM network. The Gamma network uses the same interconnection patterns as the IADM.

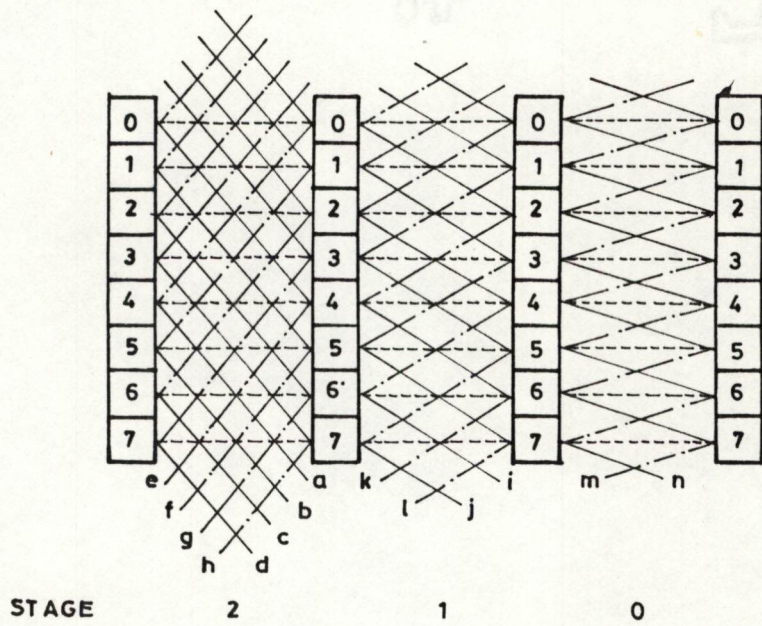


FIG. 2.10 The augmented data manipulator (ADM)

The difference between the ADM/IADM and the Gamma network is that the SEs in the former networks allow only one connection to be routed at a time, whereas the SEs in the latter are 3x3 crossbars.

Figure 2.11 shows the Gamma network for $N = 8$. The set of paths between a source-destination pair in the Gamma network can be specified using the binary fully-redundant number system. Under this system, a digit can take one of the three values - 0, 1, $\bar{1}$ ($\bar{1}$ stands for -1). An input-output connection can be routed by means of a n -digit routing tag which is the modulo- N difference between the destination and the source expressed in binary redundant number system. The value of the routing tag can be used in a straightforward manner to set up the path in the network. A switch in stage i needs to examine only the i th digit of the tag: if this digit is 0, the straight connection is used, if 1 then the downward ($+2^i$) link is used, and if $\bar{1}$ then the upward (-2^i) link is used. When the source and the destination are distinct, there are multiple representations for the routing tag; hence there are multiple paths for routing the connection in the network.

2.5.5 Time-Redundancy for Dynamic Full Access

The techniques discussed above achieve their fault-tolerance by means of redundant hardware. An alternative approach to fault-tolerance is obtained by means of redundancy in time. Thus, data may be routed in a unique-path network in the presence of faults by performing multiple passes through the network. When a fault-

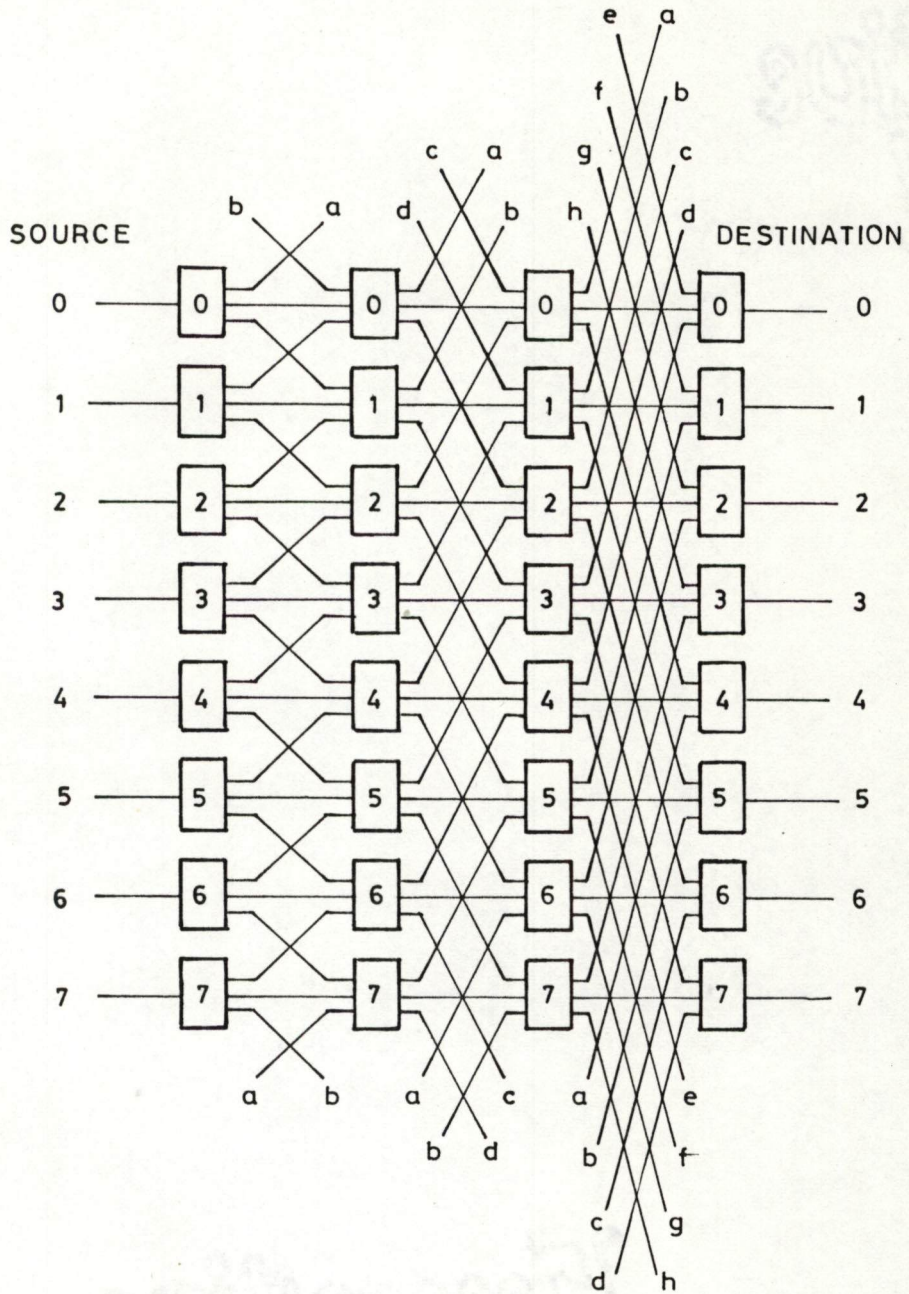


FIG. 2.11 The gamma-network for $N=8$

free path is not available, it may still be possible to route data from the input terminal to the output terminal in multiple passes by routing through intermediate destinations if the input and output terminals of the network are connected to the same set of nodes. This approach is useful when a unique-path network is used for processor-processor connection. The network is said to possess dynamic full access (DFA) capability if every processor in the system can communicate with every other processor in a finite number of passes through the network, routing the data through intermediate processing elements (PEs) if necessary [83]. Even though the failure of a single component destroys the full access capability of the omega network, a large number of faults do not destroy the DFA capability. Thus, by devising a routing procedure that allows routing through intermediate processors, connectivity of the system can be maintained.

It has been shown that the DFA capability is maintained under a large number of faults. A maximum of $(\log_2 N - 2)$ passes through the network are shown to be sufficient for the communication between any two processors in the system for a network of size $N \geq 32$ if the faults satisfy certain conditions [104]. The reconfigured system operates in a degraded mode owing to the increased latency and the additional blocking and congestion introduced by the loss of paths. However, a large waste of computational effort and resources associated with the reassignment of processors is prevented under this scheme.

2.6 CONCLUSION

In this chapter, unique-path and multipath multistage interconnection networks have been examined. Some of the commonly used techniques for providing fault-tolerance in the multistage interconnection networks were reviewed. These techniques are important in the design of fault-tolerant MINs. The next chapter describes the performance analysis of modular network (MN).

CHAPTER 3

MODULAR NETWORK

3.1 INTRODUCTION

This chapter introduces a class of fault-tolerant statically reroutable regular multistage interconnection network named as modular network (MN). Scheme for augmenting unique-path multistage interconnection networks to create redundant paths between every source-destination pair is presented. The proposed modular networks are designed to have a number of identical independent unique-path MIN modules such that a connection path between each source-destination pair can be established through any of the modules. Although an individual component failure reduces MN performance, it does not cause a total network failure. In this chapter, the analyses pertaining to, performance that incorporates the effect of faults, cost and reliability/cost are presented to provide a quantitative measurement of the capabilities of MNs.

3.2 CONSTRUCTION

Modular network MN-k of size $N \times N$ is designed by interconnecting k disjoint subnetworks, each consisting of regular unique-path MIN of size $N/k \times N/k$ where $k (\geq 2)$ and $N (> k)$ are the powers of 2. Although it is possible to have different types of regular unique-path MINs in each subnetwork but it is assumed that all the k subnetworks are of identical

type and each subnetwork consists of a baseline network [109] of size $N/k \times N/k$. The sources and destinations are connected to each of k subnetworks through $k \times 1$ multiplexers and $1 \times k$ demultiplexers. Each subnetwork consisting of a unique-path MIN along with its associated multiplexers and demultiplexers is called a module, and are denoted as M_0, M_1, \dots, M_{k-1} . Thus each module of $MN-k$ consists of N/k number of $k \times 1$ multiplexers and an equal number of $1 \times k$ demultiplexers. Further, there are n ($n = \log_2 N/k$) intermediate stages having $(n - 1)N/k$ 2×2 crossbar switches.

A network resulting from this construction is named modular networks ($MN-k$). An example of $MN-2$ of size 8×8 and $MN-4$ of size 16×16 along with their redundancy graphs are illustrated in Figures 3.1 and 3.2 respectively. The above construction procedure of MNs can be applied to the networks consisting of $r \times r$ crossbar switches (where $r \geq 2$).

3.3 ROUTING

Let $S_{i,j}$ and $D_{m,n}$ denote the source j and destination n , which are associated with modules M_i and M_m respectively based on the partition $0 \leq i, m \leq k-1$ and $0 \leq j, n \leq N/k-1$ and are represented in mixed radix as :

$$S_{i,j} = s_i, s_{(n-1)} \dots s_1 s_0.$$

$$D_{m,n} = d_m, d_{(n-1)} \dots d_1 d_0.$$

There are two ways of selecting a path between a particular source-destination pair of an $MN-k$, out of the k available paths. One way is the random selection of a path i.e a path is selected

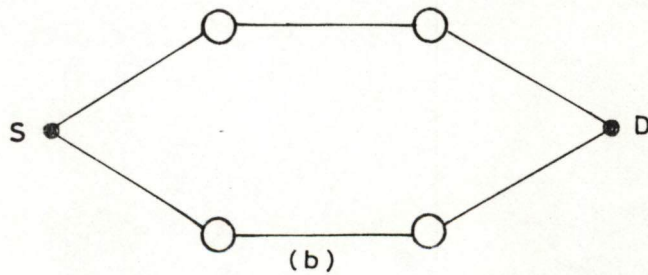
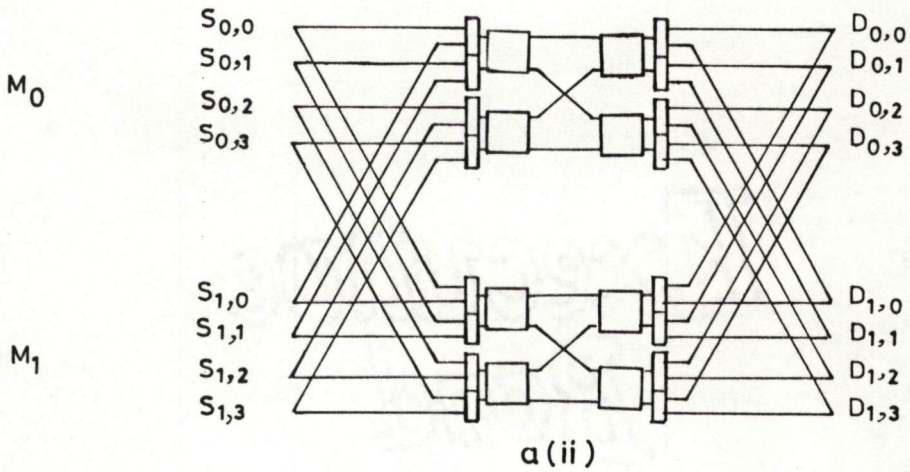
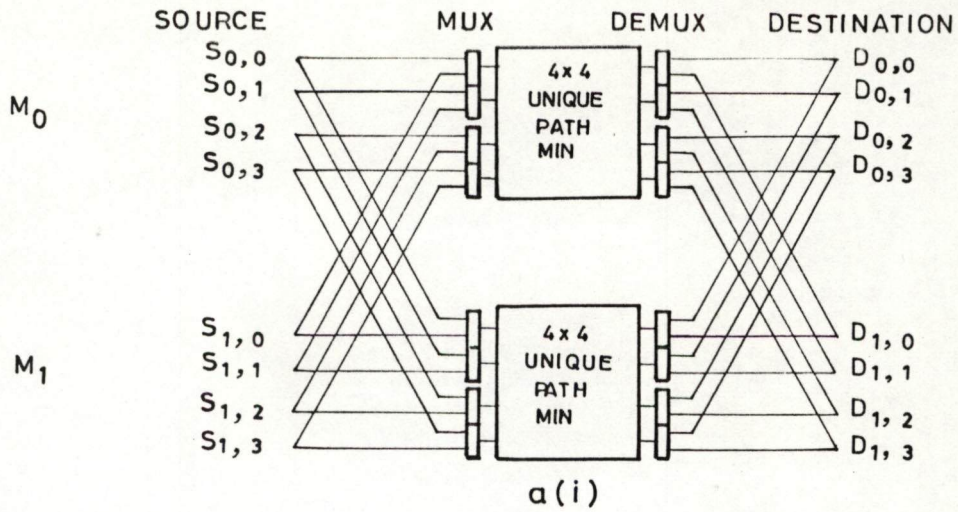
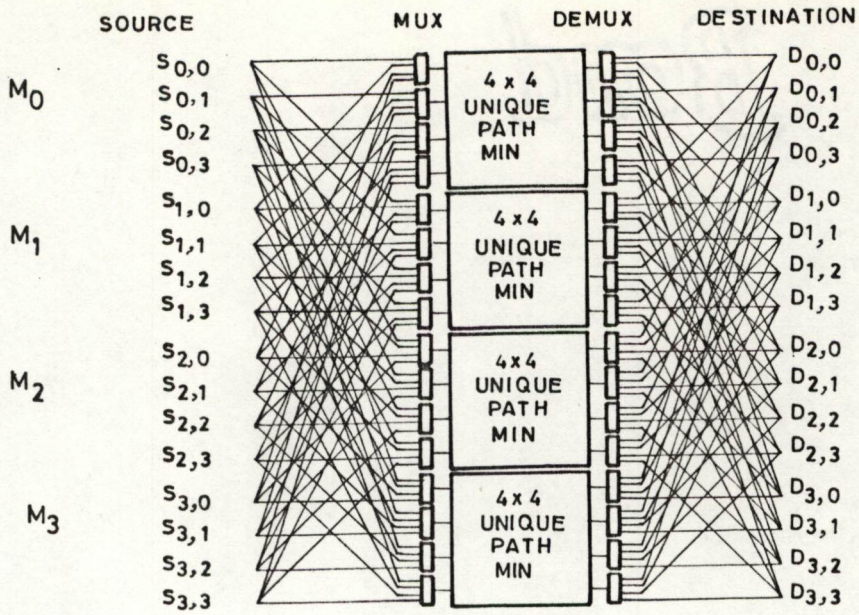
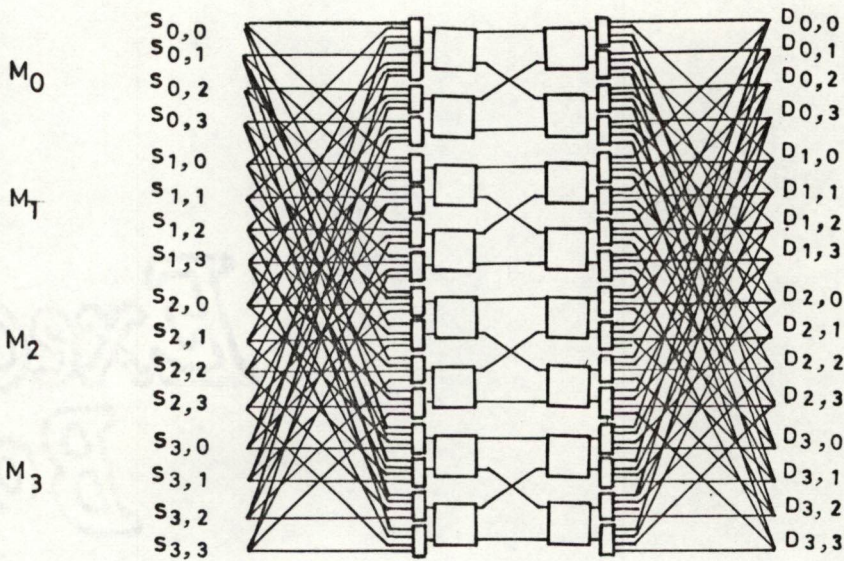


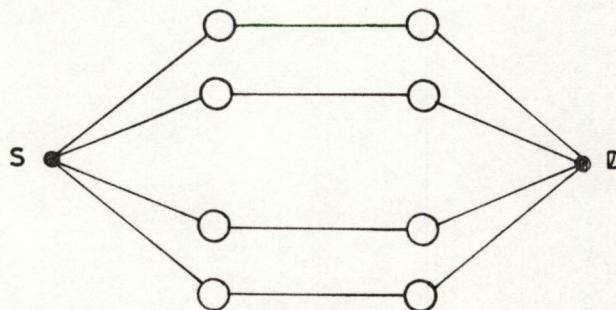
FIG. 3.1(a) An MN-2 of size 8, showing U-Path MIN as
 (i) Blocks (ii) Baseline network
 (b) Its redundancy graph



a(i)



a(ii)



(b)

FIG. 3.2 (a) An MN-4 of size 16, showing U-Path MIN as (i) blocks (ii) Baseline network (b) Redundancy graph

randomly from the available paths, each path having equal probability of selection. As one of the paths become faulty, the number of available paths decreases. In this way, one can achieve a gracefull degradation of the performance, in case of faults. The other way is the preferred path i.e the path always chosen first by the source. For example, if a source needs to communicate with a destination $D_{m,n}$, the preferred path can be through M_m module i.e the request will be sent to the corresponding multiplexer in M_m . If the preferred path is faulty (it is assumed that some techniques of fault diagnosis are available to detect and locate faults in the network [2]), the source then may try the path in one of the modules other than M_m . If that path is faulty too, the request will be submitted to an alternative path in another available module until all the paths are faulty. In this way, one can maintain a desirable performance level.

Assuming that the selection of a path out of the k available paths is already done at source, the request is submitted to the corresponding multiplexer of the selected path and the routing is as follows:

The routing tag in mixed radix representation is given as:

$$\text{Routing tag} = d_n - 1 \cdot \cdot \cdot d_1 d_0 d_m \cdot$$

Bit d_i is in radix r_i , where i th stage of every module is built by using $r_i \times r_i$ crossbar switches and d_m is in radix k , $0 \leq i < n$.

No tag bit is consumed at the multiplexer stage, because a multiplexer has only one output. After that the routing in the

intermediate stages of a module depends upon n tag bits i.e. $(d_{n-1} \dots d_1 d_0)$. The request progresses through the intermediate stages as the switch in stage i uses the tag bit d_i , the stages are numbered in sequence as $n-1, \dots, 1, 0$ from left to right, to route the incoming request via the output link with the label d_i . Finally, the request arriving at the output stage of demultiplexers, is routed to the proper destination with the help of d_m bit. Notice that the routing algorithm is independent of the selection of a particular subnetwork.

From the description of the routing procedure, it is clear that MN- k delivers a request from a source to an arbitrary destination in the presence of $(k-1)$ faults. This claim is logically stated as follows :

Theorem 1: Routing tag delivers a request generated by a source to the correct destination tolerating up to $(k-1)$ module faults.

3.4 FAULT-TOLERANCE AND REPAIR

The fault-tolerance of an MN- k can be analysed by defining a critical set of components. A critical set of SEs in MN- k is defined as the set of k SEs, each from different modules and having the same position, such that a failure will occur if all the k SEs become faulty simultaneously. Thus, there are $(N/2k \cdot \log_2 N/k + 2N/k)$ number of critical sets in MN- k . The following theorem characterizes the faults, that are always tolerated in MN- k .

Theorem 2: MN- k is $k-1$ fault-tolerant.

Proof: In the redundancy graph of MN- k , there are k paths

available between each source-destination pair. If the faults are such that they affect $k-1$ or fewer SEs in a critical set, there exists at least one fault-free path between any source-destination pair. Thus the fault-tolerance is $k-1$.

Q.E.D.

Theorem 3: MN- k possess full access capability.

Proof: Since MN- k provides k distinct paths for each source-destination pair through the k independent modules, so it is $k-1$ fault-tolerant. If there are $k-1$ or fewer faulty modules in an MN, at least one of the module is still fault free. As every module can provide a connection for any source-destination pair, the full access capability can be maintained.

The following repair scheme is proposed. If some faults develops in any module of MN- k , then the faulty module can be simply replaced by a new one. However, the overall operation of the network need not be disrupted, as the network can continue to operate at a reduced level of performance while it is being repaired. Thus the network is on-line repairable.

3.5 RELIABILITY ANALYSIS-Mean Time To Failure

A well known criterion used to measure the reliability of fault-tolerant networks is full access. Under the criterion of full access, a network is assumed to be faulty, if there is any source-destination pair that cannot be connected because of faulty components in the network. Under this criterion, the reliability of a network can be measured in terms of Mean Time to Failure (MTTF) of the network. The MTTF of a MIN, is defined as the expected time elapsed before some source is disconnected from

some destination. A fault model captures the assumed effects of physical failures on the operation of a system. Three fault models are used for MINs : the stuck-at fault model, the link-fault model, and the switch-fault model. In the stuck-at fault model, a failure causes a crossbar switch to remain in a particular state regardless of the control inputs given to it, thus affecting its ability to set up proper connections. The affected switch can be used to set up paths, if the stuck-at fault is also the required state. In the link-fault model, a failure affects an individual link of a switch, leaving the remaining part of the switch operational. In the switch-fault model, the strongest of the three, a failure makes a switch totally unusable. The switch-fault model is used for the analysis of MNs. It is assumed that any of the SEs i.e crossbar switches, multiplexers or demultiplexers in an MN can fail. In this section, the MTTF of MNs is analysed. The analysis is based upon the lower and upper bounds on the reliability of the networks. To make the analysis tractable, the following assumptions are made.

- i) Switch failures occur independently in a network with a failure rate of λ for 2x2 crossbar switches (a reasonable estimate for λ is about 10^{-6} per hour [46]).
- ii) failures of multiplexers and demultiplexers also occur independently with failure rates of λ_m and λ_d respectively, which can be different from λ .

In general, more complicated components lead to higher failure rate. Assuming that the hardware complexity of a

component is directly proportional to the gate counts of it [70,99], one can derive a failure rate of the components.

Based on the gate counts of crossbar switches the number of gates in a 2x2 crossbar switch is approximately equal to that in a 4x1 MUX or in a 1x4 DEMUX. To simplify the analysis it can be assumed that $\lambda_m = \lambda k/4$ for a kx1 MUX or $\lambda_d (= \lambda_m)$ for a 1xk DEMUX. The analysis for the upper bounds and lower bounds of MTTF are as follows:

Upper Bound

For a system with reliability function $R_S(t)$, the MTTF is given by:

$$MTTF = \int_0^{\infty} R_S(t) \cdot dt$$

$$R_S(t) = e^{-\lambda' t} \quad \text{for a SE}$$

The probability $R_{CS}(t)$, that a critical set of SEs is not faulty is:

$$R_{CS}(t) = 1 - (1 - e^{-\lambda' t})^k$$

where

$$\lambda' = \begin{cases} \lambda & \dots \text{ for a 2x2 crossbar switch,} \\ k \lambda / 4 & \dots \text{ for a kx1 multiplexer or 1xk} \\ & \text{demultiplexer.} \end{cases}$$

The probability that an MN-k is not faulty is:

$$R_{MN-k} = [1 - (1 - e^{-\lambda t})^k]^{Nn/2k} [1 - (1 - e^{-k\lambda t/4})^k]^{2N/k}$$

Thus, the upper bound of MTTF is :

$$MTTF_{MN-k} = \int_0^{\infty} R_{MN-k}(t) \cdot dt$$

Lower Bound

Here each module is considered independently and is assumed to be faulty if there is any single fault in it. Since all the modules of an MN-k cannot be simultaneously faulty if the network is to retain its connection capability. Thus, a lower bound of MTTF can be obtained based on this condition and is given as :

$$MTTF_{MN-k} = \int_0^{\infty} [1 - (1 - e^{-(Nn/2k + N/2) \lambda t})^k] \cdot dt$$

3.6 COMPARISON WITH OTHER NETWORKS

In this section, the MTTF expressions for some of the known redundant path interconnection networks are computed and the results are compared with those of the MNs. The networks used for comparison are Extra Stage Cube (ESC), 3-replicated and INDRA (with R=2). Though baseline is not a fault-tolerant network, but it is used as a yardstick to measure the reliability improvement. The MTTF expression of these networks are also obtained in the same way as is obtained for MNs and are given below:

3.6.1 Baseline Network

Since any single fault leads to a network failure under the full access criterion, so the MTTF is:

$$MTTF_{baseline} = \int_0^{\infty} [e^{-\lambda t}]^{N/2 \cdot \log N} \cdot dt$$

The lower bound and upper bound expressions are the same.

3.6.2 Extra Stage Cube (ESC) Network

Upper Bound

In Extra Stage Cube (ESC) network, a critical set consists of each 2x2 switch with two multiplexers (or two demultiplexers) in the input stage (or output stage), as it will cause some source-destination pairs disconnected if the switch and any one of its two associated multiplexers (or demultiplexers) become faulty. For these critical sets, the probability (R_{CS}) that a critical set is not faulty is :

$$R_{CS} = 1 - (1 - e^{-\lambda t}) * [1 - (e^{-\lambda t}/2)^2] = 1 - (1 - e^{-\lambda t})^2$$

There are two disjoint paths for each source-destination pair, in the stages except the input and output stages because of the extra stage provided. It is assumed that a critical set for the stages other than input and output consists of two switches of the same stage and consider each stage independently, for an upper bound of MTTF. Thus, the MTTF is:

$$MTTF_{ESC} = \int_0^{\infty} [1 - (1 - e^{-\lambda t})^2]^{N/4(\log_2 N - 1) + N} dt$$

Lower Bound

Here, the stages other than the input and output are modeled as two subnetworks, because it provides two independent paths for each source-destination pair. However, the input and output stages still need to satisfy the same condition as for the upper

bound; the failure of any critical set of switch and two multiplexers (or demultiplexers) leads to the network failure. Thus, the lower bound of MTTF can be obtained by considering the case when at least one of the subnetworks is fault free and is given as:

$$MTTF_{ESC} = \int_0^{\infty} [1 - (1 - e^{-\lambda N/4(\log_2 N - 1)t})^2] [1 - (1 - e^{-\lambda t})^2]^N dt$$

3.6.3 3-Replicated Network

Upper Bound

In this network the critical set includes three switches on the same stage of each subnetwork. Thus, the MTTF is:

$$MTTF_{3-rep} = \int_0^{\infty} [1 - (1 - e^{-\lambda t})^3]^{N/2 \log_2 N} dt$$

Lower Bound

Here the sufficient condition for the network to be operative is that at least one of the subnetworks is fault free. Thus, the MTTF is:

$$MTTF_{3-rep} = \int_0^{\infty} [1 - (1 - e^{-\lambda N/2 \log_2 N \cdot t})^3] dt$$

3.6.4 INDRA Network

Upper Bound

In this network a critical set contains two switches either on the same stage of each subnetwork or on the distribution stage

of a particular subnetwork. So the MTTF is :

$$MTTF_{INDRA} = \int_0^{\infty} [1 - (1 - e^{-\lambda t})^2]^{N/2 (\log_2 N + 1)} dt$$

Lower Bound

Since an INDRA network incorporates multiple copies of a basic network but prefaces the copies with a distribution stage of switches, the sufficient condition is that at least one of the switches in every switch pair in the distribution stage is not faulty, and at least one of the basic networks is fault free. Thus, the MTTF is:

$$MTTF_{INDRA} = \int_0^{\infty} [1 - (1 - e^{-N/2 \log_2 N \lambda t})^2] [1 - (1 - e^{-\lambda t})^2]^{N/2} dt$$

The ratios of the upper and lower bounds of MTTF for these fault tolerant networks to that of baseline network are shown in tables 3.1 and 3.2 respectively. From these tables it is clear that in most cases MNs perform more reliably than other networks.

3.7 COST ANALYSIS

In this section, the hardware cost and cost-effectiveness of MNs are computed and compared with some of the known redundant path interconnection networks. The networks used for comparison are ESC [1], 3-replicated [51], INDRA [75]. Though baseline [109] is not a fault-tolerant network, but it is used to measure the improvement in cost-effectiveness. To estimate the cost of a network, one common method is to calculate the switch complexity with an assumption that the cost of a switch is proportional to the number of gates involved, which is roughly proportional to

Table 3.1

Network Size	MN-4	MN-2	ESC	3-Rep	INDRA
16	1.3432	0.9026	0.7749	1.0336	0.6900
32	1.6138	1.0838	0.9779	1.2761	0.8843
64	1.8787	1.2624	1.1720	1.5142	1.0720
128	2.1397	1.4388	1.3590	1.7488	1.2550
256	2.3977	1.6133	1.5433	1.9805	1.4343
512	2.6532	1.7861	1.7231	2.2097	1.6107
1024	2.9066	1.9573	1.9000	2.4368	1.7847

Table 3.1: Ratio of upper bounds of MTTF for the fault-tolerant networks to that of baseline network.

Table 3.2

Network Size	MN-4	MN-2	ESC	3-Rep	INDRA
16	5.5556	2.4000	3.3641	1.8333	1.4742
32	5.9524	2.5000	3.5177	1.8333	1.4911
64	6.2500	2.5714	3.5179	1.8333	1.4968
128	6.4815	2.6250	3.4707	1.8333	1.4988
256	6.6667	2.6667	3.4178	1.8333	1.4995
512	6.8182	2.7000	3.3709	1.8333	1.4998
1024	6.9444	2.7273	3.3317	1.8333	1.4999

Table 3.2: Ratio of lower bounds of MTTF for the fault-tolerant networks to that of baseline network.

the number of 'crosspoints' within a switch [70,99]. For example, a 4x4 switch has 16 units of hardware cost, whereas 2x2 switch has 4 units. For the multiplexers and demultiplexers, it is roughly assumed that each of $k \times 1$ multiplexers or $1 \times k$ demultiplexers has k units of cost. So an MN- k , ESC, 3-replicated, INDRA, and baseline have the costs of $2N(n + k)$, $2N(\log_2 N + 5)$, $6N \log_2 N$, $4N(\log_2 N + 1)$ and $2N \log_2 N$ respectively. Graph shown in Figure 3.3 illustrates the variation of cost function with the size of the networks (2x2 switches). From the graph it is clear that MNs have lesser hardware cost than other fault-tolerant networks.

3.8 RELIABILITY/COST RATIO

A simple measure of the cost-effectiveness for reliability can be given by comparing MTTFs and the costs of these networks. Let the cost-effectiveness η of a network for reliability be the ratio of its MTTF to its cost. The MTTF/cost ratio of these networks relative to that of baseline network for both upper bound and lower bound are shown in Figures 3.4 and 3.5 respectively. From these graphs it is clear that MTTF/cost ratio of MNs are better than most other fault-tolerant networks.

3.9 PERFORMANCE MODEL

In this section, a model is described for circuit-switched MN performance. This model describes a numerical estimate of the probability that individual access requests can be successfully routed by an MN. The extension of this approach can be used to

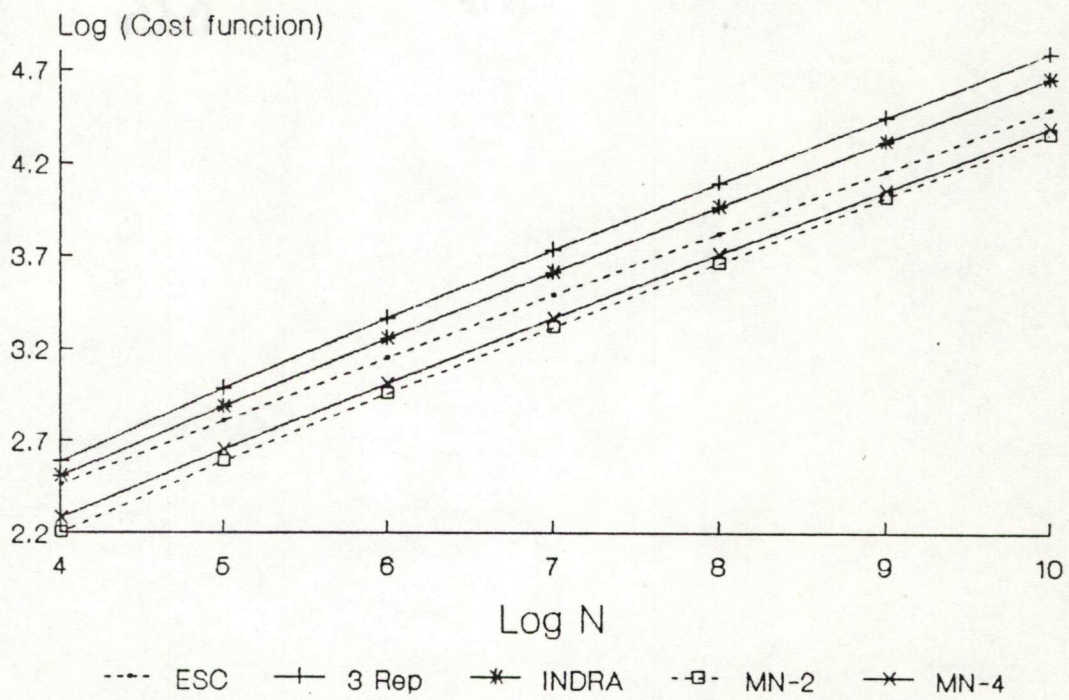


Figure 3.3 Cost function v/s Network Size.

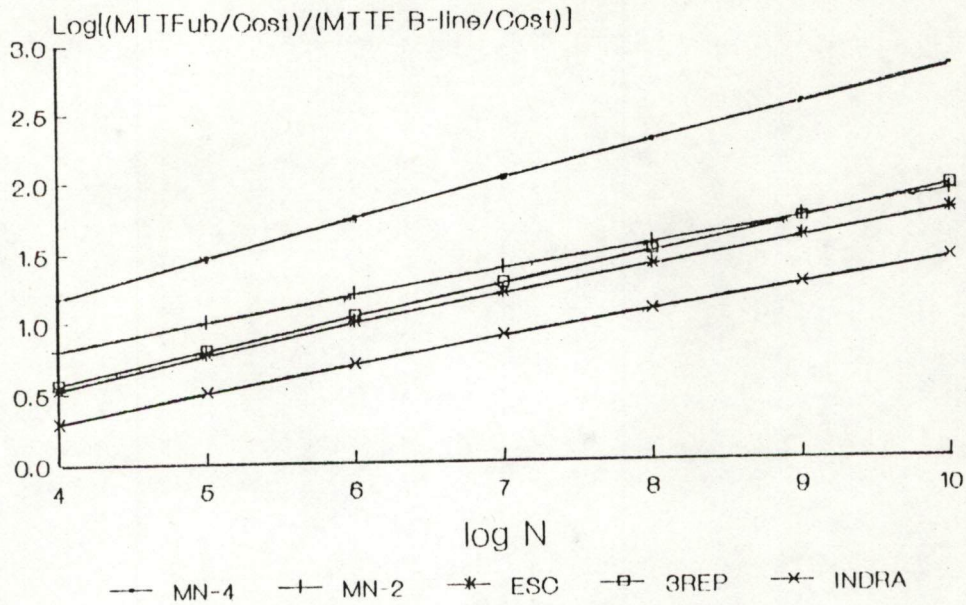


Figure 3.4 MTTFab/Cost ratio of Fault-Tolerant Networks w.r.t MTTFB-line/Cost ratio of Baseline Network.

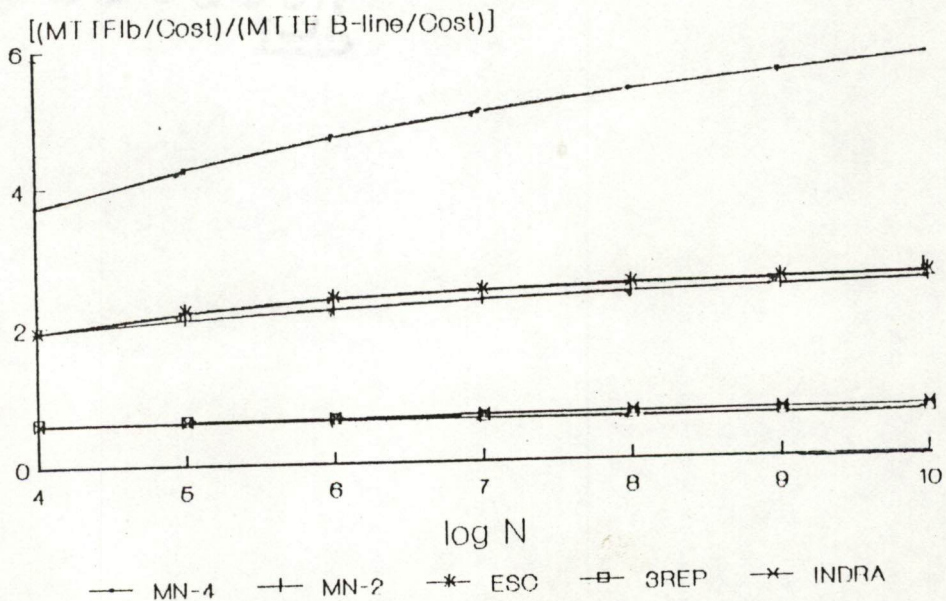


Figure 3.5 MTTF(lb)/Cost ratio of Fault-Tolerant Networks w.r.t MTTFB-line/Cost ratio of Baseline Network.

evaluate network performance in the presence of faults. Previous work [19,44,75,76,99,103] employed probability of acceptance (p_a) and bandwidth as primary performance measures. Sometimes these measures are inadequate. For example, if the network provides interconnection for a multiprocessor system, a processor receiving degraded service would become a bottleneck in a parallel computation. In order to critically examine the network, a measure known as minimum reference probability is analyzed, apart from the conventional measures such as the probability of acceptance and the bandwidth. The following assumptions are made in the performance model [13,44,70].

3.9.1 Assumptions

- i) Sources (processors) generate requests independently. (different sources could generate requests with different probability.) Requests are mutually independent within a cycle. Blocked requests are ignored and are not resubmitted. Thus, requests in one cycle are independent of requests submitted in previous cycles. Even though this part of the assumption is unrealistic, previous analytical and simulation studies for similar systems show that not making this assumption changes the results only slightly while making the analysis much more complex [70].
- ii) Path setup time is negligible. The message time is equal to the network cycle time.
- iii) Request are generated for each destination (memories) with equal probability. Thus, traffic at every link is

symmetrical w.r.t the destinations i.e a request on any link in the network is equally likely to be intended for any possible eventual destination. Thus, incoming requests to a switch are routed to each intrastage output of the switch with equal probability.

iv) the analysis is based on preferred path routing algorithm.

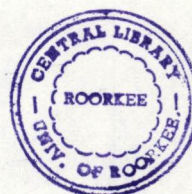
3.9.2 Performance Analysis

In this section, the performance analysis of MNs under both fault free and fault present conditions are presented to provide a quantitative measurement of the capabilities of MNs. The measures of the network performance are probability of acceptance (defined as the probability that a request submitted successfully finds a path to the destination), bandwidth (defined as total number of requests that can be routed through the network in a cycle) and the minimum reference probability (defined as the minimum of the probability of reference among all the destinations).

The symmetry of the network is lost when faults are present. So the load is different at different locations in the network, and because of this the switches in a stage cannot be treated as identical. Thus, the performance seen by individual network sources and destinations need to be considered.

If p is the probability of request generation by a source, then the output probability at the j th multiplexer is given by:

$$p_j = 1 - \prod_m (1 - p_m) \dots (1)$$



245896

where

$$p_m = \begin{cases} p & \text{if } m = j \text{ div } (N \text{ div } k) \\ p/x & \text{if } m \neq j \text{ div } (N \text{ div } k) \end{cases}$$

$$\begin{aligned} x &= 2^{n-i-1} && \text{if a switch or input link of a switch of } i\text{th stage} \\ x &= 2^n && \text{if a DEMUX} \end{aligned} \quad \left. \begin{array}{l} \text{(connected directly or} \\ \text{indirectly to the MUX,} \\ \text{corresponding to the } j\text{th} \\ \text{MUX in the preceding} \\ \text{module) is faulty} \end{array} \right\}$$

$$x = \infty \quad \text{if the module preceding the one of which } j\text{th MUX is a part i.e } [j \text{ div } (N \text{ div } k) - 1 + k] \bmod k \text{ is non faulty.}$$

As the output probability of a MUX becomes the input probability for a switch of stage (n-1), so the probability of presence of request at each input link of the switches of stage (n-1) are computed.

The distribution of traffic on the output links of a switch is calculated, given the distribution of traffic on its input links. Let p_i be the probability that a request is present at the i th input link of a switch and q_j be the probability that a request is present at the j th output link of a switch. Under a symmetric traffic assumptions all possible outputs are equally likely, so q_j is given by

$$q_j = 1 - \prod_i (1 - p_i/m) \quad \dots (2)$$

Where m represents the number of inputs and outputs of a switch.

If an input link or its associated switch is faulty, then its probability of carrying an input request is zero. If an output link is faulty or is connected to a faulty switch in the

next stage, then the probability of presence of request on that link is zero. The output probabilities would not change for the good links in the presence of faults. Since the output of a stage is the input to the next stage (except the first and the last stage), so the output probabilities can be recursively evaluated with the help of equation (2) starting from stage (n-1).

The requests which are blocked due to the faulty components in a module are to be routed by an alternate path through the first available alternate module having fault free paths corresponding to the faulty paths in the proceeding modules. So the input probabilities of the alternate module available for routing will increase to account for the additional load. The extra load in an alternate path is not uniformly directed to all the connected destinations. It is only directed to the SE which corresponds to the SE in the original path and thereafter it is directed uniformly to all the destinations. For a stage, if $p_{j,extra}$ is the extra probability of request present at the input link of a switch and $p_{j,normal}$ is the probability under fault free conditions at the j th input link of a switch, then the output probability at the i th link of the switch is given as:

$$q_i = 1 - \prod_j [1 - (p_{j,normal}/m + p_{j,extra})] \dots (3)$$

(for the output link to which the extra load is to be routed)

$$= 1 - \prod_j (1 - p_{j,normal}/m) \dots (4)$$

(for other output links)

The probability of presence of request at the output links of the switches of the final stage (i.e 0th stage) are the input probabilities for the demultiplexers. As there are no contention in the demultiplexers, so the probability of presence of request at each of the output links (leading to the destinations) is:

$1/k$ * (the probability of presence of request at the input link of the DEMUX)

If it is fault-free, and zero if it is faulty. From these probability of presence of request for connection to different destinations, the sum of probabilities of having a connection request over all the destinations, the minimum of the probabilities of reference to any of the destinations are computed. The sum of the probabilities of all the destinations gives the bandwidth of MNs. From this the probability of acceptance of request (p_a) is calculated

$$p_a = \text{Bandwidth}/p.N$$

The results of the performance improvement of MNs over unique-path baseline network are shown in Figures 3.6 and 3.7. Figure 3.6 shows the variation of p_a with the size of network for request generation probability of 1.0. Figure 3.7 shows the variation of p_a with the traffic routed through the network for network size of 128 x128.

The effect of different types of fault on MN performance are illustrated by the following example.

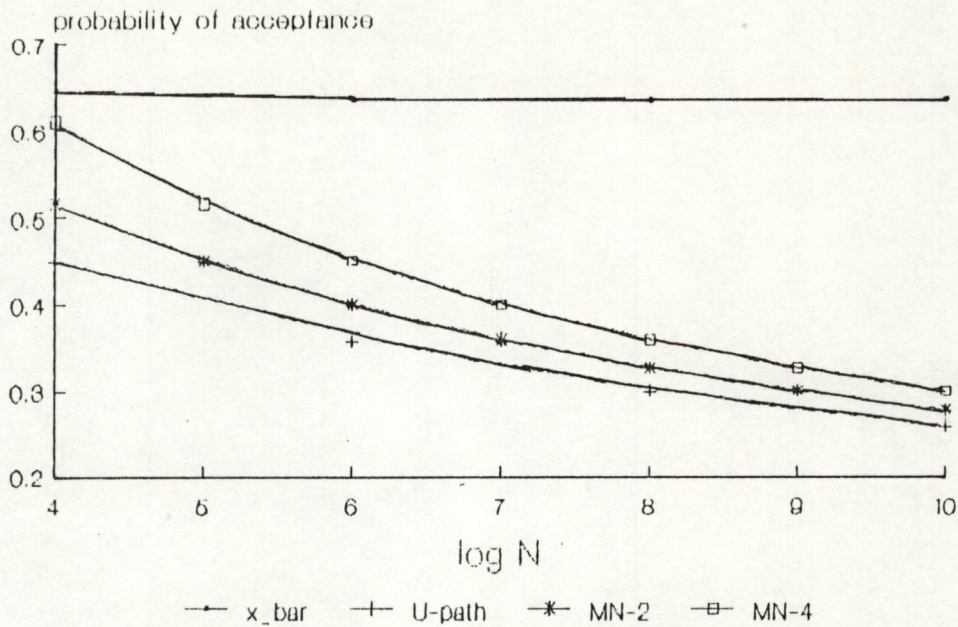


Fig.3.6 Probability of acceptance as fn. of Network size for MNs,U-path,Crossbar -at a request generation prob. of 1.0 .

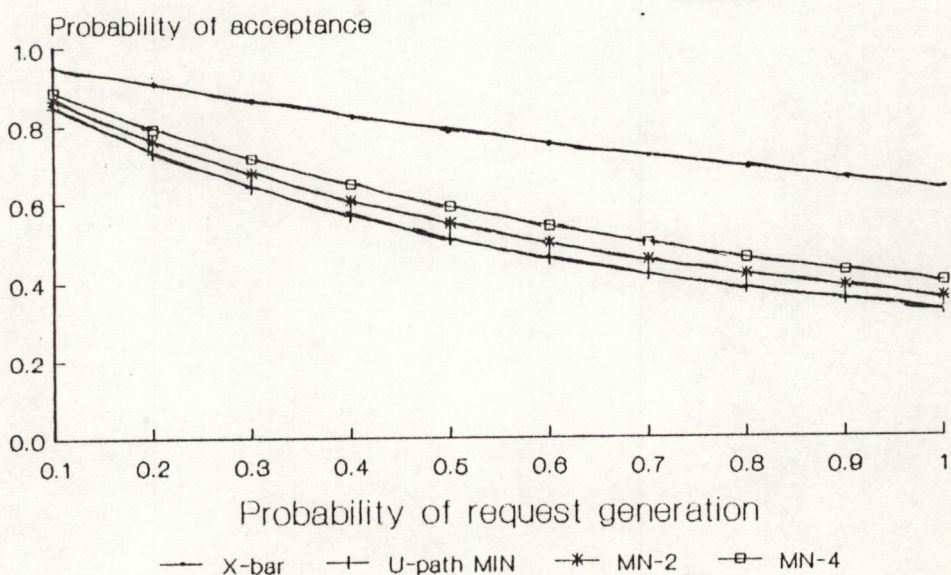


Fig.3.7 Probability of acceptance as fn. of probability of request gen. for MNs, U-path and Crossbar for Network size 128

Example

In this section, the effect of faults on the performance of MN of size 128 x 128 is evaluated under a variety of fault scenarios and for request generation probabilities of 1.0, 0.5, and 0.1. The performance is evaluated under the following conditions:

- i) No faults
- ii) a MUX fault
- iii) a switch fault
- iv) a DEMUX fault
- v) a module fault

For each category, the effect of fault in each stage of the network is analysed. In all there are ten performance models i.e no faults, a MUX fault, six different switch faults, a DEMUX fault and a module fault. The results of the performance are summarized in Tables 3.3, 3.4 and 3.5 for request generation probabilities of 1.0, 0.5 and 0.1 respectively. First column of each tables indicate the type of fault, II, III, V columns indicate the probability of acceptance, bandwidth and minimum reference probability respectively. Columns IV and VI show the percentage change in the probability of acceptance/bandwidth and minimum reference probability caused by each type of fault. The results show that the degradation in the probability of acceptance and bandwidth under such failures is very small 0(2%).

Table 3.3

Network Size: 128

Prob. of req.generation by any processor: 1.00

Sr. No.	Fault	Prob.of acceptance	Bandwidth % change	Min.ref. prob. %change
1	No Fault	0.3594	46.00 0.0	0.3594 0.0
2	MUX	0.3594	46.00 0.0	0.3594 0.0
3	S5	0.3571	45.70 0.7	0.3571 0.7
4	S4	0.3564	45.61 0.8	0.3533 1.7
5	S3	0.3557	45.53 1.0	0.3445 4.1
6	S2	0.3550	45.44 1.2	0.3245 9.7
7	S1	0.3544	45.36 1.4	0.2795 22.2
8	S0	0.3538	45.28 1.6	0.1797 50.0
9	DEMUX	0.3566	45.64 0.8	0.1797 50.0
10	MODULE	0.1797	23.00 50.0	0.1797 50.0

(Si indicate the switch of ith stage)

Table 3.4

Network Size: 128

Prob. of req.generation by any processor: 0.50

Sr. No.	Fault	Prob.of acceptance	Bandwidth % change	Min.ref. prob. %change
1	No Fault	0.5466	34.98 -	0.2733 -
2	MUX	0.5466	34.98 -	0.2733 -
3	S5	0.5435	34.78 0.6	0.2717 0.6
4	S4	0.5422	34.70 0.8	0.2690 1.6
5	S3	0.5412	34.64 1.0	0.2626 3.9
6	S2	0.5403	34.58 1.1	0.2484 9.1
7	S1	0.5395	34.53 1.3	0.2168 20.7
8	S0	0.5387	34.48 1.4	0.1476 46.0
9	DEMUX	0.5426	34.73 0.7	0.1461 46.5
10	MODULE	0.3271	20.93 40.2	0.1636 40.2

Table 3.5

Network Size: 128

Prob. of req.generation by any processor: 0.10

Sr. No.	Fault	Prob.of acceptance	Bandwidth % change	Min.ref. prob.	%change	
1	No Fault	0.8668	11.10	-	0.0867	-
2	MUX	0.8668	11.10	-	0.0867	-
3	S5	0.8651	11.07	0.2	0.0865	0.2
4	S4	0.8641	11.06	0.3	0.0861	0.6
5	S3	0.8633	11.05	0.4	0.0853	1.6
6	S2	0.8625	11.04	0.5	0.0832	4.0
7	S1	0.8618	11.03	0.6	0.0787	9.2
8	S0	0.8612	11.02	0.7	0.0686	20.9
9	DEMUX	0.8639	11.06	0.3	0.0678	21.7
10	MODULE	0.7321	09.37	15.5	0.0732	15.5

Further, the performance results show that, though, faults do not significantly reduce the average performance (i.e. probability of acceptance and bandwidth), the performance degradation seen by individual memories is large. Faults closer to the memories affect lesser number of memories, but affect them more strongly. If a module is faulty then the performance is reduced by 50% (for request generation probability of 1.0), but for light traffic (for request generation probability of 0.1) even module fault does not affect much, maximum degradation is less than 22%. Also faults closer to the sources affect more number of destinations, but affect them mildly. However, the associated sources though lesser in number are affected very badly as the sources find it increasingly difficult to route their requests because of faults.

3.10 CONCLUSION

In this chapter, a class of regular fault-tolerant multistage interconnection networks named as modular networks (MNs) has been proposed. It has been shown that reliability/cost ratio of MNs is better than that of most other fault-tolerant networks. Further, MNs provide higher probability of acceptance and bandwidth than those of unique path networks. The effect of faults on the performance of MNs has also been analysed. It has been shown that faults do not significantly affect overall network performance, but still severely degrade the performance of some parts of the system. As the number of faults increase, the performance of MNs decreases. However, for light traffic and

small size networks the performance degradation is not significant. Hence, the proposed modular networks are quite attractive for low traffic and small size networks.

AUGMENTED BASELINE NETWORK

4.1 INTRODUCTION

This chapter introduces a class of dynamically reroutable regular fault-tolerant multistage interconnection network named as augmented baseline network (ABN), which can achieve significant tolerance to faults and good performance with relatively low cost and simple control scheme. The proposed ABN technique results in reduced number of stages compared to the previously proposed fault-tolerant MINs. Fault-tolerance in ABN is achieved by chaining switches within a stage. The characteristics of ABN pertaining to performance, reliability and cost-effectiveness are analyzed.

4.2 CONSTRUCTION

To construct an ABN of size $N \times N$ i.e with N sources and N destinations, two identical groups of $N/2$ sources and $N/2$ destinations need to be formed first. Each group consists of a multiple path, modified baseline network of size $N/2$. The modified baseline network is a baseline network with one less stage, and feature links among switches belonging to the same stage and forming several loops of switches. The switches in the last stage are of size 2×2 and the remaining switches in stages 1 through $n-3$ ($n = \log_2 N$) are of size 3×3 . In each stage, the switches can be grouped into conjugate pairs i.e each switch in

such a pair has the same successor switches in the next stage. These conjugate pairs can then be grouped into conjugate subsets, where a conjugate subset is composed of all switches in a particular stage that lead to the same subset of destinations. The modified baseline network achieves the multiple path property by permitting two switches in the same conjugate subset that are not a conjugate pair to communicate through auxiliary links. The switches which communicate through the use of auxiliary links form a loop. The conjugate loops are formed in such a way that the two switches which form a loop have their respective conjugate switches in a different loop.

Each source is connected to both the groups via multiplexers. There is one 4x1 MUX for each input link of a switch in stage 1 and one 1x2 DEMUX for each output link of a switch in stage $n-2$. Each group consisting of a modified baseline network of size $N/2 \times N/2$ plus its associated multiplexers and demultiplexers is called a subnetwork. Thus an ABN consists of two identical subnetworks which are denoted by G^i . The switches in a stage of a subnetwork are being numbered from top to bottom as $0, 1, \dots, (N/4 - 1)$. For example, in Figure 4.2, switches 0, 1, 2, 3 belonging to stage 1 of a subnetwork (G^i) form a conjugate subset; within that subset, switches 0 and 1 form a conjugate pair; and the two loops consisting of switches 0, 2 and 1, 3 form conjugate loops.

Let the source S and destination D be represented in binary code as:

$$S = s_0 \cdot s_1 \cdot \dots \cdot s_{n-2} \cdot s_{n-1}$$

$$D = d_0 \cdot d_1 \cdot \dots \cdot d_{n-2} \cdot d_{n-1}$$

A source selects a particular subnetwork (G^i) based upon the most significant bit of the destination i.e $i = d_0$. As there are two paths between a source-destination pair, so each source is connected to the two switches (primary and secondary) in a subnetwork. The sources are connected to the switches of stage 1 as follows:

- i) A source S is connected to the $(s_1 \cdot \dots \cdot s_{n-2})$ primary switch in both the subnetworks through the multiplexers.
- ii) A source S is also connected to the $\{(s_1 \cdot \dots \cdot s_{n-2}) + 1\} \bmod N/4$ secondary switch in both the subnetworks through the multiplexers.

Thus an ABN of size $N \times N$ consists of N 4×1 multiplexers, N 1×2 demultiplexers, and $n-2$ stages of $N/2$ switches each (the last stage has 2×2 switches and the remaining stages have 3×3 switches). Observe that this construction procedure has two benefits. First, the network can tolerate the failure of any switch in the network. Second, it provides a topology which lends itself to on-line repair and maintainability, as a loop can be removed from any stage of the ABN without disrupting the operation of the network. Since the subnetworks are identical, so VLSI implementation of the network becomes simple. The

construction procedure given in this section can be applied to a broad class of unique-path networks which are topologically equivalent [4,67,109]. A network resulting from his construction is called augmented baseline network (ABN).

An ABN of size 16x16 along with its redundancy graph is illustrated in Figure 4.2 and the multiple paths between S=0000 and D=0100, and between S=0000 and D=1001 are being highlighted in Figure 4.3. A baseline network is already given in section 2.3 and is again shown in Figure 4.1 for comparison with ABN.

4.3 ROUTING SCHEME

In this section, the routing scheme of ABN is considered in the case that each source-destination pair tries to utilize only one path at a time. The non-backtracking scheme given below is easy to implement and performs quite well. This scheme assumes that sources and switches have the ability to detect faults in the switches to which they are connected. Faults in MINs can be detected by the application of test inputs [3,30], or by employing concurrent error-detection at the network level or at the switch level [33]. The ABNs are self routing. A request from any source S to a given destination D is routed through the ABN as:

- i) The source S selects one of the subnetwork G^i based on the most significant bit of the destination D ($i=d_0$).
- ii) There are two paths i.e primary and secondary, between each source-destination pair. Each source attempts entry into the ABN via its primary path. If the primary path is faulty (i.e

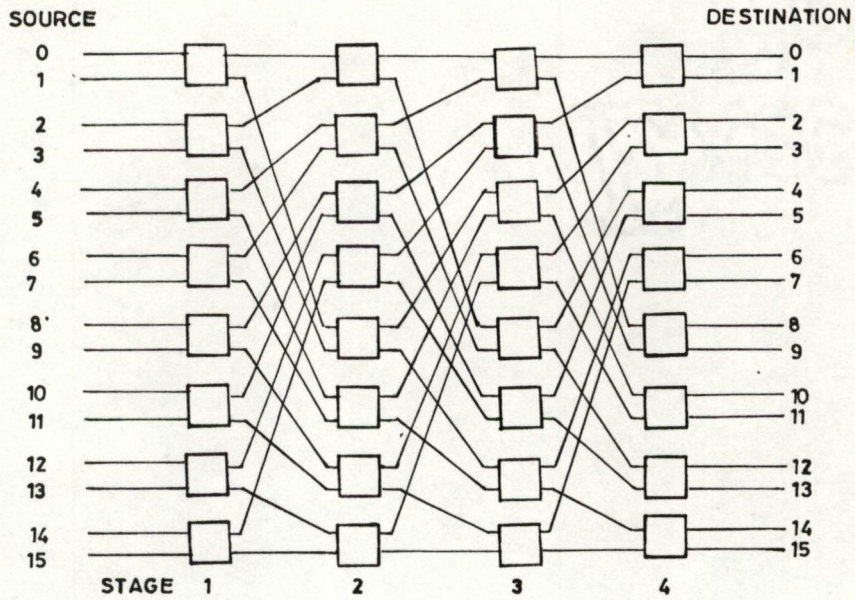
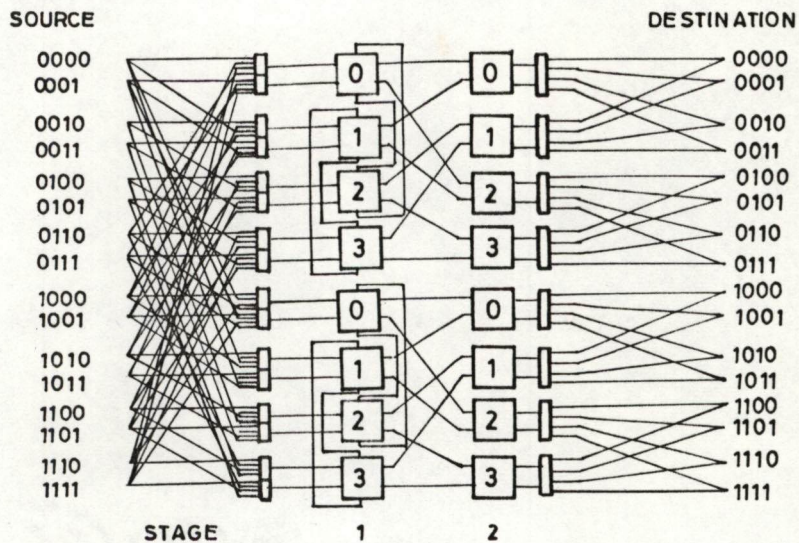
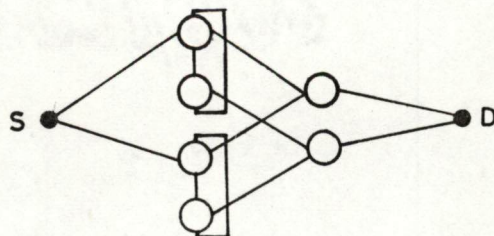


FIG. 4.1 BASELINE NETWORK OF SIZE 16



(a)



(b)

FIG. 4.2(a) AN ABN OF SIZE 16

(b) THE REDUNDANCY GRAPH

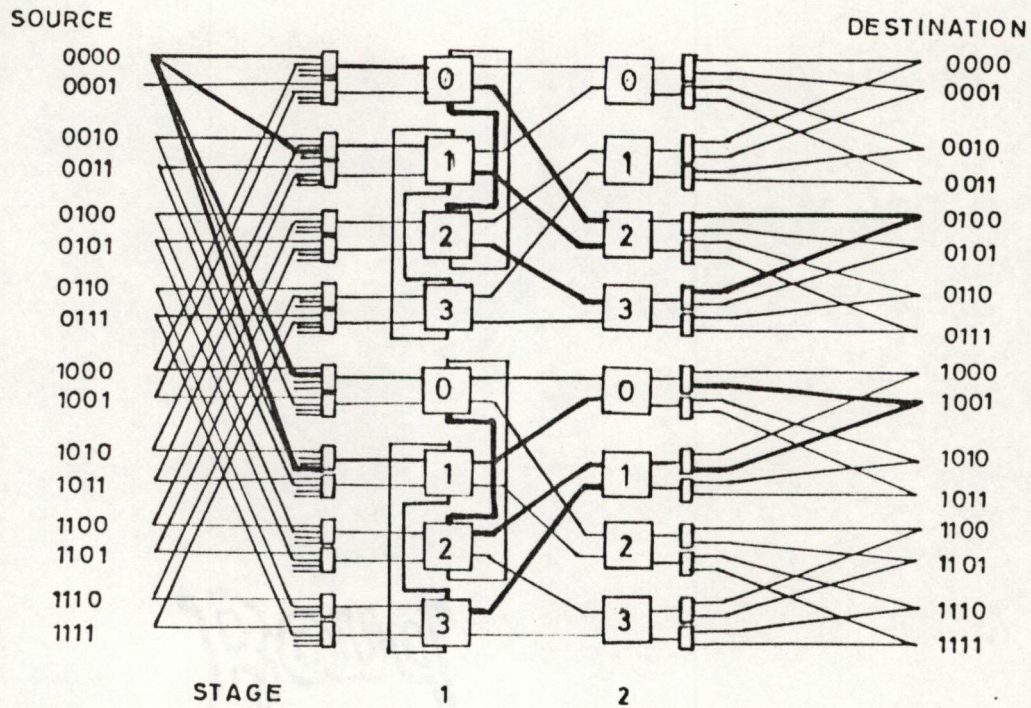


FIG. 4.3 An ABN of size 16, highlighting the multiple paths between S-D pair

either MUX or primary switch or both are faulty), then the request is routed to the secondary path. If the secondary path is also faulty, then the ABN is failed.

- iii) After the multiplexer, the routing of the request in the intermediate stages of the subnetwork depends upon $(n-2)$ tag bits. The routing tag of the ABN is the destination address with its most significant bit being trimmed (i.e. routing tag = $d_1 \dots d_{n-2} \cdot d_{n-1}$). For each switch in stage i ($i < n-2$), use tag bit d_i and route the request through the usual output link, if it is busy or if the successor switch (in the next stage) is faulty, route the request via the auxiliary output link to the other switch in the loop with the same tag bit d_i . If the auxiliary link is also unusable because it is busy or because of a fault, then drop the request. A faulty DEMUX at the output of the ABN is regarded as a failure of its associated switch in stage $n-2$. This strategy essentially enables a switch to detect a failure of its successor switch and reroutes the request whenever possible.
- iv) For a request at a switch in stage $n-2$, use bit d_{n-2} of the routing tag and route the request accordingly to one of the output links. If the required output link is busy, drop the request.
- v) For routing a request through a DEMUX, use bit d_{n-1} of the routing tag.

Since there are two choices to route at each step, except in $(n-2)$ th stage (where it is assumed that a fault in a DEMUX at

the output of a switch is a fault in that switch and that the destinations are fault free), it is clear that the routing procedure delivers a request from a source to any required destination in the presence of single switch failure. It can be easily seen that the routing algorithm of ABN is simple and routing complexity is comparable to that of the unique-path networks of the same size.

4.4 FAULT-TOLERANCE

Since ABN provide two disjoint paths for each source-destination pair, so ABN is a single switch fault-tolerant. The switch faults that ABN is guaranteed to survive, and the faults that will always cause the network to fail is first characterized. From Figure 4.2, it is clear that a minimum of two switch faults can disconnect all available paths between some source-destination pair. For example, if both the switches that a source or destination is connected become faulty, then that source or destination becomes disconnected from the rest of the network. However, if such critical faults do not occur, it is possible to tolerate multiple switch faults. To highlight the role that $S_{i,j}$ and its conjugate switch $S^C_{i,j}$ and their co-loop switches - $co(S_{i,j})$, $co(S^C_{i,j})$ play in providing the connections between source-destination pairs, the redundancy graph of an ABN is drawn and is shown in Figure 4.4. $S_{i,j}$ represent a switch i in stage j . The following theorem characterizes the faults tolerated in the ABN.

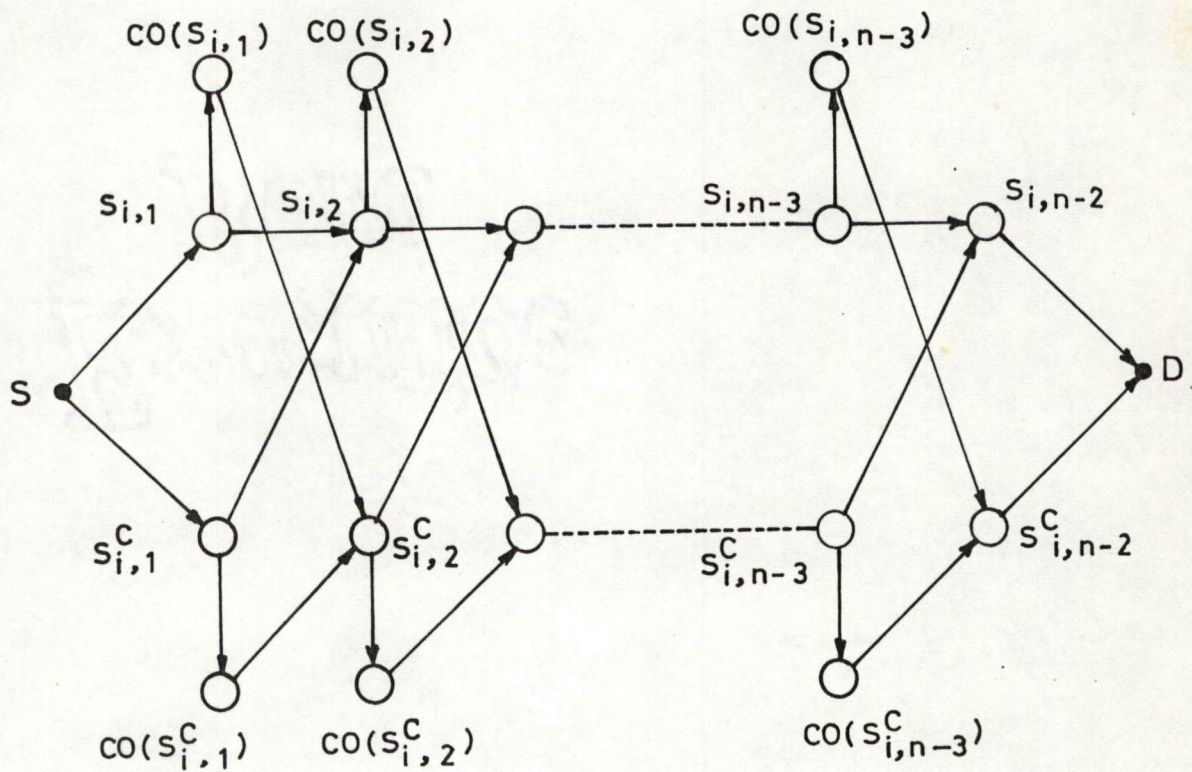


FIG. 4.4 REDUNDANCY GRAPH OF ABN FOR THE PROOF OF THEOREMS 2 AND 3

Theorem 1: In the ABN, there exists at least one fault-free path from any source to any destination, if faults occur such that at most one loop is affected in every pair of conjugate loops, and at most one switch is affected in every conjugate pair of switches in the last stage.

Proof: Since the switch faults are confined to at most one loop in every conjugate loop pair, either $(S_{i,j}$ and $co(S_{i,j}))$ or $(S^C_{i,j}$ and $co(S^C_{i,j}))$ are guaranteed to be fault-free for $1 \leq j < n-2$ where i represent the switch number in a stage. It is easy to see from the redundancy graph that a request from S to D can find a path through fault-free switches up to the switch $S_{i,n-3}$ or $S^C_{i,n-3}$ (whichever is fault-free). From stage $n-3$ to D , a path exists if either $S_{i,n-2}$ or $S^C_{i,n-2}$ is fault-free. Since at most one of these two switches is assumed to be faulty a path exists from S to $S_{i,n-3}$ (or $S^C_{i,n-3}$) and from $S_{i,n-3}$ (or $S^C_{i,n-3}$) to D .

Q.E.D.

There are certain types of multiple faults that will always cause some source to be disconnected from some destination and these are characterized by the following Theorem:

Theorem 2: In ABN, some source is disconnected from some destination, if the conjugate pair of switches i.e $S_{i,j}$ and $S^C_{i,j}$ are faulty at the same time.

Proof: Suppose the path from a source S to a destination D contains a faulty switch $S_{i,j}$ then all the available paths from S to D must pass through either $S_{i,j}$ or $S^C_{i,j}$ in stage j . Thus if

both the $S_{i,j}$ and $S^C_{i,j}$ are faulty at the same time, then S is disconnected from D .

Q.E.D

It may be mentioned here that in an ABN, if in every pair of conjugate loops at least one loop is free from faults, then the redundancy graph of every source-destination pairs remains connected. This is the reason why ABNs lend themselves to easy repair and maintenance. Loops of switches are implemented as replaceable modules; when a switch is identified to be faulty, the loop containing the faulty switch is replaced by a fault-free loop. The operation of the network need not be disrupted during the repair period since requests for connection through the loop being replaced can be routed through its conjugate loop.

4.5 RELIABILITY ANALYSIS-Mean Time To Failure

In this section, the effectiveness of the multiple paths in improving the reliability of the MINs is analysed. The reliability analysis is in terms of MTTF. To make the analysis of MTTF tractable, certain assumptions have to be made. The assumptions made in the analysis of ABN are similar to the ones that have been made previously in the other studies of fault-tolerant networks and are given in section 3.5.

4.5.1 Augmented Baseline Network (ABN)

The adaptive routing scheme described in section 4.3 considers a 2×2 switch in the last stage and its associated demultiplexers as a series system, so these three elements are considered as a single component (SE_{2d}), and based on gate count,

a failure rate of $\lambda_{2d} = 2\lambda$ can be assigned to this group of elements. Also let λ_3 be the failure rate for the 3x3 switch (SE_3), then based on gate count, $\lambda_3 = 2.25\lambda$.

Upper Bound

To obtain an upper bound for the ABN, it is observed that each source is connected to two multiplexers in each subnetwork, and each switch has a conjugate. So if it is assumed that the ABN is operational as long as one of the two multiplexers attached to a source (in a particular subnetwork) is operational and as long as a conjugate pair (loop or switch) is not faulty, then the network will permit as many as one half of the components to fail and the ABN may still be operational. This permits a simple reliability block diagram of the upper (optimistic) bound as shown in Figure 4.5(a).

The expression for the upper bound of the ABN reliability is:

$$R_{ABN-ub}(t) = [1 - (1 - e^{-\lambda_m t})^2]^{N/2} \cdot [1 - (1 - e^{-\lambda_3 t})^2]^{N/4} \cdot (n-3) \cdot [1 - (1 - e^{-\lambda_{2d} t})^2]^{N/4}$$

$$MTTF_{ABN-ub} = \int_0^{\infty} R_{ABN-ub}(t) \cdot dt$$

Lower Bound

At the input side of the ABN, the routing scheme does not consider the multiplexers to be an integral part of a 3x3 switch. For example, as long as at least one of the two multiplexers

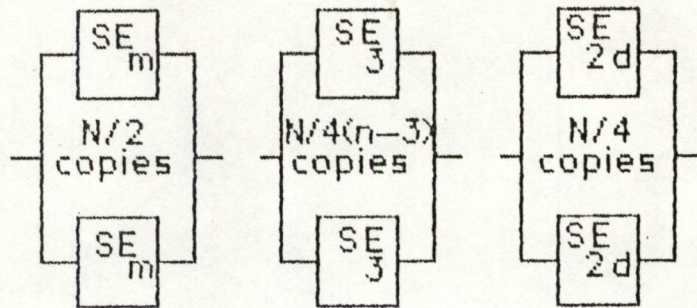


FIG. 4.5(a) Reliability block diagram of ABN for the evaluation of MTTF upper bound

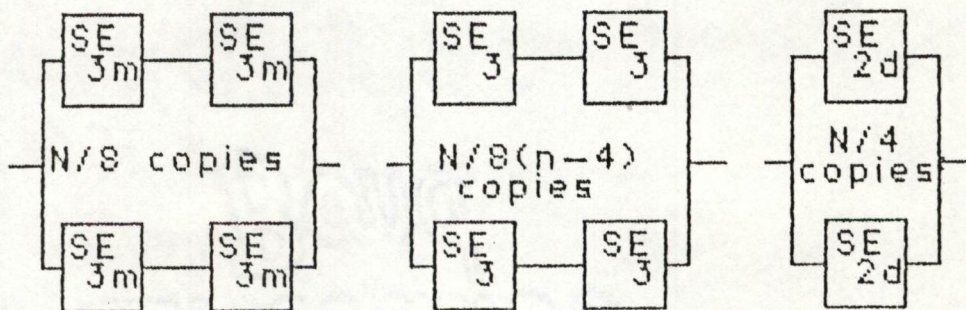
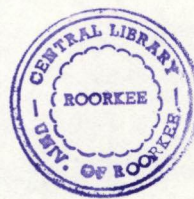


FIG. 4.5(b) Reliability block diagram of ABN for the evaluation of MTTF lower bound

attached to a particular switch is operational, switch can still be used for routing. Hence, if the two multiplexers are grouped with each switch in the input side and are considered as a series system (SE_{3m}), then a conservative estimate of the reliability of these components are obtained. Their aggregated failure rate will be $\lambda_{3m} = 4.25 \lambda$. Finally these aggregated components and the switches in the intermediate stages can be arranged in pairs of conjugate loops. To obtain the lower (pessimistic) bound on the reliability of ABN, it is assumed that the network is failed whenever more than one conjugate loop has a faulty element or more than one conjugate switch in the last stage fails. The reliability block diagram is shown in Figure 4.5(b). The expression for the lower bound of the ABN reliability is:

$$R_{ABN-1b}(t) = [1 - (1 - e^{-2\lambda_{3m}t})^2]^{N/8} \cdot [1 - (1 - e^{-2\lambda_3 t})^2]^{N/8(n-4)} \cdot [1 - (1 - e^{-\lambda_{2d}t})^2]^{N/4}$$

$$MTTF_{ABN-1b} = \int_0^{\infty} R_{ABN-1b}(t) \cdot dt$$



4.5.2 Unique-Path Baseline Network

Although unique-path baseline network is not a fault-tolerant network, but it is used as a yardstick to measure the reliability improvement of ABN. As any single fault leads to a network failure, MTTF for baseline network can be easily calculated as:

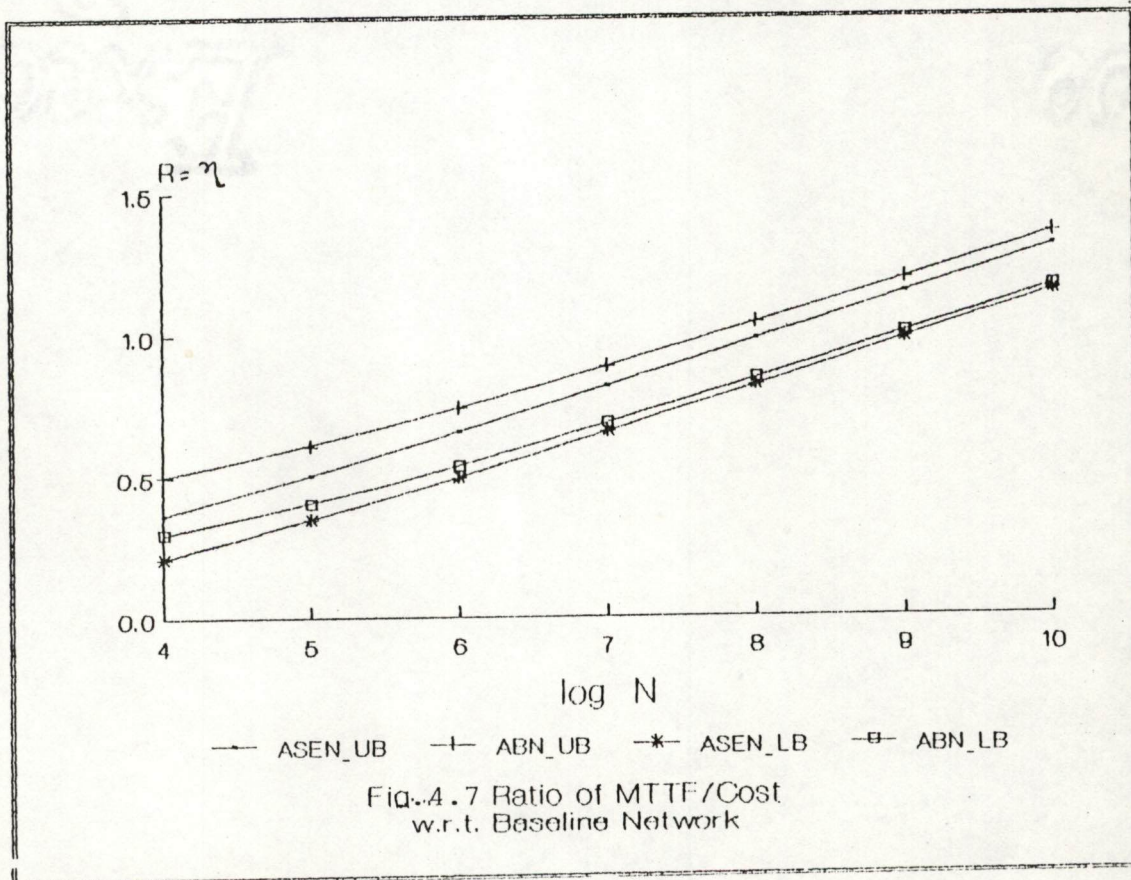
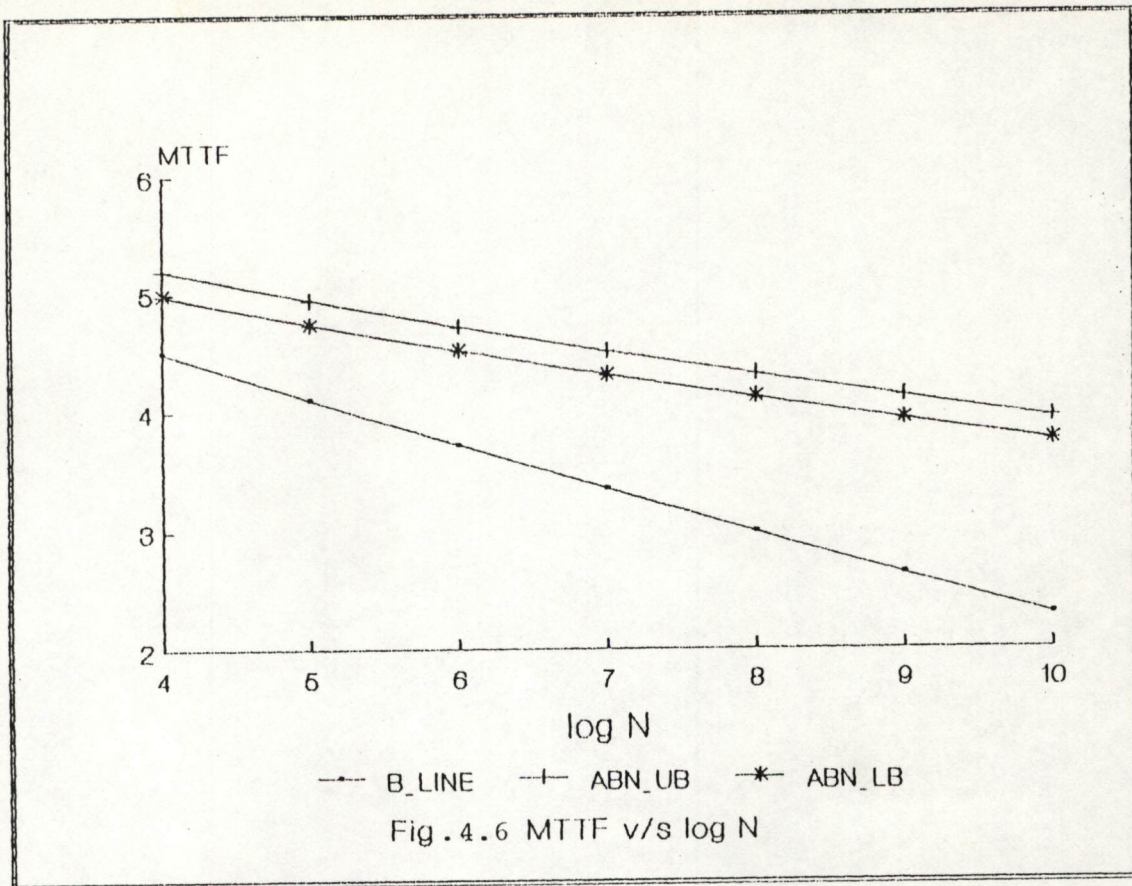
$$MTTF_{baseline} = \int_0^{\infty} [e^{-\lambda t}]^{N/2 \cdot n} \cdot dt$$

The reliability improvement of ABN over unique-path baseline network, for various network sizes is illustrated by the graph shown in Figure 4.6.

4.6 COST-EFFECTIVENESS

It is observed that ABN can provide higher or at least equal reliability compared to some other fault-tolerant networks. However, if such high reliability comes at the expense of high cost, it may have little value in practice. In this section, the cost-effectiveness of ABN is computed and the results are compared with ASEN [46] - a network with a similar fault-tolerant capability, but with increased network complexity. The assumptions for the calculation of cost are the same as described earlier (section 3.7) and the cost function of an ABN comes out to be $(9n-11)N/2$.

Now, a simple measure of the cost-effectiveness for reliability can be given by comparing MTTF and the cost of the network. Let the cost-effectiveness η of a network for reliability be the ratio of MTTF to its cost. To highlight the cost-effectiveness of ABNs, the cost-effectiveness of the ABNs and ASEN-2 relative to that of baseline network (for both upper and lower bounds) are evaluated and compared, and the improvement in results are shown in Figure 4.7. From the results, it can be observed that ABN is more cost-effective than most other fault-tolerant networks.



4.7 PERFORMANCE ANALYSIS

In this section, the performance of ABN is analysed under fault-free and fault-present conditions. The measure of the performance of ABN are probability of acceptance, bandwidth and minimum reference probability. Certain assumptions are made to simplify the analysis. The first three assumptions are the same as used for MNs and the fourth one is stated as:

Contention in switches is resolved randomly. The only exception is that requests from intrastage links are with lower priority than the interstage requests.

Let p be the probability of request generation by a source. The probability of request at the output link of a i th MUX is:

$$q_i = 1 - \prod_j (1 - p_j) \quad \dots (1)$$

where p_j is the probability of the presence of request at the j th input link of the i th MUX.

$$p_j = \begin{cases} p/2 & \text{for } j = 0 \text{ or } 1 \\ 0 & \text{for } j = 2 \text{ or } 3, \text{ if the primary path is} \\ & \text{fault-free} \end{cases}$$

otherwise

$$p_j = p/2$$

As the output probabilities of the multiplexers becomes the input probabilities for the switches of stage 1, so the probability of presence of request at each input link of the switches are computed.

Let p_j is the probability of presence of request at the input link j of a switch and q_i be the probability of request on i th output link. The probability of carrying an input request by a faulty link or its associated faulty switch will be zero. For a symmetric traffic assumption, all possible outputs are equally likely. Thus the probability that an output link i receives at least one request is given by:

$$q_i = 1 - \prod_j (1 - p_j/m) \quad . . . (2)$$

Where m indicates the number of input and output interstage links of a switch. q_i will be zero for faulty output links or for the links that are connected to the faulty switches in the next stage. The output probabilities would not change for the non-faulty links, because of the presence of faults. The network is analysed stage by stage and the probability of presence of request is obtained for all links in the network, and for the destinations. This is the probability that the output link carries a request without considering the traffic carried by the auxiliary links. The auxiliary links carry a part of the traffic blocked in the switches of a loop and route it out of the stage through an appropriate interstage output link of a switch wherever it is possible. The extra traffic routed through the switch is the request blocked in the other switch (i.e switch 1) of the loop and for which appropriate output link of the switch under consideration (switch 1' of the loop) is not busy. So, the extra traffic routed at the i th output link of a stage is given as:

$$q_{i,extra} = (\text{Prob. of no request at } i\text{th link}) \cdot (\text{Prob. of presence of request at the auxiliary link of the concerned switch}) / (\text{number of interstage output links of the switch}) \quad \dots (3)$$

$$\text{Prob. of no request at the } i\text{th link} = (i - q_i) \quad \dots (4)$$

Prob. of presence of request at the auxiliary link of switch 1' of the loop

$$= \text{prob. of request blocked in switch 1 of the loop}$$

Now

Prob. of request blocked in switch 1 of the loop is given as:

$$P_{block} = 1 - (\text{prob. of presence of } k \text{ input requests}) \cdot (\text{prob. of no collision for } k \text{ input requests and } f \text{ faulty output links}) \quad \dots (5)$$

Probability that there are k requests at a time at the input links of a switch is the sum of probability of existence of every permutation of presence of request at k input links and absence of request at rest of the input links of the switch.

prob. of presence of k requests at the input links of a switch

$$= \sum (\text{prob. of a permutation possible})$$

$$= \sum_{\sum m_s = k} \left[\prod_m A(m, m_s) \right] \quad \dots (6)$$

where

$$m_s = \begin{cases} 1 & (\text{presence of request}) \\ 0 & (\text{absence of request}) \end{cases}$$

$$A(m,0) = 1 - p_m$$

$$A(m,1) = p_m$$

m varies in such a way that all the interstage input links of the switch concerned are covered.

The probability that there is no collision due to contention among requests at input links in the presence of f faulty output links (or output links are connected to faulty switches in the next stage) is given by the counting argument as:

prob.(no collision /k input request/ f faulty links)

$$= (m - f)! / (m^k * (m - f - k)!) \quad \dots (7)$$

So from equations (5), (6) and (7)

$$P_{\text{block}} = 1 - \sum_{k=0}^{m-f} \left[\sum_{\sum m_s = k} \left(\prod_m A(m, m_s) \right) * (m - f)! / (m^k * (m - f - k)!) \right] \quad \dots (8)$$

Thus, the modified output probabilities for each switch is given as:

$$\begin{aligned} q_i^* &= q_i + q_{i,\text{extra}} \\ &= q_i + P_{\text{block}} \cdot (1 - q_i) / m \quad \dots (9) \end{aligned}$$

The modified output probability combine the requests from the interstage and the intrastage input links of a switch. From the probabilities of presence of request at input links of the switch of a stage, the probabilities of presence of request at

the output links are calculated according to equation (2) and then modified according to equation (9) for every stage except the last. No modification is required for the last stage as it does not contain any loop. The probabilities of presence of request at the output links of the switches of the final stage are the input probabilities for the demultiplexers. As there is no contention in the demultiplexer, so the probability of request at the output link of a demultiplexer is one half of the probability at its input link. Probability of presence of request at the output links of demultiplexers are the probability of request reaching the destination.

The results of the performance improvement of ABN over unique-path baseline networks are shown in Figures 4.8 and 4.9. Figures 4.8(a) and 4.8(b) shows the variation of probability of acceptance (p_a) with network size for request generation probabilities of 1.0 and 0.5 respectively. Figures 4.9(a) and 4.9(b) shows the variation of probability of acceptance (p_a) with the traffic routed through the network for network sizes of 256 and 1024 respectively. The figures for the crossbar networks is also included for comparison.

The effect of different types of fault on the performance of ABN is illustrated by the following example.

Example

In this example, the effect of faults on the performance of ABN of size 128 x 128 is evaluated for request generation probabilities of 1.0, 0.5, and 0.1. The performance is evaluated under the following conditions:

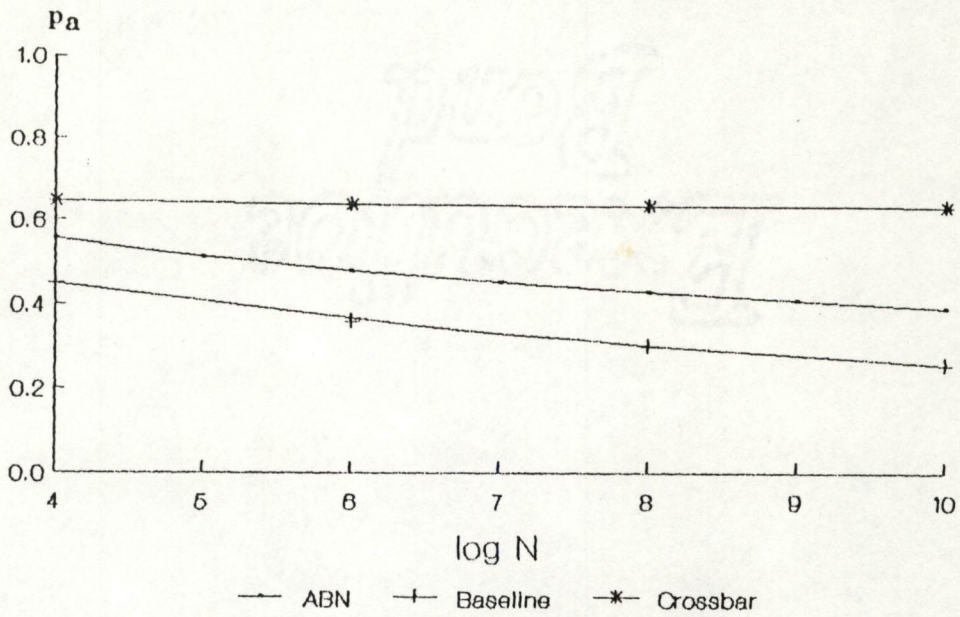


Fig. 4.8(a), probability of acceptance v/s $\log N$ - request generation probability of 1.0 .

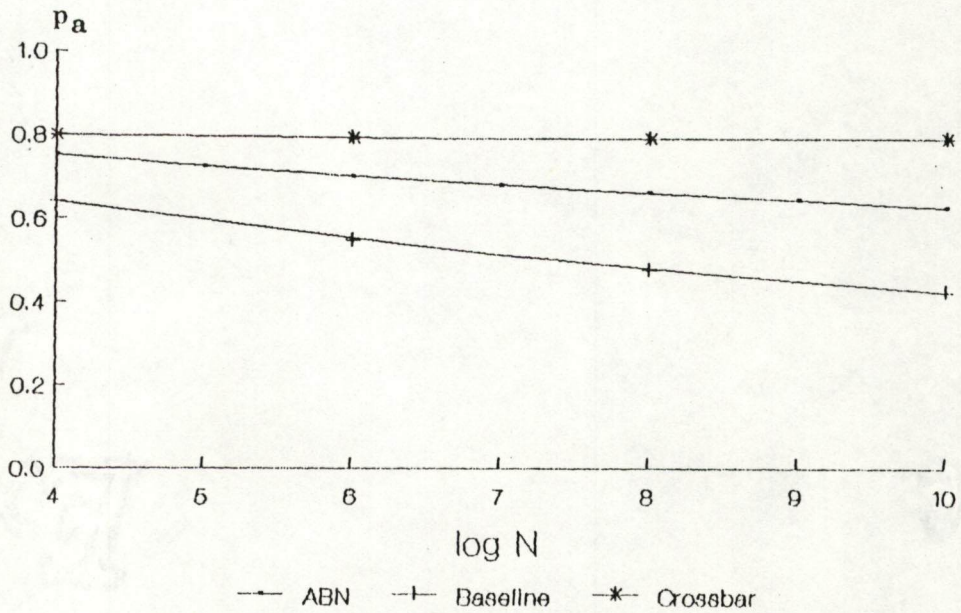
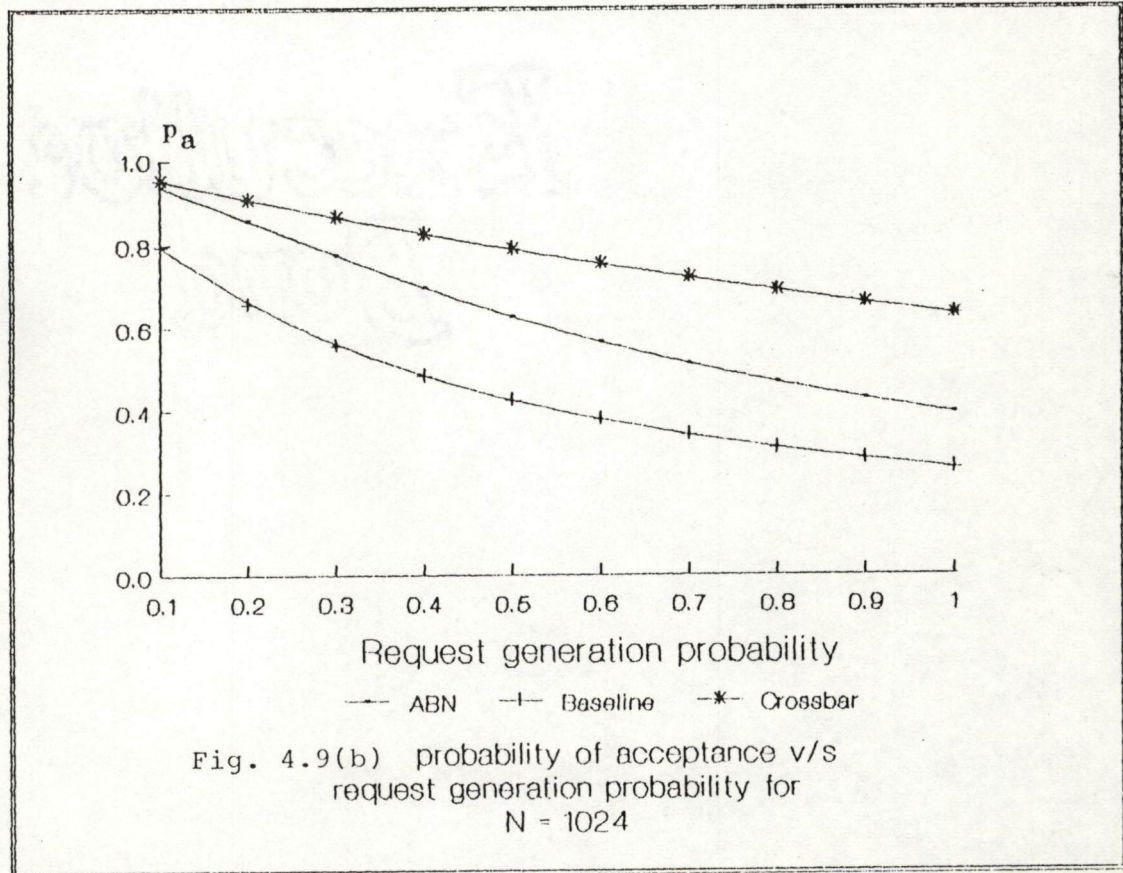
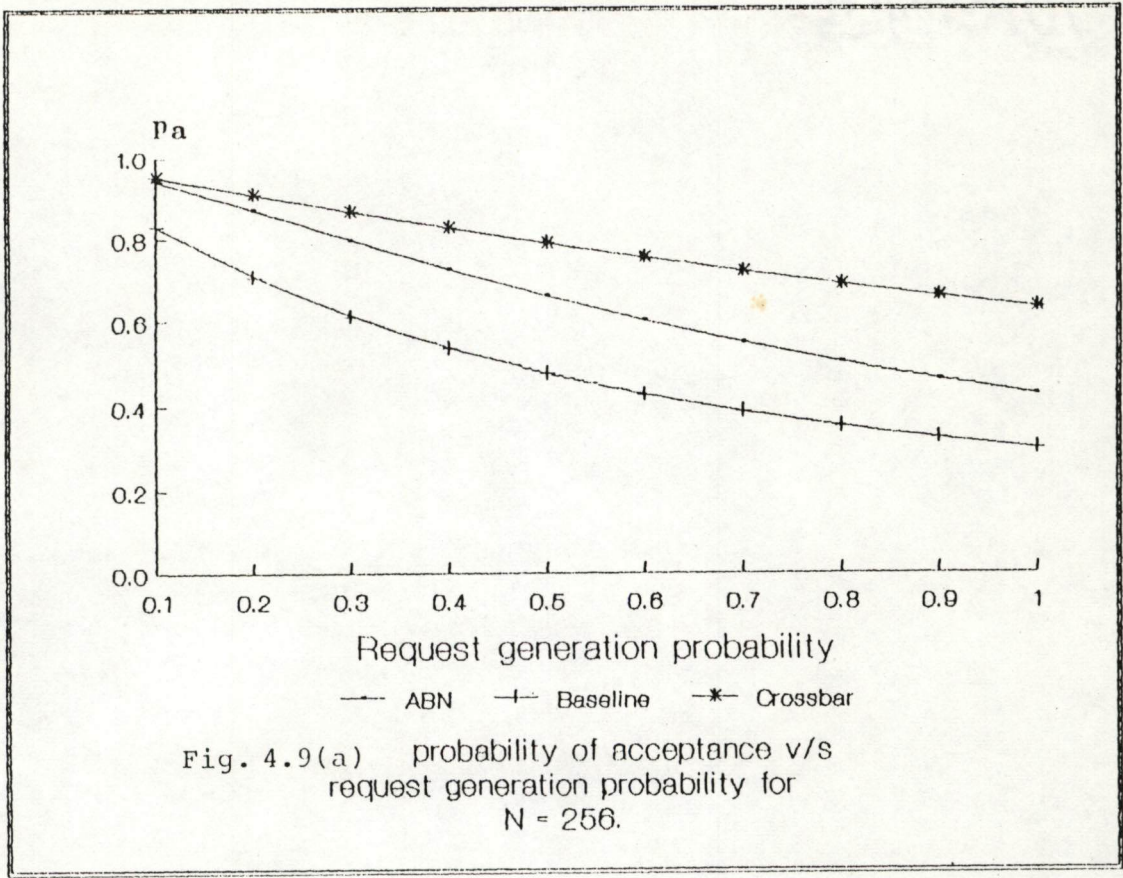


Fig. 4.8(b) probability of acceptance v/s $\log N$ - a request generation probability of 0.5 .



- i) No faults
- ii) a switch fault
- iii) a loop fault

For each category, the effect of fault in each stage of the network is analysed. In all there are ten performance models i.e no faults, five different switch faults and four different loop faults. The results of the performance are summarized in Tables 4.1, 4.2 and 4.3 for request generation probabilities of 1.0, 0.5 and 0.1 respectively. First column of each table indicate the type of fault, II, III, V columns indicate the probability of acceptance, bandwidth and minimum reference probability respectively. Columns IV and VI show the percentage change in the probability of acceptance/bandwidth and minimum reference probability caused by each type of fault. The results show that the degradation in the probability of acceptance and bandwidth under such failures is very small (3%). However, if instead of probability of acceptance and bandwidth, the performance seen by individual outputs are considered, the results are dramatically different.

The performance results show that, though, faults do not significantly reduce the average performance (i.e probability of acceptance and bandwidth), the performance degradation seen by individual memories is large. Faults closer to the memories affect lesser number of memories, but affect them more strongly. Though faults closer to the processors affect greater number of destinations, but affect them mildly. However, the associated processors, though lesser in number, are affected very badly as

Table 4.1

Network Size: 128

Prob. of Req. Generation by any Processor: 1.00

Performance Measure					
Fault type	P_a	Bandwidth	% change in P_a and BW	Minimum Ref. prob.	% change
No Fault	.4520	57.86	-	.4520	-
S1	.4498	57.45	0.7	.4457	1.4
S2	.4481	57.36	0.9	.4364	3.5
S3	.4477	57.31	1.0	.4174	7.7
S4	.4471	57.23	1.1	.3739	17.3
S5	.4472	57.24	1.1	.2980	34.1
L1	.4462	57.11	1.3	.4403	2.6
L2	.4441	56.84	1.8	.4202	7.0
L3	.4409	56.43	2.5	.3627	19.7
L4	.4400	56.32	2.7	.2597	42.5

(Si, Li indicates the switch and loop of stage i)

Table 4.2

Network Size: 128

Prob. of Req. Generation by any Processor: 0.50

Performance Measure					
Fault type	P_a	Bandwidth	% change in P_a and BW	Minimum Ref. prob.	% change
No Fault	.6790	43.46	-	.3395	-
S1	.6749	43.20	0.6	.3354	1.2
S2	.6739	43.13	0.8	.3293	3.0
S3	.6733	43.09	0.8	.3166	6.8
S4	.6728	43.06	0.9	.2899	14.6
S5	.6733	43.09	0.9	.2471	27.2
L1	.6715	42.98	1.1	.3320	2.2
L2	.6690	42.81	1.5	.3194	5.9
L3	.6607	42.28	2.7	.2661	21.6
L4	.6600	42.24	2.8	.1873	44.8

Table 4.3

Network Size: 128

Prob. of Req. Generation by any Processor: 0.10

Performance Measure					
Fault type	P_a	Bandwidth	% change in P_a and BW	Minimum Ref. prob.	% change
No Fault	.9427	12.07	-	.0943	-
S1	.9410	12.04	0.2	.0939	0.4
S2	.9405	12.04	0.2	.0934	0.9
S3	.9397	12.03	0.3	.0918	2.6
S4	.9396	12.03	0.3	.0893	5.3
S5	.9401	12.03	0.3	.0860	8.8
L1	.9398	12.03	0.3	.0937	0.6
L2	.9387	12.01	0.4	.0926	1.7
L3	.9141	11.70	3.0	.0713	24.3
L4	.9140	11.70	3.0	.0483	48.8

the processors find it increasingly difficult to route their requests because of faults. Performance of ABNs for sizes 64x64 and 256x256 are also analyzed and the results are given in appendix 1.

4.8 CONCLUSION

A fault-tolerant scheme has been proposed for a class of regular multistage interconnection networks, named as Augmented baseline network (ABN). In the analysis of ABN, any switch, any multiplexer, and any demultiplexer were assumed to have a possibility to fail. Reliability analysis is presented which allows to have a comparison between the proposed ABN and the previously proposed network having the same fault-tolerant capability. The analysis of the lower and the upper bounds of MTTF showed that ABN performs, in general, more reliably than other fault-tolerant networks. However, if such high reliability comes at the expense of high cost, it may have little value in practice. The analysis on the cost of networks showed that ABN is, in general, more cost-effective than other fault-tolerant MINs. Performance study has also been carried out. The figures show that ABN provides higher probability of acceptance and bandwidth than those of its counterpart unique-path networks of same size and compares favourably with other fault-tolerant networks. The performance of ABN has also been analysed under faulty conditions. It is observed that as the number of faults increases, the performance of ABN decreases. However, for light traffic such performance degradation is not significant. Further,

it has been shown that faults do not significantly affect overall network performance, but the performance seen by individual processors and memories is severely degraded. The worst effect is the increase in network cycle time.

MODIFIED FAULT-TOLERANT DOUBLE TREE (MFDOT) NETWORK

5.1 INTRODUCTION

In a multistage interconnection network, the delay is directly proportional to the path length between a source-destination pair. With a view to reduce the path length to a value lower than any regular fault-tolerant MIN and to reduce the hardware complexity, a new class of statically reroutable irregular fault-tolerant multistage network named as modified fault-tolerant double tree (MFDOT) network is introduced and analyzed in this chapter.

5.2 MODIFIED DOUBLE TREE (MDOT) NETWORK

The double tree (DOT) network [57,58,63,80] discussed in section 2.3, has many desirable properties for use in multiprocessors systems, but lacks fault-tolerant capability.

The DOT network has the following disadvantages:

- . Not even a single switch fault-tolerant.
- . The number of favourite memory modules of a processor are only two irrespective of the size of the network.
- . Less bandwidth.
- . High probability of contention in a path leading to unfavourite memory modules.
- . flip controlled.

In general, for a non-uniform reference model, irregular network gives larger throughput than any regular type of INs because of its smaller path length for favourite processor-memory pair.

Figure 5.1 shows a 8x8 DOT network. The connections between the switches of a DOT network are slightly changed so that the routing from a source to any destination can be performed in a distributed manner using the destination address routing tag. The resulting network shown in Figure 5.2 is named as modified double tree (MDOT) network. The MDOT network is used in the construction of FDOT, MFDOT and QT networks which are developed in this thesis.

A switch can fail in two ways, it can either be struck-at-through (s-a-t) or struck-at-cross (s-a-x). The following theorems characterize the fault-tolerant and full access capability of MDOT network.

Theorem 1: MDOT network is zero fault-tolerant.

Proof: The center switch C_S in MDOT network is critical. If C_S is s-a-t, then the MDOT network will be split into disjoint upper and lower halves. Hence the fault-tolerance of MDOT is zero.

Q.E.D.

Theorem 2: MDOT network has full access.

Proof: Every input terminal of MDOT network can reach the center switch C_S , and C_S can reach every output terminal. Hence MDOT network has full access.

Q.E.D.

A modification of MDOT network is now proposed to make it fault-tolerant.

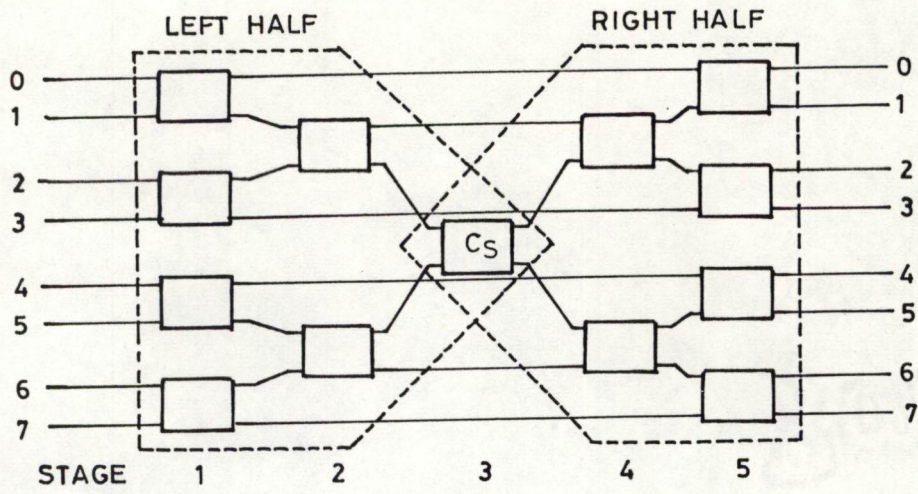


FIG. 5.1 THE 8 x 8 DOT NETWORK

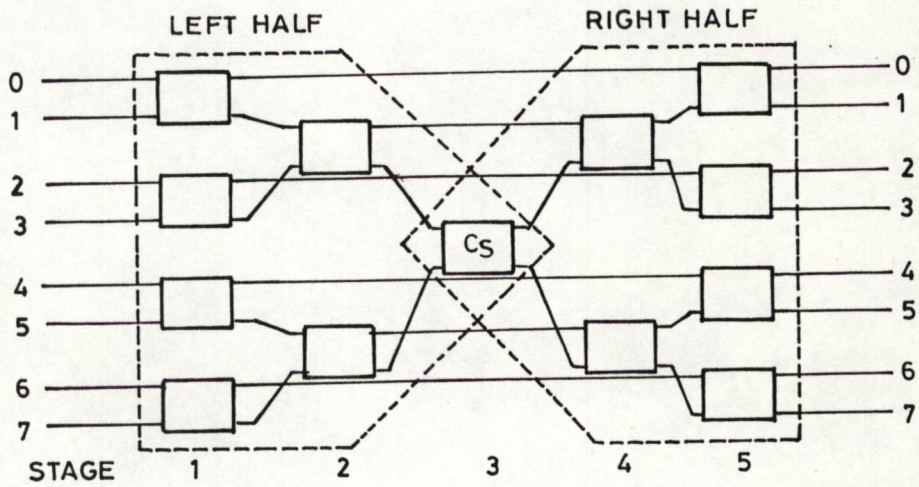


FIG. 5.2 THE 8x8 MDOT NETWORK

5.3 FAULT-TOLERANT DOUBLE TREE (FDOT) NETWORK

A fault-tolerant double tree network, FDOT-k of size $N \times N$ is designed by dividing N sources and N destinations into k disjoint partitions of N/k sources and N/k destinations where, $k (\geq 2)$ and $N (> k)$ are the powers of 2. There are k independent subnetworks, and an extra one, such that a connection path between each source-destination pair can be established via any one of the subnetwork. All the $(k + 1)$ subnetworks are of identical type and consists of a MDOT network of size $N/k \times N/k$. The extra subnetwork helps to enhance fault-tolerant capability and to keep a desired level of performance even in the presence of faults. There are no particular sources and destinations associated with it, but it is for added redundancy. Each source and each destination are connected to all the $(k + 1)$ subnetworks via $k \times 1$ multiplexers and $1 \times k$ demultiplexers.

Thus an FDOT-k consists of $(2n - 1)$ number of stages and $(k + 1) (2^{n+1} - 3)$ number of switches, where $n = \log_2 N/k$. Further, it consists of $N \times (1 + 1/k)$ number of $k \times 1$ multiplexers and an equal number of $1 \times k$ demultiplexers. Each of the stage i and $2(n - 1) - i$ have $(k + 1) \times 2^{n-i-1}$ switches for $i = 0, 1, \dots, (n - 1)$. Each MDOT network of size $N/k \times N/k$ plus its associated multiplexers and demultiplexers constitute a module. These modules are denoted by $M_0, M_1 \dots M_k$. Any module can be assumed to be extra one, but for the sake of convenience M_k is treated as extra one.

Let $S_{i,j}$ and $D_{i,j}$ denote the source j and destination j respectively, which are associated with module M_i based on the

partition $0 \leq i \leq (k - 1)$ and $0 \leq j \leq (N/k - 1)$. Also, let $MUX_{l,m}$ and $DEMUX_{l,m}$ represent the MUX m and DEMUX m in M_l module respectively, where $0 \leq l \leq k$ and $0 \leq m \leq (N/k - 1)$.

The sources and destinations are connected as follows:

- i) Each $S_{i,j}$ is connected to every i th input port of the $(k + 1)$ multiplexers from $MUX_{0,j}$ to $MUX_{k,j}$.
- ii) Every i th output port of the $(k + 1)$ demultiplexers from $DEMUX_{0,j}$ to $DEMUX_{k,j}$ is connected to each $D_{i,j}$.

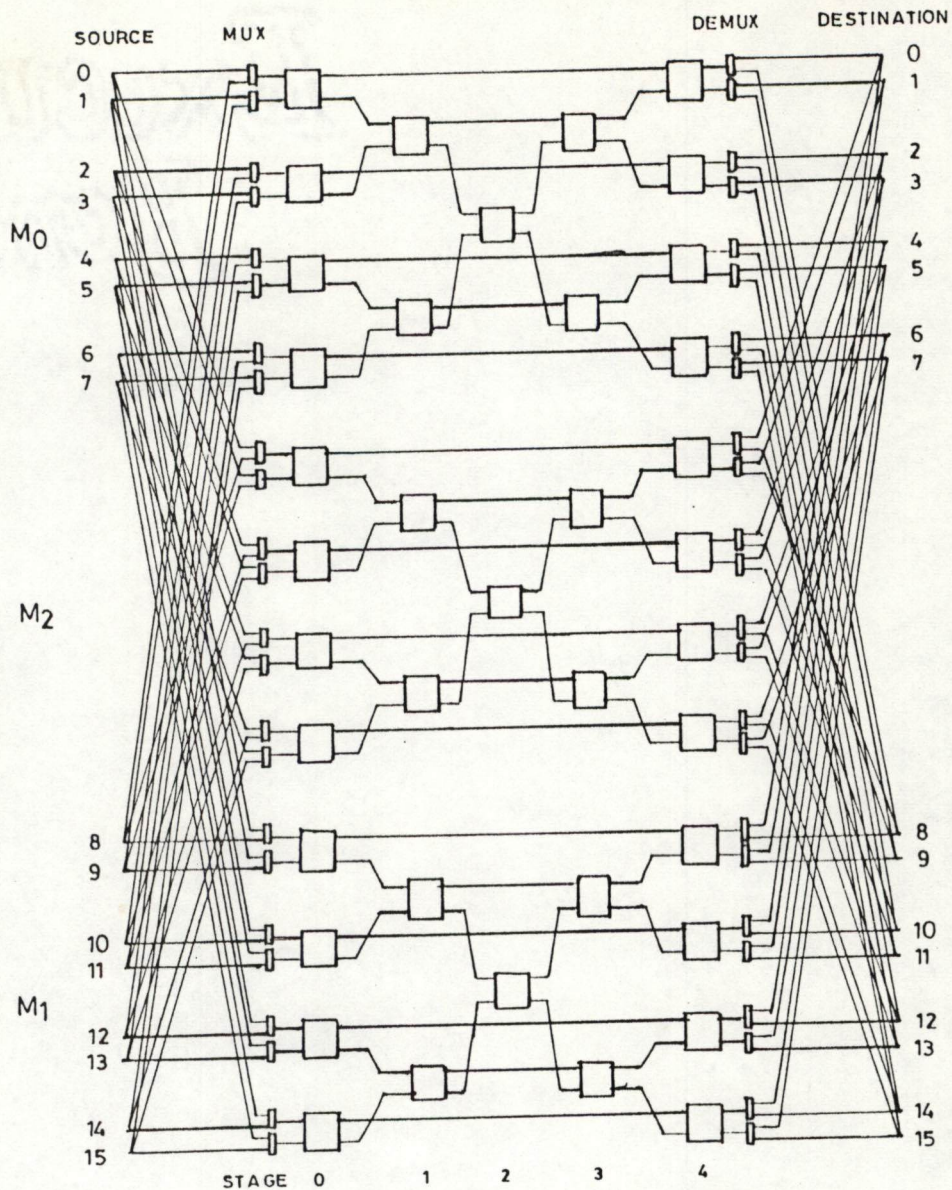
A network resulting from this construction is called FDOT- k . Example of FDOT-2 network of size 16×16 along with its redundancy graph is illustrated in Figure 5.3.

A further modification of FDOT network is proposed to reduce the communication delay.

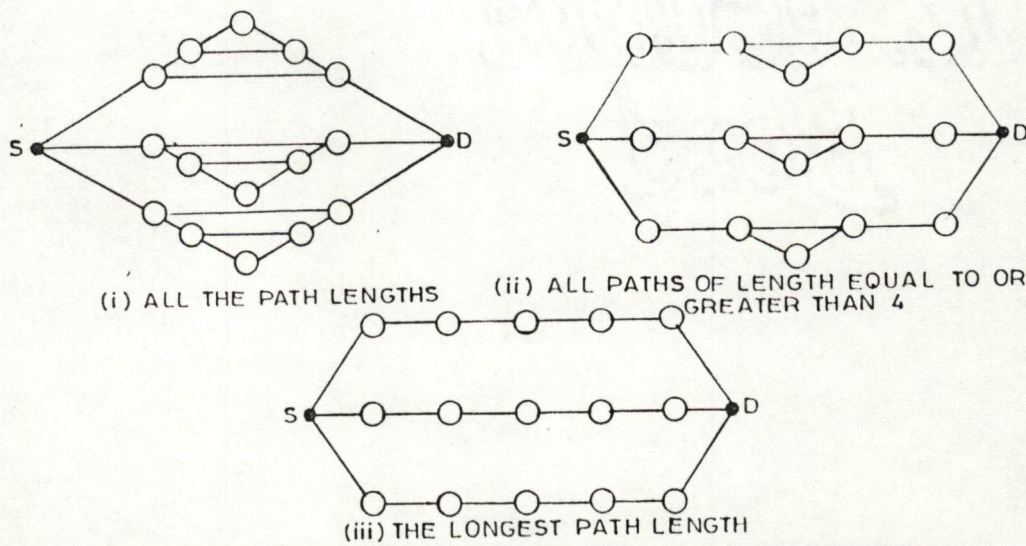
5.4 MODIFIED FAULT-TOLERANT DOUBLE TREE (MFDOT) NETWORK

It is easily seen that the central switch in each subnetwork of FDOT is redundant. Thus, the central switch in each subnetwork is replaced with a X connection. The resulting network is named as modified fault-tolerant double tree (MFDOT) network. Like the FDOT, the MFDOT network also possesses full access capability. However, the number of stages, and hence the communication delay has been reduced. The number of switches have also been reduced to $(k + 1) (2^{n+1} - 4)$. With the reduction in the number of stages, MFDOT network displays better delay characteristics compared to DOT and FDOT networks.

An example of MFDOT-2 of size 16×16 along with its redundancy graph is illustrated in Figure 5.4.



(a) AN 16x16 FDOT-2 NETWORK



(b) REDUNDANCY GRAPHS FOR S-D TERMINAL PAIR CONTAINING
FIG. 5.3

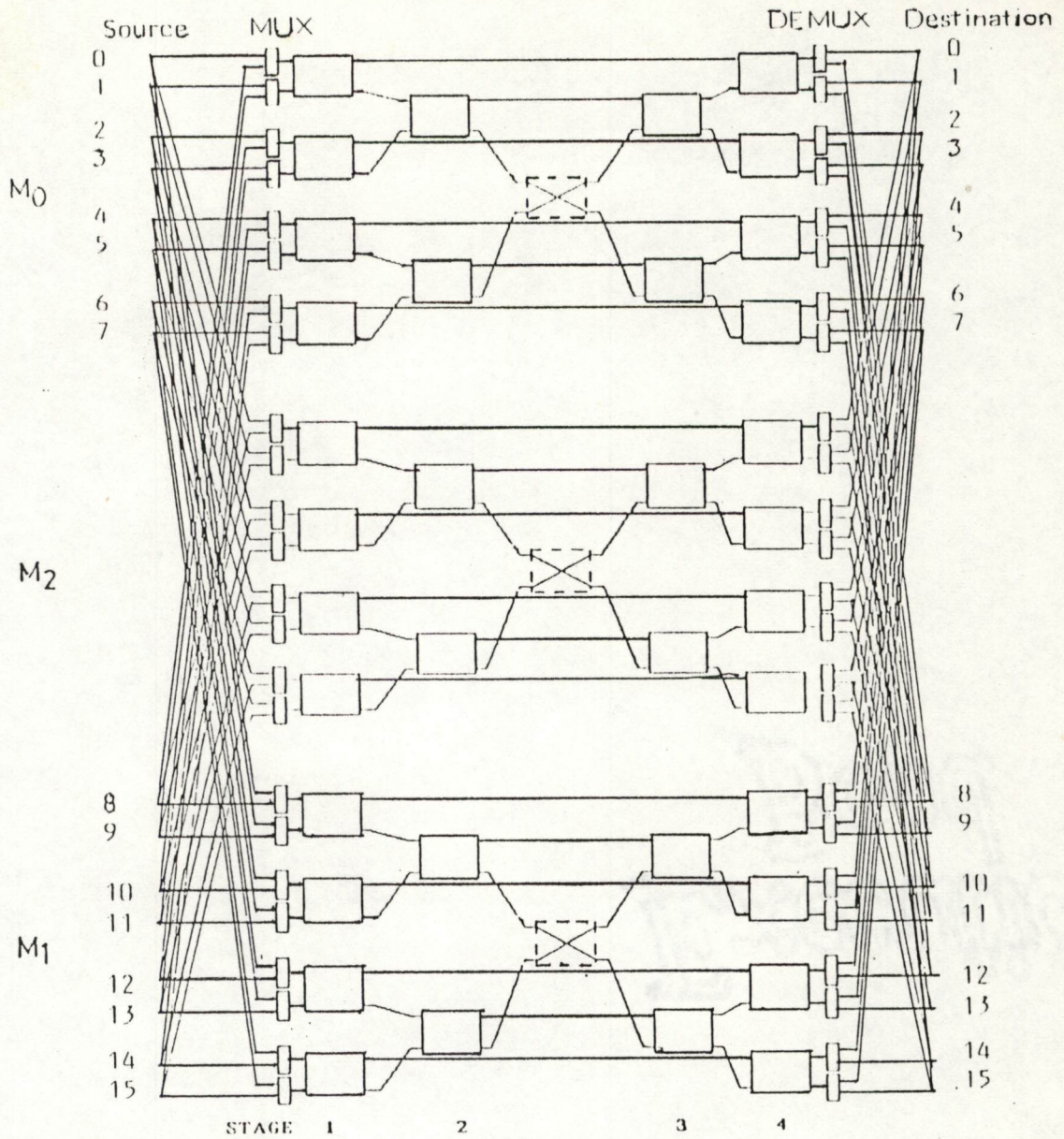


FIG. 5.4(a) AN 16 x 16 MFDOT-2 NETWORK

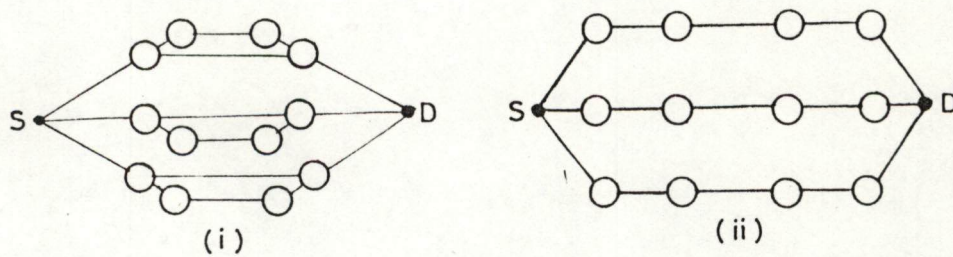


FIG. 5.4(b) REDUNDANCY GRAPHS FOR S-D TERMINAL PAIR CONTAINING
 (i) ALL THE PATH LENGTH
 (ii) THE LONGEST PATH LENGTH

5.5 ROUTING SCHEME OF MFDOT NETWORK

In the following section, the routing scheme of MFDOTs is described.

5.5.1 Path Length Algorithm

For a given source-destination pair, there are multiple paths of varying path lengths in an MFDOT-k. The number of possible paths between a source-destination pair varies from $(k + 1)$ to $(k + 1) \cdot (n - 1)$, depending upon the addresses of the source and destination terminals. Path length algorithm gives the information about the various possible paths between a source-destination pair.

Let the source S and destination D be represented in binary code as:

$$S = S_{i,j} = s_n , (s_{n-1} \cdot \cdot \cdot \cdot s_0)$$

$$D = D_{i,j} = d_n , (d_{n-1} \cdot \cdot \cdot \cdot d_0)$$

In any module the path length algorithm is:

if

$$[(s_{n-1} \oplus d_{n-1}) + (s_{n-2} \oplus d_{n-2}) + \dots + (s_1 \oplus d_1)] \text{ is zero}$$

(\oplus Represents an exclusive - OR operation and

+ represents a logical OR operation)

then

minimum path length is 2 and all paths of various lengths are possible i.e $2, 4, \dots, 2n-2$.

else

if

$[(s_{n-1} \oplus d_{n-1}) + (s_{n-2} \oplus d_{n-2}) + \dots + (s_2 \oplus d_2)]$ is zero

then

all paths of length equal to or greater than 4 are possible.

else

.
. .
.

if

$[(s_{n-1} \oplus d_{n-1}) + (s_{n-2} \oplus d_{n-2}) + \dots + (s_j \oplus d_j)]$ is zero

(where $1 \leq j \leq n-1$)

then

all paths of length equal to or greater than $2j$ are possible.

else

path of length $2n-2$ (i.e longest path) is possible only.

5.5.2 Routing Tag Algorithm

Routing tag algorithm gives the information about the distributed routing control required to establish a path between any source-destination terminal pair for a given path length.

The required algorithm is given as:

if

$2 \leq x \leq 2n-2$

(where x is the path length)

then

routing tag=

$(1.1\dots 1) (\lfloor x/2 \rfloor - 1) \cdot 0 \cdot (d_{(\lfloor x/2 \rfloor - 1)} \dots d_0) \cdot d_n$

else

if

$$x = 2n-2$$

then

routing tag=

$$(1.1\dots1)_{(n-2)} \cdot (d_{n-1}\dots d_0) \cdot d_n$$

Example

The functioning of routing tag algorithm is illustrated with examples for an MFDOT-2 of size 16x16.

Case 1: Let the data be routed from source 0 to destination 1.

The source S and destination D are represented as:

$$S = 0, 000 \quad D = 0, 001$$

$$\text{then } n = \log_2 N/k = 3$$

Therefore, in this case path of lengths 2 and 4 are possible i.e $x = 2$ and 4.

So for $x = 2$ routing tag = 0.1.0 and

for $x = 4$ routing tag = 1.0.0.1.0

Case 2: Let the data be routed from source 0 to destination 15.

The source S and destination D are represented as:

$$S = 0, 000 \quad D = 1, 111$$

Therefore, in this case longest path of path length 4 is only possible i.e $x = 4$.

for $x = 4$ the routing tag = 1.1.1.1.1

As can be seen from Figure 5.4, these tags give the desired destinations.

5.5.3 Routing Procedure

One way of selecting a module for routing in MFDOT, is to use a "preferred module" for a path between a source-destination pair, which is the path always chosen first by the source. For example in Figure 5.4, if a source needs to communicate with a destination $D_{i,j}$, the preferred module can be M_i i.e the source will sent a request to the corresponding MUX in M_i . If the preferred module is faulty (it is assumed that some techniques of fault diagnosis are available to detect and locate faults in the network [2]), the source then may try a path in a module other than M_i . If that module is faulty too, the request will be submitted to an alternative module untill all the $(k + 1)$ modules are faulty. In this way one can maintain a "desirable" performance level.

Another possible way is the "uniform random" selection of a module out of the $(k + 1)$ available modules, selecting a module randomly with equal probability. As one of the module becomes faulty, the number of modules a source can utilize decreases. In this way, one can achieve a graceful degradation of the performance in case of faults.

Assuming that the selection of a module out of the $(k + 1)$ modules is already done at source, first find the minimum path length possible between a source-destination pair and then the corresponding routing tag. No tag bit is consumed at the multiplexer stage, because a multiplexer has only one output. After the multiplexer, the routing in the intermediate stages of a module depends upon routing tag. With the help of routing tag

and the routing procedure described above, a path is established to the proper destination.

5.6 FAULT-TOLERANCE OF MFDOT NETWORK

Fault-tolerance of MFDOT network can be analyzed by defining a critical set of components. A critical set of SEs in an MFDOT-k is defined as the set of $k+1$ SEs, each from different modules and having the same position, such that a failure will occur if all the $k+1$ SEs become faulty simultaneously.

The following theorem characterizes certain minimum faults that are always tolerated in an MFDOT-k.

Theorem 3: The fault-tolerance of MFDOT-k is k .

Proof: In the redundancy graph of MFDOT-k, there are $k+1$ paths available between each source-destination pair. If the faults are such that they affect k or fewer modules, there exists at least one fault free module. Thus the fault-tolerance is k .

Q.E.D.

Theorem 4: MFDOT-k has full access.

Proof: Since an MFDOT-k provides at least $(k+1)$ disjoint paths for each source-destination pair, through the $(k+1)$ identical modules, and if there are k or fewer faulty modules, then at least one of the module is still fault free. Since every module of MFDOT-k has full access, so MFDOT-k has full access.

Q.E.D.

However, $(k + 1)$ faults can cause a failure in an MFDOT-k. For instance, if $(k + 1)$ faulty components are distributed in $(k + 1)$ modules and the position of each faulty component in all modules is the same, then some sources and destinations become

disconnected from the rest of the network. Some instances of even more than k faults can be tolerated; for example, if the faulty switches are confined to k or less modules, then the system will still work properly.

A network is robust in the presence of k faults, if it can tolerate some instances of k faults. An MFDOT- k is robust in the presence of more than k faults. The maximum number of faults it can tolerate comes from the case that only one of the module is fault free. Thus, an MFDOT- k is robust upto $(n.N/2 + 2N)$ faults in the network.

5.7 PERMUTATION CAPABILITY OF MFDOT NETWORK

Permutation is an important criterion of any MIN. For an MFDOTs, the goal is to maintain permutation capability in the presence of any single fault. This goal is achieved by the extra module provided in an MFDOT. It is clear that the extra one can be used to replace the faulty module without increasing conflicts. So an MFDOT- k can maintain the permutation capability as long as there are k fault free modules.

5.8 RELIABILITY AND REPAIRABILITY OF MFDOT NETWORK

Since MFDOT- k is k fault-tolerant so it is quite reliable as compared to other networks containing nearly same amount of hardware. The added reliability comes at the cost of the general performance of the network, because basically a stage of switches is replaced by multiplexers and demultiplexers, which are less complex. Due to the presence of k modules, each module connects

every source to every destination, so even a single module will have the full access capability. If some faults develops in any module, then the faulty module can be simply replaced by the new one without causing a breakdown in the system. Thus, the system is on-line repairable.

5.9 COMMUNICATION DELAY IN MFDOT NETWORK

Even though parallel processing has increased computational speed, still there is scope for increasing it further by reducing the communication delay in an IN. In a module of MFDOT, each source is connected to two "favourite" destinations with the minimum path length i.e 2. Thus, the total number of output favourite memory modules/processors to which a input processor can communicate with the shortest possible path length is $2k$ in an MFDOT- k . The reduced path length results in shorter communication delay and hence increased speed.

5.10 CONCLUSION

In this chapter, a new class of statically reroutable irregular fault-tolerant multistage interconnection network named modified fault-tolerant double tree (MFDOT) network has been proposed and analyzed. The network achieves significant tolerance to faults and good performance with relatively low cost and simple control scheme. MFDOT- k is k fault-tolerant and robust in the presence of more than k faults. Moreover, due to the presence of extra subnetwork, MFDOT can maintain permutation capability when any single fault occurs. Algorithms for path length and

routing have been developed. In short, MFDOT-k network offers the following advantages:

- . k switch fault-tolerant.
- . On-line repairability of faulty components.
- . Significantly improved reliability and performance.
- . Multiple paths of varying lengths are available between each source-destination pair.
- . For a connection to a favourite memory module, the path length is only 2 irrespective of the size of the network.
- . The number of favourite memory modules is increased from two to 2k compared to a DOT network of similar size.
- . Cost-effective.

CHAPTER 6

QUAD TREE (QT) NETWORK

6.1 INTRODUCTION

In this chapter, a new class of dynamically reroutable irregular multistage interconnection network named quad tree (QT) network, that provides multiple paths of varying lengths between a source-destination pair, is proposed and analyzed. QT network is obtained by chaining switches within a stage so that data can sidestep a faulty switch. QT network can provide 'full access' capability in the presence of multiple faults and is cost-effective compared to other fault-tolerant MINs with a similar fault-tolerant capability. Rerouting in the presence of faults can be accomplished dynamically without resorting to backtracking. Algorithm for path lengths and routing are developed for QT network.

6.2 CONSTRUCTION

Quad tree (QT) network of size $N \times N$ is constructed with the help of two identical groups $G^{S^{n-1}}$, each consisting of MDOT network of size $N/2 \times N/2$, which are arranged one above the other ($n = \log_2 N$). The two groups are formed based on the most significant bit (MSB) of the source/destination terminals. Thus one half of the source/destination terminals with a MSB 0 falls into G^0 group and the others having MSB as 1 falls into G^1 . Each of the stage i and $2m - i$ in a group have $2^n - 1 - i$

switches and are numbered as $1, 2, \dots, 2^{n-i} - 1$ (where $i = 1, 2, \dots, m$, $m = \log_2 N/2$). The switches in a stage having the same number in both the groups, forms a loop. Such loops of switches are formed in all the stages except the last. Each source and destination are connected to both the groups with the help of multiplexers and demultiplexers.

Let the source S and destination D be represented in binary code as:

$$S = s_{n-1} \cdot \cdot \cdot \cdot s_1 s_0$$

$$D = d_{n-1} \cdot \cdot \cdot \cdot d_1 d_0$$

The sources and destinations are connected to the multiplexers and demultiplexers as follows:

- i) If $(s_{n-2} \cdot \cdot \cdot \cdot s_1 s_0)$ bits are the same for the two sources, then these two sources are linked through the same pair of multiplexers.
- ii) If $(d_{n-2} \cdot \cdot \cdot \cdot d_1 d_0)$ bits are the same for the two destinations, then these two destinations are linked together through the same pair of demultiplexers. together through the same pair of demultiplexers.

Thus, a QT network of size $N \times N$ consists of $(2m - 1)$ number of stages, $(2^m + 2 - 6)$ number of switches - out of which 2^{n-1} are of size 2×2 and the rest are of size 3×3 . There are N number of 2×1 multiplexers and an equal number of 1×2 demultiplexers. Each of the stage i and $(2m - i)$ have 2^{n-i} switches.

A network resulting from this construction is called quad tree (QT) network - each quarter of the network resembles a

binary tree. An example of QT network of size 16x16 along with its redundancy graph is illustrated in Figure 6.1.

6.3 ROUTING SCHEME

The following section describes the routing scheme of QT network:

6.3.1 Path Length Algorithm

For a given source-destination pair, there are multiple paths of different path lengths in a QT network. The possible path lengths between a particular source-destination pair varies from 2 to $2m - 1$ for a $N \times N$ network, depending upon the addresses of the source and destination terminals.

The path length algorithm is:

if

$[(s_{n-2} \oplus d_{n-2}) + (s_{n-3} \oplus d_{n-3}) + \dots + (s_1 \oplus d_1)]$ is zero

then

minimum path length is 2 and all paths of different lengths are possible i.e paths of length 2, 4, 6, . . . , $(2m - 2), (2m - 1)$.

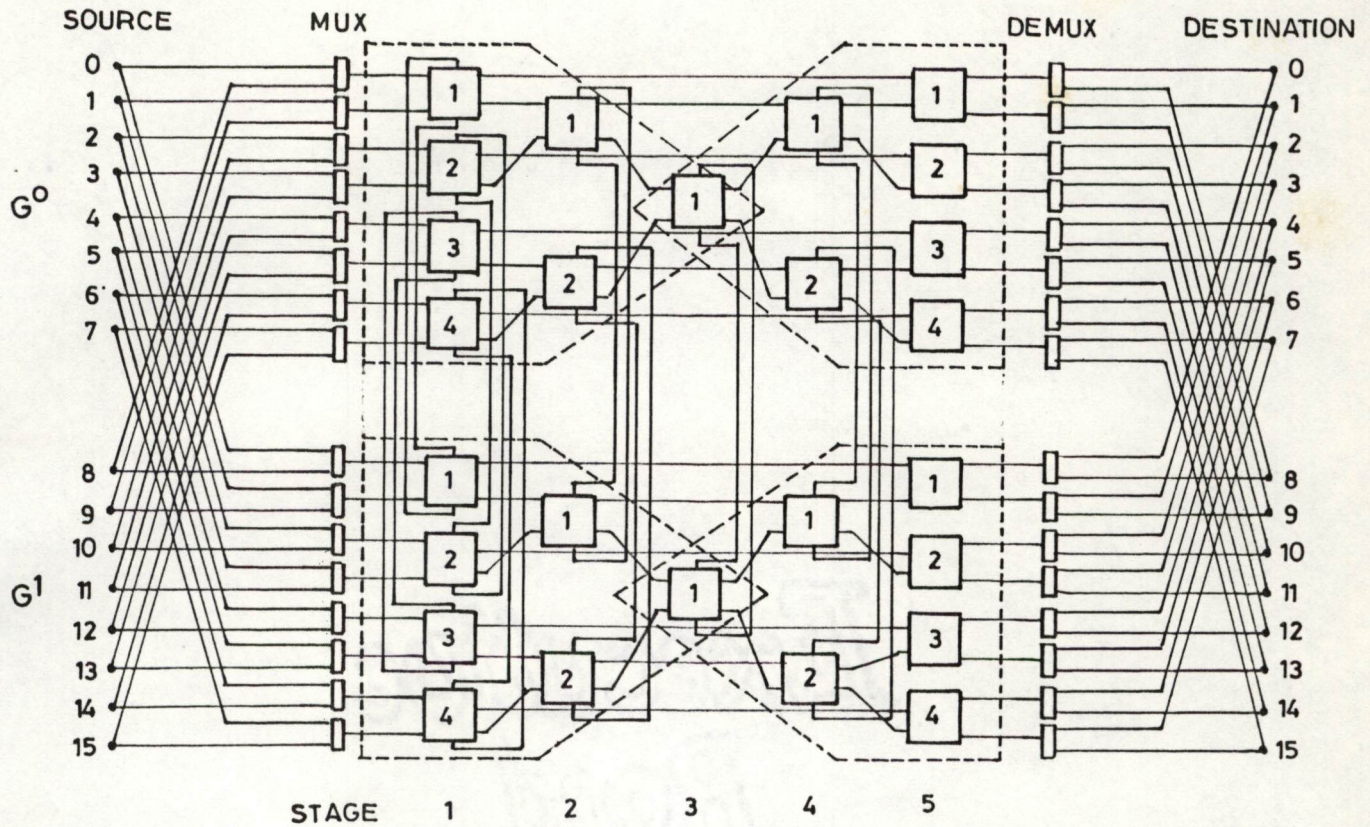
else

if

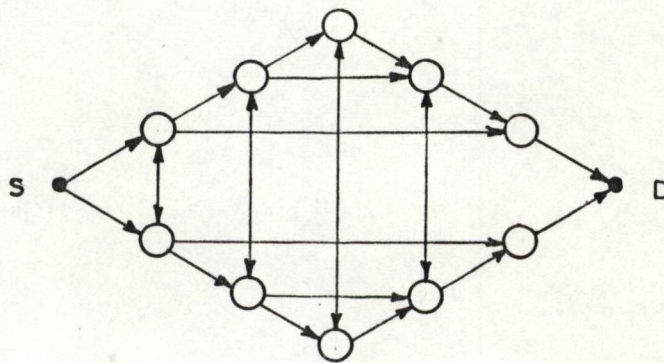
$[(s_{n-2} \oplus d_{n-2}) + (s_{n-3} \oplus d_{n-3}) + \dots + (s_2 \oplus d_2)]$ is zero

then

all paths of length equal to or greater than 4 are possible.



(a)



(b)

FIG. 6.1 (a) QT NETWORK OF SIZE 16 X 16
(b) ITS REDUNDANCY GRAPH

else

·
·
·

if

$[(s_{n-2} \oplus d_{n-2}) + (s_{n-3} \oplus d_{n-3}) + \dots + (s_j \oplus d_j)]$ is zero

{ where $1 \leq j \leq (n - 2)$ }

then

all paths of length equal to or greater than $2j$ are possible.

else

path of length $2m - 1$ (i.e longest path) is possible only.

The path length algorithm is explained here with an example.

Example

Let the data be routed from $S = 0000$ to the various destinations of a 16×16 QT network. The path lengths are calculated for sets of destinations and are summarized in table 6.1.

6.3.2 Routing Tag Algorithm

Routing tag algorithm for QT network gives the information about the distributed routing control required to establish a path between any source-destination terminal pair for a given path length.

If

$$2 \leq x < (2m - 1)$$

(where x is the path length)

Table 6.1

S	D	Path length(s) available (x)
0000	0000	
	0001	2,4,5
	1000	
	1001	
	0010	
	0011	4,5
	1010	
	1011	
	0100	
	0101	
	0110	
	0111	5
	1100	
	1101	
	1110	
	1111	

then

routing tag =

$(1.1 \dots .1) (\lfloor x/2 \rfloor - 1) .0. (d_{\lfloor x/2 \rfloor} \dots d_0) .d_{n-1}$

else

if

$x = (2m - 1)$

then

routing tag =

$(1.1 \dots .1) \lfloor x/2 \rfloor (d_{\lfloor x/2 \rfloor} \dots d_0) .d_{n-1}$

The functioning of routing algorithm is illustrated with examples for an QT network of size 16x16.

Example

Case 1: Let the data be routed from source 0 to destination 1.

The source S and destination D are represented as:

S = 0000 D = 0001

then $m = \log_2 N/2 = 3$ and $n = 4$

Therefore, in this case path of lengths 2, 4 and 5 are possible i.e $x = 2, 4$ and 5.

So for $x = 2$ routing tag = 0.1.0

for $x = 4$ routing tag = 1.0.0.1.0 and

for $x = 5$ routing tag = 1.1.0.0.1.0

Case 2: Let the data be routed from source 0 to destination 15.

The source S and destination D are represented as:

S = 0000 D = 1111

Therefore, in this case longest path of path length 5 is only possible i.e $x = 5$.

for $x = 5$ the routing tag = 1.1.1.1.1.1

As can be seen from Figure 6.1, these tags give the desired destinations. Redundancy graphs for the above examples are given in Figures 6.2(a) and 6.2(b).

6.3.3 Routing Procedure

The reliability and performance improvement obtained from a multiple path network depends upon how effectively the alternate paths available are used by the routing algorithm. One can use a backtracking routing algorithm that extensively searches for an available fault-free path. However, implementation of backtracking is relatively expensive in terms of hardware and backtracking can, in some situations, take an inordinately long time to set up connections. The algorithm given in this section is easy to implement and performs quite well. This algorithm assumes that sources and switches have the ability to detect faults in the switches to which they are connected. Faults in MINs can be detected by the application of test inputs [3,31], or by employing concurrent error-detection at the network level or at the switch level [33].

For any source-destination pair, find the minimum path length possible and then the corresponding routing tag. The request is routed as follows:

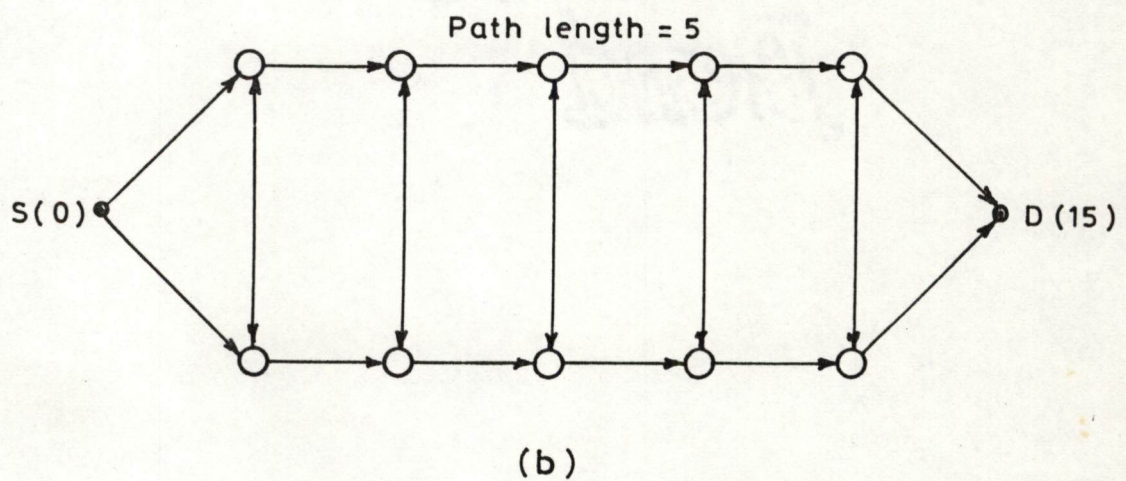
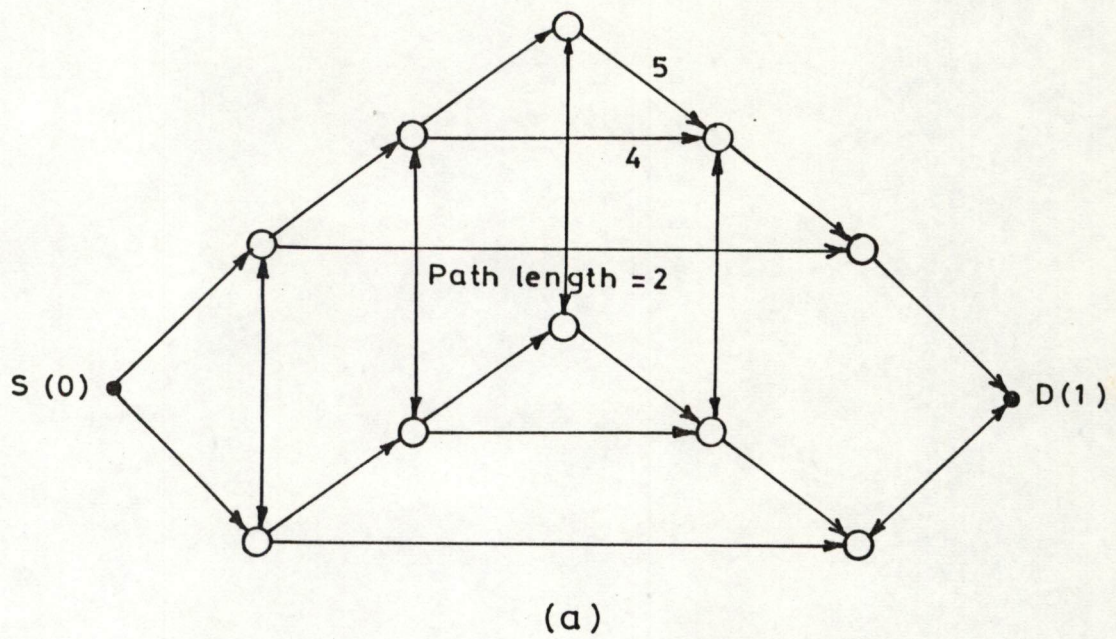


FIG. 6.2 REDUNDANCY GRAPH FOR
 (a) $S = 0$ TO $D = 1$
 (b) $S = 0$ TO $D = 15$

Submit the request for connection to the primary path. If the request is routed through the same group to which the source belongs (i.e MSB of the source = group number), then the path is called the primary path otherwise the path is secondary. If the primary path is faulty (i.e multiplexer or the switch or both are faulty) then submit the request to the secondary path. If the secondary path is also faulty then drop the request. For each switch in stage i ($i < 2m - 1$): request for connection may arrive at any of the three input links. For each request, use the appropriate routing tag bit and connect to the output link labeled correspondingly. If the required output link is busy or cannot be used because of a fault in the next stage, route the request via the third output link known as auxiliary link to the other switch in the loop. If the auxiliary link is also unusable, because it is busy or because of a fault, then drop the request. A fault in the demultiplexer at the output of a switch in a stage $(2m - 1)$ is regarded as a fault in that switch. From the demultiplexer, the request is routed to the upper or lower destinations according to the least significant bit of the tag (i.e $d_n - 1$).

6.4 FAULT-TOLERANCE

The routing algorithm for QT network allows two ways of routing a connection request in every stage except the final i.e any given source-destination pair have a fork available at every stage. If the destinations are assumed to be fault free it allows that all single switch faults can be tolerated if the routing

algorithm described in the previous section is used. QT networks are strictly speaking single switch fault-tolerant. If both switches in a loop are simultaneously faulty, then clearly some sources are disconnected from some destinations. However, if such critical combination of switches are not present in a fault pattern, several multiple faults (upto half the total number of switching elements in the network) can be tolerated.

The following theorems characterize the faults that are tolerated in QT network. Here switch i in stage j is denoted by $S_{i,j}$ and its associated switch in the loop is denoted by $asc(S_{i,j})$.

Theorem 1: In the QT network, if faults occur such that at most one switch is affected in every pair of switches in a loop, and at most one switch is affected out of the pair of switches having the same switch number in the final stage, then there exists at least one fault-free path from any source to any destination.

Proof: Since the switch faults are confined to at most one switch in every loop, either $S_{i,j}$ or $asc(S_{i,j})$ is guaranteed to be fault-free for $1 \leq j \leq (2m - 2)$. It is easy to see from the redundancy graph that a request from S to D can be routed through fault free switches up to the switch $S_{i,2m - 2}$ or $asc(S_{i,2m - 2})$, whichever is fault-free. From stage $2m - 2$ to D , a path exists if either of the same numbered switch is fault-free. Since at most one of these two switches is assumed to be faulty, a path exists from S to $S_{i,2m - 2}$ (or to $asc(S_{i,2m - 2})$) and from $S_{i,2m - 2}$ (or $asc(S_{i,2m - 2})$) to D .

Q.E.D

Theorem 2: In QT network, some of the source-destination pairs become disconnected, if the two switches bearing the same number in a stage are faulty at the same time.

Proof: Suppose that the path from a source S to a destination D contains a faulty switch $S_{i,j}$. Then all the available paths from S to D must pass through either of the two same numbered switches in a stage. Thus if both the switches having the same number in a stage are faulty at the same time, then S is disconnected from destination D.

Q.E.D.

6.5 COST-ANALYSIS AND COMPARISON WITH OTHER NETWORKS

In this section, the hardware cost of MFDOT-2 and QT networks are computed and compared with some of the known redundant path regular interconnection networks. The networks used for comparison are ESC, 3-replicated, INDRA with $R = 2$ and ASEN - 2. The cost function of these networks are calculated as per the assumptions used in section 3.7. So ESC, 3-replicated, INDRA, ASEN - 2, MFDOT-2, and QT networks have costs of $2N(\log_2 N + 5)$, $6N\log_2 N$, $4N(\log_2 N + 1)$, $3N(1.5\log_2 N - 1)$, $(k + 1)(2^n + 3 + 2N - 16)$ and $(9.75 * 2^n + 1 - 54)$ respectively.

Graph shown in Figure 6.3 illustrates the variation of cost function with the size of the networks (2x2/3x3 switches). It is clear that irregular MFDOT and QT networks are more cost-effective than other fault-tolerant multistage interconnection networks and that such advantage becomes more profound as the network size becomes larger.

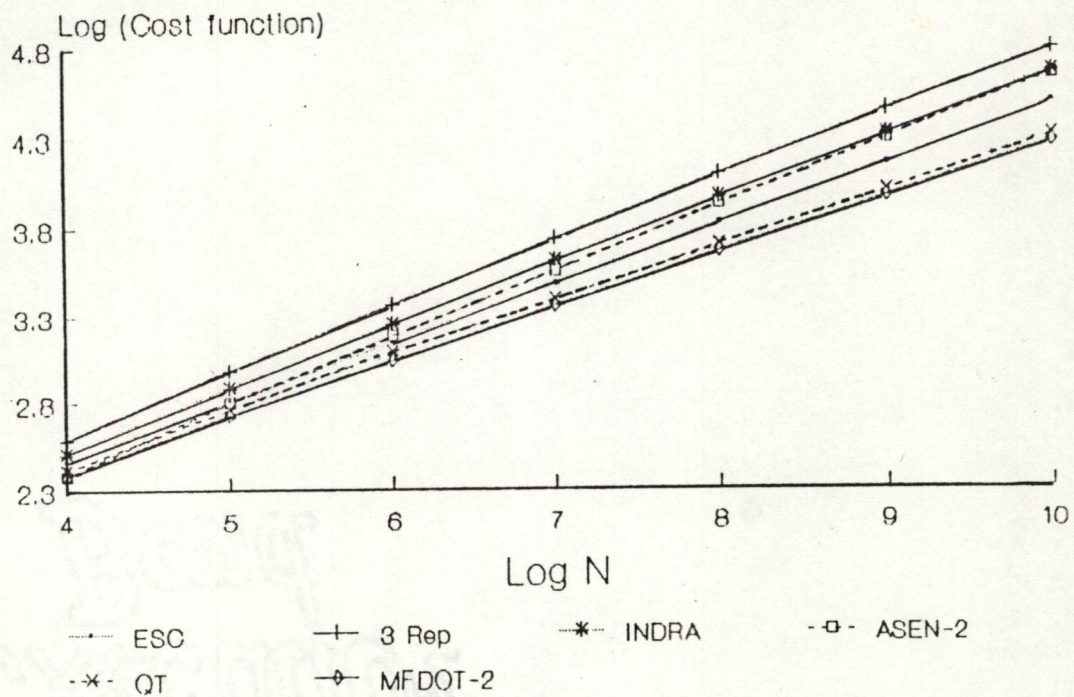


Figure 6.3 Cost function v/s Network Size.

6.6 CONCLUSION

A new class of dynamically reroutable irregular fault-tolerant multistage interconnection network, named quad tree (QT) network, has been proposed and analyzed. A routing scheme is described for communication in a multiprocessor system employing dynamically reroutable QT network. The scheme avoids faulty switching elements by routing the data through an auxiliary link. The QT network results in the following advantages:

- . Single switch fault-tolerant.
- . Alternate routing without backtracking in all stages.
- . Significantly improved reliability and performance.
- . Multiple paths of varying lengths are available between each source-destination pair.
- . For a connection to a favourite memory module, the path length is only 2 irrespective of the size of the network.

CHAPTER 7

CONCLUSIONS

This chapter finally sums up the conclusions that have been arrived at during the design and analysis of the proposed fault-tolerant multistage interconnection networks. Suggestions for future work have also been incorporated.

7.1 CONCLUSIONS

To improve the performance and reliability of the existing statically reroutable, fault-tolerant regular MINs, a modular network (MN) has been proposed. The characteristics of MN pertaining to reliability, cost-effectiveness and performance have been analyzed. The analyses showed that MNs are more reliable and are cost-effective than most other statically reroutable fault-tolerant regular MINs. The performance analysis for both fault free and fault present conditions has also been carried out. It has been shown that MN provides higher probability of acceptance than unique-path MINs. Obviously as the faults increase the performance decreases. However, the performance degradation is not significant for small size MNs and for light traffic.

Conventional performance analysis (under faulty conditions) available in the literature does not give a complete picture of the system capability. In order to critically examine the network a parameter known as minimum reference probability has been used

for the analysis of MNs alongwith the global parameters such as probability of acceptance and bandwidth etc. the results showed that though faults do not significantly reduce the average performance, the degradation seen by individual outputs is large. Faults closer to the memories affect lesser number of memories but affect them more strongly. This unsymmetrical degradation is very harmful, as it increases the network cycle time. This points out the need for load balancing and dynamic scheduling.

With a view to reduce the number of stages and hence increase the computational speed, a fault-tolerant regular MIN named augmented baseline network (ABN), which is dynamically reroutable, has been designed. The results of the analyses showed that ABN perform, in general, more reliably than other regular MINs having similar fault-tolerant capabilities. Performance study has been carried out: due to the availability of multiple paths, an ABN provides higher probability of acceptance and bandwidth than those of its counterpart unique-path networks and compares favourably with other regular multipath networks.

The effect of component failures on the performance of ABN has also been studied. Based on the results, it has been found that degradation in probability of acceptance and bandwidth is very small, 0(3%). However, if instead of these measures, the performance seen by individual inputs and outputs are considered, the results are dramatically different. Like MN, it has been observed that faults closer to the outputs affect lesser number of memories, but affect them more strongly. Further, faults closer to the inputs affect more number of outputs, but affect

them mildly; the associated processors though lesser in number are affected very badly, as these processors find it increasingly difficult to route their requests because of faults. However, the performance degradation is not significant for small size ABNs. In short, higher reliability and performance for small size ABNs, coupled with ease of maintenance and repair makes ABN attractive for use in multiprocessor systems.

The regular MINs proposed in the literature have path length $O(\log N)$, which puts a limitation on the computational speed of an interconnection network. With a view to reduce the path length to a value lower than any regular MIN, a new class of irregular, fault-tolerant, cost-effective multistage networks named as modified fault-tolerant double tree (MFDOT) network and quad tree (QT) network has been proposed. In the analysis of these networks, any switch, any multiplexer and any demultiplexer were assumed to have a possibility to fail. In the proposed networks, multiple paths of varying lengths are available between each source-destination pair. The networks provide full access capability even in the presence of multiple faults. Algorithms for computing the path lengths and routing have been developed for the MFDOT and QT networks.

An MFDOT- k , which is statically reroutable, is k fault-tolerant and robust in the presence of more than k faults. Due to the presence of an extra module, an MFDOT- k maintains the permutation capability of a modified double tree (MDOT) network, when any single fault occurs. Moreover, compared to the existing non fault-tolerant double tree (DOT) network, the number of

favourite memory modules of a processor increases from two to $2k$ in case of MFDOT- k network. Quad tree (QT) network is a single switch fault-tolerant in all stages including the first and the last. Rerouting in the presence of faults is accomplished dynamically without resorting to backtracking. The cost analysis showed that the MFDOT and QT networks are cost effective compared to other fault-tolerant multistage networks. This advantage becomes more profound as the network size increases. The major advantage of the proposed irregular MFDOT and QT networks is that there exists shortest path length (i.e 2 only), irrespective of the size of the network, between a processor and its favourite memory modules. For a non-uniform reference model these networks gives larger throughput. In short, the proposed irregular fault-tolerant multistage networks offer many attractive features and appear to be a good candidate for use in multiprocessor systems.

7.2 SUGGESTIONS FOR FUTURE INVESTIGATIONS

The following are the suggestions for extending the work further.

- . The subject of combining multiple layers of subnetworks to improve reliability and performance needs to be explored further.
- . Much work remains to be done in analyzing the capabilities of irregular networks and finding other useful application for these networks.

- The networks proposed in this thesis are realized by using 2x2/3x3 crossbar switches. It would be interesting to study their realization using larger switches.
- The effect of faults on the permutation capability of the proposed networks needs to be studied in detail.
- VLSI implementation of the proposed networks needs investigations.

REFERENCES

1. G.B.Adams III, H.J.Siegel, " The Extra Stage Cube: a Fault-Tolerant Interconnection Network for Supersystems," IEEE Transactions on Computers, Vol. C-31, May 1982, pp. 443-454.
2. G.B.Adams III, D.P.Agrawal, H.J.Siegel, " A Survey and Comparison of Fault-Tolerant Multistage Interconnection Network," IEEE Computer, Vol. 20, No. 6, June 1987, pp. 14-27.
3. D.P.Agrawal, " Testing and Fault-Tolerance of Multistage Interconnection Networks," IEEE Computer, Vol. 15, April 1982, pp. 41-53.
4. D.P.Agrawal, " Graph Theoretical Analysis and Design of Multistage Interconnection Networks," IEEE Transactions on Computers, Vol. C-32, July 1983, pp. 637-648.
5. G.A.Anderson, E.D.Jensen, " Computer Interconnection Networks: Taxonomy, Characteristics and Examples," ACM Computing Surveys, Vol. 7, December 1975, pp. 197-213.
6. A.Avizienis, " Fault-Tolerant Systems," IEEE Transactions on Computers, Vol. C-25, Dec. 1976, pp. 1304-1312.
7. J.L.Baer, Computer Systems Architecture, Potomac, Md: Computer Science Press, 1980.
8. G.H.Barnes et al., " The ILLIAC IV Computer," IEEE Transactions on Computers, Vol. C-17, August 1968, pp. 746-757.
9. K.E.Batcher, " STARAN Parallel Processor System Hardware," AFIPS 1974 Natl. Computer Conf., May 1974, pp. 405-410.
10. K.E.Batcher, " The Flip Network in STARAN," Proc. International Conference on Parallel Processing, August 1976, pp. 65-71.
11. V.E.Benes, Mathematical Theory of Connecting Networks and Telephone Traffic, Academic Press, New York, 1965.
12. D.P.Bhandarkar, " Analysis of Memory Interference in Multiprocessors," IEEE Transactions on Computers, Vol. C-24, Sept. 1975, pp. 897-908.
13. L.N.Bhuyan, D.P.Agrawal, " Design and Performance of Generalized Interconnection Networks," IEEE Transactions on Computers, Vol. C-32, Dec. 1983, pp. 1081-1090.

14. L.N.Bhuyan, " An Analysis of Processor-Memory Interconnection Networks," IEEE Transactions on Computers, Vol. C-34, March 1985, pp. 279-283.
15. L.N.Bhuyan, Q.Yang, D.P.Agrawal, " Performance of Multiprocessor Interconnection Networks," IEEE Computer, Vol. 22, Feb. 1989, pp. 25-37.
16. J.T.Black, K.S.Trivedi, " Multistage Interconnection Network Reliability," IEEE Transactions on Computers, Vol. C-38, Nov. 1989, pp. 1600-1604.
17. J.R.Burke, C.Chen, T-Y Lee, D.P.Agrawal, " Performance Analysis of Single Stage Interconnection Networks", IEEE Transactions on Computers, Vol. C-40, March 1991, pp. 357-365.
18. V.Cherkassky, E.Opper, M.Malek, " Reliability and Fault Diagnosis Analysis of Fault-Tolerant Multistage Interconnection Networks," Proc. 14th International Symposium on Fault-Tolerant Computing, June 1984, pp. 246-251.
19. L.Ciminiera, A.Serra, " A Fault-Tolerant Connecting Network for Multiprocessor Systems," Proc. International Conference on Parallel Processing, Aug. 1982, pp. 113-122.
20. L.Ciminiera, A.Serra, " A Connecting Network with Fault - Tolerance Capabilities," IEEE Transactions on Computers, Vol. C-35, June 1986, pp.578-580.
21. C.Clos, " A Study of Nonblocking Switching Networks," Bell System Technical Journal, Vol. 32, March 1953, pp. 406-424.
22. C.R.Das, L.N.Bhuyan, " Computation Availability of Multiple Bus Multiprocessors," Proc. International Conference on Parallel Processing, 1985, pp. 807-813.
23. J.B.Dennis, " Data Flow Supercomputers," IEEE Computer, Vol. 13, Nov. 1980, pp. 48-56.
24. N.Deo, Graph Theory with Application to Engineering and Computer Science, Prentice-Hall Inc., Englewood Cliffs, NJ 1974.
25. D.M.Dias, J.R.Jump, " Analysis and Simulation of Buffered Delta Networks," IEEE Transactions on Computers, Vol. C-30, April 1981, pp.273-282.
26. D.M.Dias, J.R.Jump, " Packet Switching Interconnection Networks for Modular Systems," IEEE Computer, Vol. 14, No. 12, Dec. 1981, pp. 43-54.

27. P.H.Enslow," Multiprocessor Organization - A Survey," Computing Surveys, Vol. 9, March 1977, pp. 103-130.
28. T.Y.Feng," Data Manipulating Functions in Parallel Processors and their Implementations", IEEE Transactions on Computers, Vol. C-23, March 1974, pp. 309-318.
29. T.Y.Feng," Guest Editorial: An Overview of Parallel Processing," Computing Surveys, Vol. 9, March 1977, pp. 1-2.
30. T.Y.Feng, C.L.Wu," Fault Diagnosis for a class of Multistage Interconnection Networks", IEEE Transactions on Computers, Vol. C-30, Oct. 1981, pp. 743-758.
31. T.Y. Feng," A Survey of Interconnection Networks," IEEE Computer, vol. 14, No.12, Dec. 1981, pp. 12-27.
32. M.J.Flynn," Very High Speed Computing Systems," Proceedings IEEE, Vol. 54, Dec. 1966, pp. 1901-1909.
33. W.K.Fuchs, J.A.Abraham, K.H.Huang," Concurrent error detection in VLSI Interconnection Networks", Proc. 10th Annual Symposium on Computer Architecture, June 1983, pp. 309-315.
34. L.R.Goke, G.J.Lipovski," Banyan Networks for Partitioning Multiprocessing Systems," Proc. 1st Annual Symposium on Computer Architecture, Dec. 1973, pp. 21-28.
35. J.A.Harris, D.R.Smith," Hierarchical Multiprocessor Organizations," Proc. Fourth Annual Symposium on Computer Architecture, March 1977, pp. 41-48.
36. J.P.Hayes, T.N.Mudge, Q.F.Stout, S.Colley," Architecture of a Hypercube Supercomputer," Proc. International Conference on Parallel Processing, Aug. 1986, pp. 653-660.
37. R.W.Hockney, C.R.Jeshope, Parallel Computers, Bristol, England: Adam Hilger Ltd., 1981.
38. K.Hwang, F.A.Briggs, Computer Architecture and Parallel Processing. New York: Mcgraw-Hill, 1984.
39. K.Hwang," Supercomputers: Design and Applications," Tutorial, IEEE Computer Society, 1984.
40. M.Jeng, H.J.Siegel," A Fault-Tolerant Multistage Interconnection Network for Multiprocessor Systems using Dynamic Redundancy," Proc. 6th International Conference on Distributed Computing Systems, May 1986, pp. 70-77.
41. D.J.Kuck," A Survey of Parallel Machine Organization and Programming," Computing Surveys, Vol. 9, March 1977, pp. 29-60.

42. H.T.Kung, " Why Systolic Architectures?," IEEE Computer, Vol. 15, January 1982, pp. 37-46.
43. V.P.Kumar, S.M.Reddy, " Design and Analysis of Fault-Tolerant Multistage Interconnection Networks with low link Complexity," Proc. 12th International Symposium on Computer Architecture, June 1985, pp. 376-386.
44. V.P.Kumar, " On Highly Reliable, High Performance Multistage Interconnection Networks," Ph.D. dissertation, University of Iowa, Dec. 1985.
45. M.Kumar, J.R.Jump, " Performance of Unbuffered Shuffle-Exchange Networks," IEEE Transactions on Computers, Vol. C-35, June 1986, pp. 573-578.
46. V.P.Kumar, S.M.Reddy, " Augmented Shuffle-Exchange Multistage Interconnection Networks," IEEE Computer, Vol. 20, No. 6, June 1987, pp. 30-40.
47. V.P.Kumar, A.L.Reibman, " Failure Dependent Performance Analysis of a Fault-Tolerant Multistage Interconnection Network," IEEE Transactions on Computers, Vol. C-38, Dec. 1989, pp. 1703-1713.
48. V.P.Kumar, S.J.Wang, " Reliability Enhancement by Time and Space Redundancy in Multistage Interconnection Networks", IEEE Transactions on Reliability, Vol. 40, No. 4, Oct. 1991, pp. 461-473.
49. I.Koren, Z.Koren, " Analyzing the Connectivity and Bandwidth of Multi-Processors with Multistage Interconnection Networks," Concurrent Computations: Algorithms, Architecture, and Technology, S.Tewksbury, B.Dickinson, S.Schwartz, Eds. New York: Plenum, 1988, pp. 525-540.
50. I.Koren, Z.koren, " On the Bandwidth of a Multistage Networks in the Presence of Faulty Components," Proc. 8th International Conference on Distributed Computing Systems, 1988, pp. 26-31.
51. C.P.Kruskal, M.Snir, " Performance of Multistage Interconnection Networks for Multiprocessors," IEEE Transactions on Computers, Vol. C-32, Dec. 1983, pp. 1091-1098.
52. C.P.Kruskal, M.Snir, " A Unified Theory of Interconnection Network Structure," Theoret. Computer Sci., Vol. 48, No. 1, 1986, pp. 75-94.
53. T.Leng et al., " Bandwidth of Crossbar and Multibus Connections for Multiprocessors," IEEE Transactions on Computers, Vol. C-31, December 1982, pp. 1227-1234.

54. D.H.Lawrie," Access and Alignment of Data in an Array Processor," IEEE Transactions on Computers, Vol. C-24, Dec.1975, pp. 1145-1155.
55. M.Lee, C.L.Wu," Performance Analysis of Circuit Switching Baseline Interconnection Networks," Proc. Eleventh Annual Symposium on Computer Architecture, June 1984, pp. 82-90.
56. G.Lee, C.P.Kruskal, D.J.Kuck," The Effectiveness of Combining in Shared Memory Parallel Computers in the Presence of 'hot spots'," Proc. International Conference on Parallel Processing, Aug.1986, pp. 35-41.
57. C.K.C.Leung, J.B.Dennis," Design of Fault-Tolerant Packet Communication Computer Architecture", Proc. International Symposium on Fault-Tolerant Computing, August 1980, pp. 328-335.
58. K.N.Levitt, M.W.Green, J.Goldberg," A Study of the data Communication Problems in a Self repairable Multiprocessor", Proc. AFIPS Conf., Vol. 32 (1968 SJCC).
59. G.J.Lipovski, M.Malek, Parallel Computing: Theory and Comparisons. New York: John Wiley, 1987.
60. M.T.Liu," Distributed Loop Computer Networks," Advances in Computers, Vol. 17, Academic press, 1978, pp. 163-221.
61. R.J.Mcmillen, H.J.Siegel," Performance and Fault Tolerance Improvements in the Inverse Augmented Data Manipulator Network," Proc. 9th Annual Symposium on Computer Architecture, June 1982, pp. 63-72.
62. R.J.Mcmillen, H.J.Siegel," Routing Schemes for the Augmented Data Manipulator Network in an MIMD System," IEEE Transactions on Computers, Vol. C-31, Dec. 1982, pp. 1202-1214.
63. R.Mittal, N.K.Nanda," Control and Mapping Algorithm for a Double Tree (DOT) Network," Int. J. Electronics, Vol. 63, No. 3, 1987, pp. 317-329.
64. T.N.Mudge, J.P.Hayes, D.C.Winsor," Multiple Bus Architecture," IEEE Computer, Vol. 20, No. 6, June 1987, pp. 42-48.
65. K.Padmanabhan, D.H.Lawrie," Fault-Tolerance Schemes in Shuffle/Exchange Type Interconnection Networks," Proc. International Conference on Parallel Processing, August 1983, pp. 71-75.
66. K.Padmanabhan, D.H.Lawrie," A Class of Redundant Path Multistage Interconnection Networks," IEEE Transactions on Computers, Vol. C-32, December 1983, pp. 1099-1108.

67. D.S.Parker," Notes on Shuffle/Exchange Type Switching Networks," IEEE Transactions on Computers, Vol. C-29, March 1980, pp. 213-222.
68. D.S.Parker, C.S.Raghavendra," The gamma Network," IEEE Transactions on Computers, Vol. C-28, Aug. 1980, pp. 694-702.
69. D.S.Parker, C.S.Raghavendra," The Gamma network: A Multiprocessor Interconnection Network with Redundant Paths," Proc. 9th Annual Symposium on Computer Architecture, June 1982, pp. 73-80.
70. J.H. Patel," Performance of Processor-Memory Interconnections for Multiprocessors," IEEE Transactions on Computers, Vol. C-30, Oct. 1981, pp. 771-780.
71. M.C.Pease," The Indirect Binary n-cube Microprocessor Array," IEEE Transactions on Computers, Vol. C-26, May 1977, pp. 458-473.
72. D.K.Pradhan. Ed., Fault-Tolerant Computing, Theory and Techniques, Vol. 1. Englewood cliffs, NJ : Prentice Hall, 1986.
73. G.F.Pfister et al.," The IBM Research Parallel Processor Prototype (RP3): Introduction and Architecture," Proc. International Conference on Parallel Processing, August 1985, pp. 764-771.
74. C.S.Raghavendra, D.S.Parker," Reliability Analysis of an Interconnection Network," Proc. 4th International Conference on Distributed Computing Systems, May 1984, pp. 461-471.
75. C.S.Raghavendra, A.Varma," INDRA: A Class of Interconnection Networks with Redundant Paths," Proc. Real-Time Systems Symposium, December 1984, pp. 153-164.
76. S.M.Reddy, V.P.Kumar," On Fault-Tolerant Multistage Interconnection Networks," Proc. International Conference on Parallel Processing, August 1984, pp.155-164.
77. S.M. Reddy, V.P.Kumar," On Multipath Multistage Interconnection Networks," Proc. 5th International Conference on Distributed Computing Systems, May 1985.
78. A.Satyanarayana, J.N.Hagstrom," A New Algorithm for Reliability Analysis of Multiterminal Networks," IEEE Transactions on Reliability, Vol. R-30, No. 10, October 1981, pp. 325-334.
79. C.L.Seitz," The Cosmic Cube," Communications of the ACM, Vol. 28, No. 1, January 1985, pp. 22-33.

80. J.P.Shen, J.P.Hayes," Fault-Tolerance of a Class of Connecting Networks," Proc. 7th Annual Symposium on Computer Architecture, 1980, pp. 61-71.
81. J.P.Shen," Fault-Tolerance of β -networks in Interconnected Multicomputer Systems," Ph.D.dissertation, Department of Electrical Engineering, University of South California, USA, August 1981.
82. J.P.Shen," Fault-Tolerance Analysis of Several Interconnection Networks," Proc. International Conference on Parallel Processing, August 1982, pp.102-112.
83. J.P.Shen, J.P.Hayes," Fault-Tolerance of Dynamic-Full-Access Interconnection Networks," IEEE Transactions on Computers, Vol. C-33, March 1984, pp. 241-248.
84. H.J.Siegel, S.D.Smith," Study of Multistage SIMD Interconnection Networks," Proc. Fifth Annual Symposium on Computer Architecture, April 1978, pp. 223-229.
85. H.J.Siegel, R.J.Mcmillan, P.T.Mueller, jr.," A Survey of Interconnection Methods for Reconfigurable Parallel Processing Systems," AFIPS 1979 Nat. Comput. Conf., June 1979, pp. 529-542.
86. H.J.Siegel," Interconnection Networks for SIMD Machines," IEEE Computer, Vol. 12, No. 6, June 1979, pp. 57-65.
87. H.J.Siegel," The Theory Underlying the Partitioning of Permutation Networks," IEEE Transactions on Computers, Vol. C-29, Sept. 1980, pp. 791-801.
88. H.J.Siegel, R.J.Mcmillen," Dynamic Routing Tag Schemes for the Augmented Data Manipulator Network," Proc. 8th Annual Symposium on Computer Architecture, May 1981, pp. 505-516.
89. H.J.Siegel, R.J.Mcmillen," The Multistage Cube: A Versatile Interconnection Network," IEEE Computer, Vol. 14, No. 12, Dec. 1981, pp. 65-76.
90. H.J.Siegel, Interconnection Networks for Large-Scale Parralel Processing: Theory and Case Studies, Lexington, MA: Lexington Books, 1985.
91. H.J.Siegel, W.G.Nation, C.P.Kruskal, L.M.Napolitano jr.," Using the Multistage Cube Network Topology in Parallel Supercomputers," Proceedings IEEE, Vol. 77, No. 12, December 1989, pp. 1932-1953.
92. Special Issue on Multiprocessing Technology, IEEE Computer, Vol.18, No. 6, June 1985.

93. H.S.Stone," Parallel Processing with the Perfect Shuffle," IEEE Transactions on Computers, Vol. C-20, Feb. 1971, pp. 153-161.
94. H.S.Stone, High Performance Computer Architecture, Reading, MA: Addison-Wesley, 1987.
95. S.L.Tanimoto," A Pyramidal Approach to Parallel Processing," Proc. 10th Annual Symposium on Computer Architecture, June 1983, pp. 372-378.
96. S.Thanawastien, V.P.Nelson," Interference Analysis of Shuffle/Exchange Networks," IEEE Transactions on Computers, Vol. C-30, August 1981, pp. 545-556.
97. K.J.Thurber," Parallel Processor Architectures - Part 1; General Purpose Systems," Comput. Design, Vol. 18, Jan. 1979, pp. 89-97.
98. K.S.Trivedi, Probability and Statistics with Reliability, Queuing and Computer Science Applications. Englewood Cliffs, NJ: Prentice - Hall, 1982.
99. N. Tzeng, P. Yew, C. Zhu," A Fault-Tolerant Scheme for Multistage Interconnection Networks," Proc. 12th Annual Symposium on Computer Architecture, June 1985, pp. 368-375.
100. N.Tzeng, P. Yew, C. Zhu," The Performance of a Fault-Tolerant Scheme for Multistage Interconnection Network," Proc. International Conference on Parallel Processing, Aug. 1985, pp. 458-465.
101. N.F.Tzeng, P.C.Yew, C.Q.Zhu," Realizing Fault-Tolerant Interconnection Networks via Chaining," IEEE Transactions on Computers, Vol. C-37, April 1988, pp. 458-462.
102. A.Varma, C.S.Raghavendra," Realizations of Permutations on Generalized Indra Networks," Proc. International Conference on Parallel Processing, Aug. 1985, pp. 328-333.
103. A.Varma, C.S.Raghavendra," Performance Analysis of a Redundant Path Interconnection Network," Proc. International Conference on Parallel Processing, Aug. 1985, pp. 474-479.
104. A.Varma, C.S.Raghavendra," Fault-Tolerant Routing of Permutations in Extra-stage Networks," Proc. 6th International Conference on Distributed Computing Systems, May 1986, pp. 54-61.
105. A.Varma, C.S.Raghavendra," Fault-Tolerant Routing in Multistage Interconnection Networks," Proc. 16th Annual Symposium on Fault-Tolerant Computing, June 1986, pp. 104-109.

106. A.Varma, C.S.Raghavendra," Fault-Tolerant Routing in Multistage Interconnection Networks", IEEE Transactions on Computers, Vol. C-38, March 1989, pp. 385-393.
107. S. Wei, G. Lee," Extra group network : a cost-effective fault-tolerant multistage interconnection network," Proc. 15th Annual Symposium on Computer Architecture, June 1988, pp. 108-115.
108. L.D.Wittie," Communication Structures for Large Networks of Microcomputers," IEEE Transactions on Computers, Vol. C-30, April 1981, pp. 264-273.
109. C.Wu, T.Feng," On a Class of Multistage Interconnection Networks," IEEE Transactions on Computers, Vol. C-29, Aug. 1980, pp. 694-702.
110. C.L.Wu, T.Feng, Tutorial: Interconnection Networks for Parallel and Distributed Processing. IEEE Computer Science Press, 1984.
111. D.W.Yan, J.H.Patel, E.S.Davidson," Memory Interference in Synchronous Multiprocessor Systems," IEEE Transactions on Computers, Vol. C-31, November 1982, pp. 1116-1121.
112. C.Q.Zhu, P.C.Yew," A Synchronization Scheme and its Applications for Larger Multiprocessor Systems," Proc. International Conference on Distributed Computing Systems, May 1984, pp. 486-493.

APPENDIX 1

Table 4.4

Network Size: 256

Prob. of Req. Generation by any Processor: 1.00

Fault type	P_a	Bandwidth	% change in P_a and BW	Minimum Ref. prob.	% change
No Fault	.4292	109.88	-	.4292	-
S1	.4279	109.53	0.3	.4265	0.6
S2	.4275	109.45	0.4	.4225	1.6
S3	.4274	109.41	0.4	.4145	3.4
S4	.4272	109.36	0.5	.3969	7.5
S5	.4270	109.30	0.5	.3571	16.8
S6	.4270	109.32	0.5	.2887	32.7
L1	.4267	109.24	0.6	.4243	1.2
L2	.4257	108.99	0.8	.4153	3.2
L3	.4243	108.63	1.1	.3902	9.1
L4	.4238	108.49	1.3	.3426	20.2
L5	.4234	108.40	1.3	.2445	43.0

(Si, Li indicates the switch and loop of stage i)

Table 4.5

Network Size: 256

Prob. of Req. Generation by any Processor: 0.50

Fault type	P_a	Bandwidth	% change in P_a and BW	Minimum Ref. prob.	% change
No Fault	.6596	84.43	-	.3298	-
S1	.6577	84.19	0.3	.3279	0.6
S2	.6572	84.12	0.4	.3251	1.4
S3	.6570	84.09	0.4	.3194	3.2
S4	.6569	84.08	0.4	.3079	6.6
S5	.6566	84.05	0.4	.2825	14.3
S6	.6569	84.08	0.4	.2422	26.6
L1	.6562	83.99	0.5	.3264	1.0
L2	.6548	83.82	0.7	.3203	2.9
L3	.6511	83.34	1.3	.2957	10.3
L4	.6506	83.28	1.4	.2580	21.8
L5	.6503	83.24	1.4	.1813	45.0

Table 4.6

Network Size: 256

Prob. of Req. Generation by any Processor: 0.10

Fault type	P_a	Bandwidth	% change in P_a and BW	Minimum Ref. prob.	% change
No Fault	.9406	24.08	-	.0941	-
S1	.9398	24.06	0.1	.0939	0.2
S2	.9395	24.05	0.1	.0936	0.5
S3	.9391	24.04	0.2	.0929	1.3
S4	.9391	24.04	0.2	.0916	2.6
S5	.9391	24.04	0.2	.0891	5.3
S6	.9393	24.05	0.1	.0858	8.8
L1	.9392	24.04	0.2	.0938	0.3
L2	.9385	24.03	0.2	.0932	0.9
L3	.9264	23.72	1.5	.0827	12.1
L4	.9263	23.71	1.5	.0712	24.3
L5	.9263	23.71	1.5	.0482	48.8

Table 4.7

Network Size: 64

Prob. of Req. Generation by any Processor: 1.00

Fault type	P_a	Bandwidth	% change in P_a and BW	Minimum Ref. prob.	% change
No Fault	.4791	30.67	-	.4791	-
S1	.4715	30.18	1.6	.4639	3.2
S2	.4701	30.08	1.9	.4428	7.6
S3	.4685	29.98	2.2	.3939	17.8
S4	.4685	29.98	2.2	.3084	35.6
L1	.4650	29.76	2.9	.4509	5.9
L2	.4612	29.52	3.7	.4074	15.0
L3	.4540	29.06	5.2	.2783	41.9

Table 4.8

Network Size: 64

Prob. of Req. Generation by any Processor: 0.50

Fault type	P_a	Bandwidth	% change in P_a and BW	Minimum Ref. prob.	Ref. % change
No Fault	.7003	22.41	-	.3502	-
S1	.6914	22.12	1.3	.3412	2.6
S2	.6896	22.07	1.5	.3288	6.1
S3	.6873	21.99	1.9	.2980	14.9
S4	.6881	22.02	1.7	.2525	27.9
L1	.6838	21.88	2.4	.3336	4.7
L2	.6799	21.76	2.9	.3092	11.7
L3	.6613	21.16	5.6	.1939	44.6

Table 4.9

Network Size: 64

Prob. of Req. Generation by any Processor: 0.10

Fault type	P_a	Bandwidth	% change in P_a and BW	Minimum Ref. prob.	Ref. % change
No Fault	.9448	6.05	-	.0945	-
S1	.9414	6.02	0.4	.0938	0.7
S2	.9406	6.02	0.4	.0928	1.8
S3	.9386	6.01	0.7	.0895	5.3
S4	.9396	6.01	0.6	.0861	8.8
L1	.9389	6.01	0.6	.0933	1.3
L2	.9374	6.00	0.8	.0915	3.2
L3	.8872	5.68	0.1	.0484	48.8

RESEARCH PAPERS OUT OF THE WORK

I - Papers Published/Accepted

1. P.K.Bansal, R.C.Joshi, K.Singh, " Fault-Tolerant Routing in Irregular type of Multistage Interconnection Network," Conference and Exhibition on Parallel Computing PARCOM-90, C-DAC, INDIA, Dec. 1990.
2. P.K.Bansal, R.C.Joshi, K.Singh, " Fault-Tolerant Routing in Modified Double Tree (MDOT) Network," Proc. National Conference on Real Time System, INDIA, Feb. 1991, pp. 233-237.
3. P.K.Bansal, K.Singh, R.C.Joshi, " A New Concept for Cost-Effective Fault-Tolerant Multistage Interconnection Network," Proc. 14th National System Conference, INDIA, March 1991, pp. 310-312.
4. P.K.Bansal, K.Singh, R.C.Joshi, G.P.Siroha, " Fault-Tolerant Double Tree Network," Proc. International Conference IEEE INFOCOM 91, USA, April 1991, pp. 462-468.
5. P.K.Bansal, R.C.Joshi, K.Singh, G.P.Siroha, " Fault-Tolerant Augmented Baseline Multistage Interconnection Network," Proc. International Conference IEEE TENCON 91, INDIA, Aug. 1991, pp. 200-204.
6. P.K.Bansal, R.C.Joshi, K.Singh, " On a Irregular Fault-Tolerant Multistage Interconnection Network," International Conference Parallel Computing 91, U.K., Sept. 1991.
7. P.K.Bansal, R.C.Joshi, K.Singh, " Performance and Reliability Analysis of Fault-Tolerant Modular Interconnection Network," Proc. International Conference IEEE SICON 91, SINGAPORE, Sept. 1991.
8. P.K.Bansal, K.Singh, R.C.Joshi, " Quad Tree: a Cost-Effective Fault-Tolerant Multistage Interconnection Network," International Conference IEEE INFOCOM 92, ITALY, May 1992.
9. P.K.Bansal, K.Singh, R.C.Joshi, " Reliability and Performance Analysis of a Modular Multistage Interconnection Network," Accepted in Journal of Microelectronics & Reliability.
10. P.K.Bansal, K.Singh, R.C.Joshi, " Routing and Path Length Algorithm for a Cost-Effective Four Tree Multistage Interconnection Network," Accepted in International Journal of Electronics.

II - Papers Communicated

1. "On a Fault-Tolerant Multistage Interconnection Network," IEE Proceedings - E, Computers and Digital Techniques.
2. "On a Double Tree Multistage Interconnection Network," IEE Proceedings - E, Computers and Digital Techniques.
3. "Fault-Tolerant Routing for an Irregular Multistage Interconnection Network," An International Journal of Computers and Electrical Engineering.