FAULT TOLERANT MULTISTAGE INTERCONNECTION NETWORKS: PERFORMANCE/RELIABILITY TRADEOFFS

BY

SHIH-CHIAN YANG AND JOHN A SILVESTER

Technical Report Ceng 89-22

University of Southern California
Department of Electrical Engineering-Systems
Los Angeles, CA 90089-0781

Abstract

Although multipath multistage interconnection networks (MIN) can tolerate failure to some degree, the network performance is seriously degraded due to nonuniform traffic generated from failed switches. For some applications, it is important to know how much degradation there is and to find ways to reduce it. This paper addresses this problem for packet switching fault tolerant multistage interconnection networks(FTMIN). Two classes of FTMIN's, dual MIN's and dual extra stage MIN's are proposed. It is shown that the bandwidth under single failure is greatly improved among similar FTMIN's without hardware overhead and without sacrificing the bandwidth under fault free operation. The reliability of these networks (full access capability) is reduced under multiple failures, however. This multiple failure scenario has very low probability in the useful range of reliability values and hence the penalty is relatively minor. The concepts of processor traffic set, memory traffic set, processor merging operation and memory forking operation are presented and are found to be useful in designing packet switching FTMIN's.

Index Terms - fault tolerance, graceful degradation, network reliability, multistage interconnection network, packet switching, performance evaluation.

1 Introduction

Research into Multistage Interconnection Networks (MIN) is motivated by the necessity for high bandwidth processor/memory communication in modern computers [7]. In this paper, we are primarily interested in MIN's to support Multiple Instruction stream Multiple Data stream (MIMD) designs. For this type of machine, requests to the interconnection network are asynchronous and may access any memory in random manners that change from cycle to cycle [6]. Packet switching is generally more efficient than circuit switching to support this kind of dynamic traffic pattern. In addition, it is desirable to provide buffering capability at internal stages of the switch to handle conflicting requests. All the networks considered in this paper are packet switched with buffering. References to previous designs that did not originally have such a capability should be construed to mean a suitably extended version.

We limit our attention to MIN's with symmetrical topologies so that the bandwidth and reliability between all processor/memory pairs are identical, since this is most appropriate for MIMD machines.

For large networks, reliability is a crucial issue and redundancy must be included in the design. Many fault tolerant multistage interconnection networks (FTMIN) have been proposed for various applications [2]. Much work has been done on the reliability of these networks [2], where the main concern is that all processors be able to communicate with all memories after failure of some of the switches has occurred. For symmetrical networks, the number of paths per processor/memory pair is a good indication of the reliability. More paths for every processor/memory pair generally improves reliability.

Of related interest is the throughput or bandwidth that can be supported by the MIN, both in the normal operating state and when failures have occurred. This has been little studied. During failure states, we find that nonuniform traffic, which seriously degrades the network bandwidth, is generated. We exploit the tradeoff between reliability and bandwidth for a given hardware complexity and design new networks accordingly. The new networks increase the bandwidth in failure states

without performance degradation of the normal operating state.

In the following section, we compare several MIN's found in the literature, looking at hardware cost, bandwidth, and the number of paths from each processor to each memory. From the comparison, we identify two simple classes of FTMIN's for further study. In section 3, we study the performance of these networks. We also propose some simple topological modifications to the connections, producing a "dual network", that maximizes the bandwidth of this class of networks.

The reliability of these designs is studied in section 4. The modified designs suffer from some reduction in reliability, but this is found to be due to multiple failures which has extremely low probability in the useful failure range.

2 Performance and Reliability Comparison

In Table 1, some parameters of several FTMIN's operated in a packet switching mode are summarized. The intention of the comparison is to give a qualitative feel for the performance of the different design rather than a precise quantitative comparison.

The *hardware cost* of a network is calculated based on the cost of a 2×2 switch. A $R \times R$ switch costs $\frac{R^2}{4}$ and an $1 \times R$ or $R \times 1$ switch costs $\frac{R}{4}$. The numbers may be slightly different from the numbers found in papers defining these networks, but this is due to the different assumptions that we have made.

The *bandwidth* of a network is the maximum throughput it can provide. Exact solution of the bandwidth is dependent on the network configuration and is in general a hard problem [6]. Rough estimates are given and are normalized to the bandwidth of the unique path MIN. Feedback links [13] are not considered in this paper, only one pass is allowed in routing a packet, and therefore all processors and memories are equally important.

As noted above, we consider only symmetrical FTMIN's in this paper, thus the improved IADM[9], the Gamma network[10] and other similar networks are excluded.

It is interesting to note that in comparing with the unique path MIN, the bandwidth differences of the extra stage MIN[1], the Beneš network[4], the chained MIN[14]

FTMIN's	Hardware cost	Bandwidth	# of paths per proc/mem pair
unique path MIN	$\frac{N}{2}logN$	1	1
extra stage (e.g. ESC)	$\frac{N}{2}logN + \frac{N}{2}$	1	2
Beneš	$NlogN - \frac{N}{2}$	1	N
chained MIN,ASEN	$\frac{9}{8}NlogN$	1	v. large
two copy MIN	$NlogN + N$	2	2
INDRA	$R\frac{N}{2}logN + \frac{NR^2}{4} + \frac{NR}{2}$	R	R^2
ACN	$2NlogN + N$	2	2N
2-MDN	$2NlogN - N$	2	2N
F-net	$4NlogN$	4	N

Table 1: Comparison of hardware cost, Bandwidth, number of paths per P/M pair

and ASEN[8] are insignificant. The additional hardware is mainly devoted to higher reliability. A similar conclusion that much higher reliability can be achieved with these type of topologies can be found in [5] [8].

On the other hand, the bandwidth is proportional to the hardware cost if the number of paths per processor/memory pair is *constant*, e.g. the two copy MIN, which will be defined in section 3, and the INDRA[11]. In these cases, the extra hardware is used mostly to improve performance. For the rest of the networks in the table, the F-net[5], the ACN[12] and the 2-MDN[12], the hardware is used to improve both reliability and performance.

Two important techniques that use extra hardware to improve performance or reliability are *i*) extra stage and *ii*) multiple copy. As can be seen in the above comparison these are reasonable examples of FTMIN's. We focus on these two FTMIN's in this paper and show how their performance can be improved by a simple rearrangement. The techniques developed can also be applied to other FTMIN's.

3 Performance degradation of two path MIN's

A *unique path MIN of size N* is constructed using $n = log_R N$ stages of $R \times R$ switches and there are $\frac{N}{R}$ switches at each stage. Every *outnode* of a switch is connected to an *innode* of a switch at next stage such that there exists a unique path between any innode of the switches at the first stage and any outnode of the switches at the last stage. From the source side to the destination side, the switch stages are numbered from 0 to $n-1$ while the link stages are numbered from 0 to n.

In packet switching, the bit controlled property[6] is essential. It is assumed that each packet contains a destination tag and exactly one bit of the tag is decoded at each stage regardless of its source. The backward bit controlled property, i.e., from destination to source, is also assumed for simplicity.

The unique path MIN defined above forms the building block of the networks to be discussed.

3.1 Two copy MIN's

To interconnect two copies of a unique path subnetwork, sets of processor switches and memory switches are needed. These switches are considered exterior to the MIN and their failures are not included in the following analysis.

Definition:

A *two copy MIN* connects N processors and N memories via two copies of a unique path MIN, using N 1×2 switches to attach the processors and N 2×1 switches to attach the memories, such that there are exactly two paths for each processor/memory pair. □

Although a single failure does not destroy connectivity of the two copy MIN, the network performance is degraded. For this analysis, a uniform and independent traffic pattern to the network is assumed, i.e. all processors generate packets independently and with equal probability. It is also assumed that all memories are equally likely destinations for a packet. At each processor, the existence of an infinite length buffer, called the *processor queue*, is assumed. In addition, there is a finite buffer, called

the *switch queue*, at each innode of every switch. If the first switch queue is full, the newly generated packet is inserted into the processor queue.

The *traffic flow* on the path from processor i to memory j, defined to be A_{ij}, is given by:

$$A_{ij} = \lim_{t \to \infty} \frac{A_{ij}(t)}{t}$$

where,

$$A_{ij}(t) = \#\ packets\ leaving\ processor\ i\ for\ memory\ j\ during\ t\ cycles$$

Note that there is no assumption about how the traffic is distributed to different destinations. It is just the average flow rate of packets sent from a processor to a memory.

A processor queue is saturated when the queue length goes to infinity (as time goes to infinity). Since no packets are dropped, the network traffic flow is equal to the network throughput if none of the processor queues is saturated. A good performance measure of a MIN is the bandwidth which is defined as follows.

Definition:

The *bandwidth* of the network is the maximum network traffic flow per processor such that none of the processor queues is saturated. □

For simplicity, we assume that a generated packet is randomly directed to either copy with probability $\frac{1}{2}$. The processors and memories are assumed to be fast, i.e. it is possible for a processor to generate more than one packet at a time and a memory to service more than one packet at a time.

A *normal two copy MIN* with two identical unique path subnetworks is shown in Figure 1. The sources of these two subnetworks at the same location are attached to the same processor. The memories are attached in a similar way. Since no packet is dropped, the total traffic flow on any link stage is equal to all others. If there is no failure, the traffic distribution of all links at the same stage are identical. Hence the traffic flows for all the links are identical and are given by:

$$A = N \cdot A_{ij} \qquad \forall i, j$$

If a switch of one subnetwork is broken, the packets through that switch have to be rerouted to the other subnetwork and a nonuniform traffic pattern is created. In Figure 1, the second switch of stage 1 is broken, hence the traffic flow through the innodes and outnodes of that switch, which consists of the traffic from processor 0, 1, 2, 3 to memory 4, 5, 6, 7, is zero. One half of the original traffic from processor 0, 1, 2 and 3 will still be directed to this subnetwork. The other half is rerouted to the second subnetwork. If the topologies of these two subnetworks are identical, the switch corresponding to the failed switch in the second subnetwork becomes the bottleneck, since all rerouted traffic passes through it. The total amount of traffic flow at the duplicate switch is twice as much as that in the original network. The bottleneck can be removed by distributing the rerouted packets to other switches, e.g. as in Figure 2, by connecting the link stage 0 and link stage n in different patterns. In the design shown, the bottleneck is shifted to link stage 0 which has a flow of $\frac{3A}{2}$ instead of the maximum of $2A$ found in the original design. Also, the extra traffic due to rerouting is shared by more links at all stages.

3.2 Processor traffic set and memory traffic set

For a unique path network (i.e. of the two subnetworks), two packets from two different processors with the same destination will merge at some switch and will always pass through the same switches thereafter. Conversely, the packets from a particular processor directed toward different memories will fork at some switch and will never pass through the same switch thereafter. These ideas are similar to the buddy property in[3]. In order to calculate the traffic flow on each link and explore the problem of the nonuniform traffic pattern caused by the failure, the merging and forking properties are characterized by the processor traffic sets and memory traffic sets.

Definition:

For a unique path subnetwork, the *processor traffic set* associated with a link is defined by:

i) At link stage zero, the set contains only the processor attached to the link.

ii) At any other link stage, the set associated with a link is the union of the sets associated with the previous stage links that are connected to the same switch. □

Definition:

For a unique path subnetwork, the *memory traffic set* associated with a link is defined by:

i) At link stage n, the set contains only the memory attached to the link.

ii) At any other link stage, the set associated with a link is the union of the sets associated with the next stage links that are connected to the same switch element. □

The processor traffic sets associated with the outnodes of the same switch are the same since they are the union of the processor traffic sets associated with the innodes. Also, if two outnodes merge at some switch, then their processor traffic sets are the same. The same arguments can be applied to the memory traffic sets. The packets that pass through a switch originate from the processors that are the elements of the processor traffic set associated with the outnode links and are destined toward the memories that are the elements of the memory traffic set associated with the innode links. Hence the cross product of these two traffic sets contains all the processor/memory pairs passing through the switch. If this switch is broken, all the packets via these paths have to be rerouted.

The processor traffic set associated with a link contains all the processors that send packets through that link. The memory traffic set associated with a link contains all the memories to which packets using the link are destined. The switches combine the possible processors and distinguish the possible memories. At stage zero, the switches merge two possible processors and fork the memories into different groups, containing the possible memories of each outnode link respectively. Until at the last link stage, the only possible memory is the memory attached to the link and all processors may use this link.

Definition: *processor merging operation*

The processor traffic sets associated with the innode links of a switch element are merged to the processor traffic set associated with any outnode link of the same

switch element. □

Definition: *memory forking operation*

The memory traffic set associated with an innode link of a switch element forks into the memory traffic sets associated with all outnode links of the same switch element. □

Definition: *processor merging (memory forking) sequence*

The sequences of merging (forking) processors (memories) in n stages. □

3.3 Dual MIN's

The bottleneck of the two copy MIN operating under a single failure is due to the rerouted packets all passing through the duplicate switch corresponding to the failed switch. If a different processor pattern is assigned in the second copy subnetwork, such that the processor merging sequence is reversed, the rerouted packets from different processors will be merged at the latest possible stage. The processors carrying rerouted traffic will be merged only with the processors without rerouted traffic at early stages. Similarly, if a different memory pattern is assigned in the second copy subnetwork such that the memory forking sequence is reversed, the rerouted packets to different memories will be forked at the earliest possible stage.

Definition:

A *dual MIN* is a two copy MIN with reversed memory forking sequence and reversed processor merging sequence in the second copy subnetwork. □

To prove that the bandwidth of the dual MIN is maximal for any single failure, we consider the failure at a particular stage and then extend it to all stages.

Lemma:

The bandwidth of a two copy network under a uniform independent traffic pattern with a single switch failure, at stage s, is maximal if and only if all the memory traffic sets forked after stage s in the failure subnetwork are forked before stage $n-s-1$ in the other subnetwork and all the processor traffic sets merged before stage s in the failure subnetwork are merged after stage $n-s-1$ in the other subnetwork. □

Proof:

The traffic through the failed switch has to be rerouted to the other subnetwork. The paths that need rerouting are the cross product of the processor traffic set associated with the outnode and the memory traffic set associated with the innode of the failed switch. There are R^{s+1} processors and R^{n-s} memories in the cross product set. The traffic flow for each path is $\frac{A}{N}$. Hence, if none of the processor queues is saturated, the total rerouted traffic flow is

$$R^{s+1}R^{n-s}\frac{A}{N} = R \cdot A$$

At each link stage, the rerouted traffic has to be shared by some links while the other links carry only the original traffic. At link stage 0, only the R^{s+1} links that are in the rerouted processor traffic set carry rerouted traffic with traffic flow $R^{n-s} \cdot \frac{A}{N}$ per link.

At link stage $i \leq n-s-1$, the rerouted processor traffic has not yet been merged but the rerouted memory traffic has been forked stage by stage. Hence $R^{s+1} \cdot R^i$ links share the rerouted traffic and the rerouted traffic flow is $1 \cdot R^{n-s-i} \cdot \frac{A}{N}$ per link. After stage $n-s-1$, the rerouted processor traffic is merged stage by stage but the rerouted memory traffic is not forked any more. At link stage $i \geq n-s$, there are

$$R^{s+1-[i-(n-s-1)]} \cdot R^{n-s} = R^{n-i} \cdot R^{n-s}$$

links with rerouted traffic flow $R^{s+1-n+i} \cdot 1 \cdot \frac{A}{N}$ per link. Note that, in addition to the rerouted traffic, the second subnetwork still carries its original traffic. Since the rerouted packets are added on top of the original uniformly spread traffic, the links carrying rerouted packets are more heavily loaded than the other links. Therefore these links will reach the channel capacity first as more traffic is added to the network. If there are less links sharing the rerouted traffic or more rerouted traffic in these links, the network will be saturated with a smaller traffic flow.

Assume that for some stage j, $j < n-s-1$, there is a processor merging operation for rerouted traffic at stage j, the number of links that share the rerouted traffic after stage j will be decreased by a factor of R and the rerouted traffic flow will be increased by a factor of R per link. Similarly, if there is a memory forking operation for rerouted

traffic at stage j, the number of links that share the rerouted traffic after stage j will be decreased by a factor of R and the rerouted traffic flow will be increased by a factor of R per link. Between stage j and s, each link stage has less links sharing rerouted traffic, hence less bandwidth.

Conversely, since there are $s+1$ processor merging operations required for the rerouted packets and at most $s+1$ processor forking operations after stage $n-s-1$, it is not possible to move any other processor forking operation to after stage $n-s-1$. Similarly, since there are $n-s$ memory forking operations required for the rerouted packets and only $n-s-1+1$ possible memory forking operations before stage $n-s-1$, it is not possible to move any other memory forking operation to before stage $n-s-1$. There is no other processor merging, memory forking sequence for which the rerouted traffic is shared over *more* links. QED

Theorem:

The bandwidth of the two copy MIN under a uniform independent traffic pattern with a single switch failure is maximal if and only if it is a dual MIN. □

Proof:

For the failure in either subnetwork, the blocked traffic will be routed to the other subnetwork. Since the processor merging sequence and the memory forking sequence are reversed, from the lemma, the bandwidth is maximal.

Conversely, assume that the two copy network is not a dual network. First, consider the case that it is due to the processor merging sequence. We can always find a processor merging operation at stage i and the corresponding processor merging operation of the other subnetwork at stage $n-j-1$, $j > i$. If the switch failure is at a stage s, $j > s \geq i$, then the processor merging operation for the rerouted traffic of the other subnetwork occurs before stage $n-s-1$. From the lemma, the network is not maximal. Similarly, consider the case that it is due to the memory forking sequence. We can always find a memory forking operation at stage i and the corresponding memory forking operation of the other subnetwork at stage $n-j-1$, $j < i$. If the switch failure is at a stage s, $j < s \leq i$, then the memory forking operation for the rerouted traffic of the other subnetwork occurs after stage $n-s-1$. QED

3.4 Dual extra stage MIN's

Another way to construct a multipath MIN is by introducing an extra stage[1] in front of a unique path MIN as shown in Figure 3. In this example, the dark switches form a unique path subnetwork with size $\frac{N}{2}$ while the shaded switches form another unique path subnetwork. Any processor can access all memories via each outnode of the switch that it is attached to, hence multiple paths are provided. The network reliability is increased by adding only one stage of switches instead of a complete unique path MIN. The bandwidth of this network with a single switch failure can also be increased by proper topological design. In this section, only MIN's using 2×2 cross bar switches are considered.

Definition:

The *dual extra stage MIN* is an extra stage MIN using 2×2 crossbar switches such that, excluding the extra and last stages, the network is divided into two subnetworks which are dual, with reversed memory forking sequence and reversed processor merging sequence. □

Theorem:

If a single switch failure occurs in an interior stage, the bandwidth of a dual extra stage MIN under a uniform independent traffic pattern is maximal among the extra stage MIN's using 2×2 switches. □

Proof:

The interior subnetwork acts like a two copy network of size $\frac{N}{2}$. The bandwidth of the interior subnetwork with a single switch failure is maximal if and only if it is a dual MIN, hence the dual extra stage MIN is maximal. QED

As suggested by[1], if the switch failure occurs at the extra stage or the last stage, bypass links are required to maintain the connectivity. In this case, the dual extra stage MIN offers the same performance as any other extra stage MIN.

3.5 Simulation results for two path MIN's

The *normalized bandwidth*, which is the bandwidth per copy, of the normal two copy MIN and the dual MIN under a single failure have been studied by simulation. In Figure 5, normalized bandwidth vs. location of failure (i.e., which stage) of a six stage two copy MIN with 2×2 switches is depicted. The switch buffer size is 4. The top curve is for the failure free case. The bottom curve is for a single failure where the two subnetworks use the same processor merging sequence and the same memory forking sequence. A failure occurring near the processors is worse because blocking due to finite switch buffer size propagates back to the processor. The middle curve shows the normalized bandwidth for a dual MIN. If the failure occurs at interior stage, the normalized bandwidth is quite close to the failure free case.

In Figure 6, normalized bandwidth vs. number of processors is shown (also with switch buffer size of 4). The normalized bandwidth shown is the average over all possible single failures. It is shown that the improvement of the dual MIN is slightly larger as the network size increases. Hence, we can conclude that at least a *constant* improvement is achieved over all network size.

In Figure 7, normalized bandwidth vs. switch buffer size is plotted. The improvement of the dual MIN is observed over all switch buffer sizes. Although the normalized bandwidth for small buffer size is poor, there is still some improvement.

In Figure 8, bandwidth vs. location of failure of a six stage (one extra stage and five stages for the unique path MIN) extra stage MIN's with 2×2 switches and 4 switch buffers is depicted. The failures at the extra stage and the last stage are not shown since they destroy the connectivity of the network. A similar improvement can be observed in these curves.

4 Network reliability of two path MIN's

Full access capability, i.e. at least one path exists between any processor/memory pair, is a good reliability criterion and has been used by many researchers[2]. The switch fault model is used in this analysis, and also it is assumed that the failure

probabilities of the switches are independent and identical.

In a two path MIN, a single failure can be tolerated, while multiple failures may destroy the network full access capability. However, some multiple failures do not destroy the network full access capability. Failure of any of the switches carrying rerouted traffic destroys the full access capability. For the normal two copy MIN in Figure 1, there are five such switches, whereas for the dual MIN in Figure 2, failure of any switch in the second subnetwork carrying rerouted traffic is a problem. It is obvious that the network reliability of a normal two copy MIN is better than that of a dual MIN, which is the price paid for more switches sharing rerouted traffic in order to reduce bottlenecks. The penalty is insignificant in the range of interest of failure probabilities, however, where the high failure count is of less importance.

4.1 Reliability bounds of two copy MIN's

Consider a failure pattern V in the first subnetwork. The number of switch failures in the pattern V is $k(V)$ and the number of switches *covered by* the failure pattern V (i.e., the number of switches carrying rerouted traffic in the second subnetwork) is $r(V)$. Let $S = \frac{NlogN}{2}$ be the number of switches in the n stage unique path MIN. Assuming that the reliability of a switch is p, then the network reliability for a two copy MIN is

$$R^{2c} = \sum_{all\ V} p^{S-k(V)}(1-p)^{k(V)}p^{r(V)}$$

The exact solution of network reliability is quite involved, and hence only an upper bound and lower bound are studied.

The network reliability can also be computed as follows:

$$R^{2c} = \sum_{k=0}^{S} R_k$$

where $R_k = \Pr\{\text{full access capability exists} \mid \text{exactly } k \text{ failures in any one subnetwork and at least } k \text{ failures in the other subnetwork}\}$. For three failures in a two copy MIN, they can be grouped as $(3,0)$, $(2,1)$, $(1,2)$ or $(0,3)$, where the first component is the number of switch failures in first subnetwork and second component is the number of switch failures in the second network. $(3,0)$, $(0,3)$ are contained in R_0 while $(2,1)$,

$(1,2)$ are contained in R_1. Hence, R_0 and R_1 include all the failure patterns for less than or equal to three failures and for some higher failure counts. $R_0 + R_1$ is thus a reasonable lower bound for the reliability range of interest. (Since two copy MIN's are designed to be used with only one or two failures.) The lower bound $\underline{R}^{2c}$ is:

$$\underline{R}^{2c} = R_0 + R_1 < R^{2c}$$

Evaluation of R_0 and R_1 is given in the Appendix.

Let X be Pr{full access capability exists | each copy has at least two failures} and

$$X = \sum_{k=2}^{S} R_k = R^{2c} - R_0 - R_1$$

For any failure pattern of the first subnetwork, let us arbitrarily choose one of the failed switches as a tagged switch. Let P_s = Pr{the tag switch is at stage s and there are at least two failures in the first subnetwork}. Then,

$$P_s = \frac{1 - p^S - Sp^{S-1}(1-p)}{n}$$

Let X_s be the conditional probability of the network is operational when there are at least two failures and the tagged switch is at stage s. Then

$$X = \sum_{s=0}^{n-1} P_s X_s$$

X_s is very hard to compute. A simple upper bound can be obtained by considering only those switches in the second subnetwork which carry extra traffic due to the failure of the tagged switch.

$$X_s < \overline{X}_s$$

where $\overline{X}_s$ is an upper bound given by:

$$\overline{X}_s = p^{\Gamma_s}[1 - p^{S-\Gamma_s} - (S - \Gamma_s)p^{S-\Gamma_s-1}(1-p)]$$

where Γ_s is the number of switches in the second subnetwork covered by the single failure of the tagged switch in stage s of the first subnetwork. The upper bound of the network reliability, $\overline{R}^{2c}$, is:

$$\overline{R}^{2c} = R_0 + R_1 + \sum_{s=0}^{n-1} P_s \overline{X}_s$$

By combinatorial calculation, Γ_s can be obtained and are given in the Appendix.

The reliability of a unique path MIN, upper and lower bounds on the reliabilities of a normal two copy MIN and a dual MIN are plotted in Figures 9 and Figure 10 for 5 and 10 stages respectively. The bounds are satisfactory in the range of interest where small number of failures is more important. The improvement over the unique path MIN is quite significant for both the two copy MIN and the dual MIN. The difference between these two networks is relatively small in the range of interest. Therefore, the penalty paid for the dual MIN is acceptable.

4.2 Reliability bounds of extra stage MIN's

In the following analysis, it is assumed that there is no bypass link for the switches at the extra stage and the last stage, therefore, any failure in these two stages destroys the network full access capability. As shown in Figure 3, the interior stages form two unique path MIN's with the size $\frac{N}{2}$ and reliability R^{2c}. The network reliability of an extra stage MIN, R^x, therefore, is

$$R^x = p^{2\cdot\frac{N}{2}} R^{2c} = p^N R^{2c}$$

The bounds of the network reliabilities of the normal extra stage MIN and the dual extra stage are shown in Figure 10. The bounds are quite satisfactory since the high failure count case is insignificant because of the fault free requirement of the extra and last stages. A clear improvement of extra stage MIN and dual extra stage MIN over the unique path MIN is observed over the range of interest, as expected.

5 Conclusion

In this paper, we note that a single switch failure may seriously degrade the bandwidth of a two path MIN, even though full access capability is maintained. The degradation which is due to the bottleneck in the non-failed copy of the MIN can be reduced by properly rearranging the connections. Two new classes of FTMIN's: dual MIN and

dual extra stage MIN, are proposed accordingly. The construction technique described can easily be extended to various FTMIN's.

For single failures, it is shown that the system bandwidth of the new networks are maximal among the class of MIN's to which they belong. Simulation results support this and show that a significant improvement over a replicated MIN is obtained. For this design, there is no additional hardware overhead and the bandwidth of the failure free MIN is not sacrificed. However, the network reliability is slightly degraded when higher numbers of failures are considered. Upper and lower bounds on network reliability are found and it is shown that the degradation is not significant over the useful range of reliability values.

We introduce two essential concepts to packet switching MIN design: processor merging, and memory forking operations, which can be used to calculate traffic flows. We also use these to show the maximality of our design and anticipate that they are useful for many other considerations.

Appendix

R_0 can be evaluated by subtracting the probability of both subnetworks being failure free from the sum of the probabilities of each being failure free.

$$R_0 = 2 \cdot p^S - p^{2 \cdot S}$$

Since single failures at the same stage have the same distribution, R_1 can be evaluated by the conditional probability that the single failure occurred at stage s.

$$\begin{aligned} R_1 &= 2 \sum_{s=0}^{n-1} 2^{n-1} p^{S-1}(1-p) p^{\Gamma_s}(1 - p^{S-\Gamma_s}) \\ &- \sum_{s=0}^{n-1} 2^{n-1} p^{S-1}(1-p)(S-\Gamma_s) p^{S-1}(1-p) \end{aligned}$$

where Γ_s is the number of switches in the second subnetwork covered by the single failure of the tagged switch in stage s of the first subnetwork.

For a single failure at stage s, the set of paths corresponding to the cross product of 2^{s+1} processors and 2^{n-s} memories is rerouted. For a normal two copy MIN, the rerouted processor sets are merged stage by stage until stage s while the rerouted memory traffic sets are not forked. There are 2^{s-i} switches that carry rerouted traffic at stage i. After stage s, the rerouted processor sets are not merged any more, but the rerouted memory traffic sets start to be forked. There are $2^{n-s-1-i}$ switches that carry rerouted traffic at stage $n-1-i$. Let Γ_s^n be the number of covered switches for a normal two copy MIN. Then,

$$\begin{aligned} \Gamma_s^n &= \sum_{i=0}^{s} 2^{s-i} + \sum_{i=0}^{n-s-2} 2^{n-s-1-i} \\ &= 2^{s+1} + 2^{n-s} - 3 \end{aligned}$$

For a dual MIN, the rerouted processor sets are not merged until stage s while the rerouted memory traffic sets are forked at each stage. After stage s, the rerouted processor sets are now merged, but the rerouted memory traffic sets are no longer forked. Hence, the number of switches carrying rerouted traffic can be computed by considering the stages before and after s. Let Γ_s^d be the number of covered switches for a dual MIN. Then,

$$\Gamma_s^d = r_{before} + r_{after}$$

where

$$r_{before} = \begin{cases} \sum_{i=0}^{n-s-2} 2^{s+i+1} = 2^{n-1} - 2^{s+1} & for\ s \leq n-3 \\ 0 & otherwise \end{cases}$$

$$r_{after} = \begin{cases} \sum_{i=0}^{s-2} 2^{n-s+i} = 2^{n-1} - 2^{n-s} & for\ s \geq 2 \\ 0 & otherwise \end{cases}$$

References

[1] G. B. Adams III and H. J. Siegel, "The extra stage cube: A fault tolerant interconnection network for supersystems," IEEE Trans. Comput., vol. C-31, pp. 443-454, May 1982.

[2] G. B. Adams III, D. P. Agrawal and H. J. Siegel, "A Survey and Comparison of Fault-Tolerant Multistage Interconnection Networks," Computer, Jun. 1987, pp.14-27.

[3] D. P. Agrawal, "Graph Theoretical Analysis and design of Multistage Interconnection Networks," IEEE Trans. Computers, Jul. 1983, pp.637-648

[4] V. E. Beneš, Mathematical Theory of Connecting Networks and Telephone Traffic. New York: Academic, 1965.

[5] L. Ciminiera and A. Serra, "A Connecting Network with Fault Tolerance Capabilities," IEEE Trans. Computers, Jun. 1986, pp. 578-580.

[6] D. M. Dias and J. R. Jump, "Analysis and simulation of buffered delta networks," IEEE Trans. Comput., Apr. 1981, pp. 273-282.

[7] T. F. Feng, "A Survey of Interconnection Network," Computer, Dec. 1981, pp.12-27.

[8] V. P. Kumar and S. M. Reddy, "Augmented Shuffle-Exchange Multistage Interconnection Networks," Computer, Jun. 1987, pp30-40.

[9] R. J. McMillen and H. J. Siegel, "Performance and Fault Tolerance Improvements in the Inverse Augmented Data Manipulator Network," 9th Symp. Computer Architecture, Apr. 1982, pp. 63-72.

[10] D. S. Parker and C. S. Raghavendra, "The Gamma Network," IEEE Trans. Comput., Vol. C-33, pp367-373, Apr. 1984.

[11] C. S. Raghavendra and A. Varma, "INDRA: A Class of Interconnection networks with Redundant Paths," 1984 Real-Time System Symp., Computer Society Press, Silver Spring, Md., 1984, pp.153-164.

[12] S. M. Reddy and V. P. Kumar, "On Fault-Tolerant Multistage Interconnection Networks," 1984 Int'l Conf. Parallel Processing, Computer Society Press, Silver Spring, Md., 1984, pp.155-164.

[13] J. P. Shen and J. P. Hayes, "Fault-Tolerance of Dynamic-Full-Access Interconnection Networks," IEEE Trans. Comput., pp. 241-248, Mar. 1984.

[14] N. F. Tzeng, P. C. Yew, and C. Q. Zhu, "The Performance of a Fault-Tolerant Multistage Interconnection Networks," 1985 Int'l Conf. Parallel Processing, Computer Society Press, Silver Spring, Md., 1985, pp.458-465.

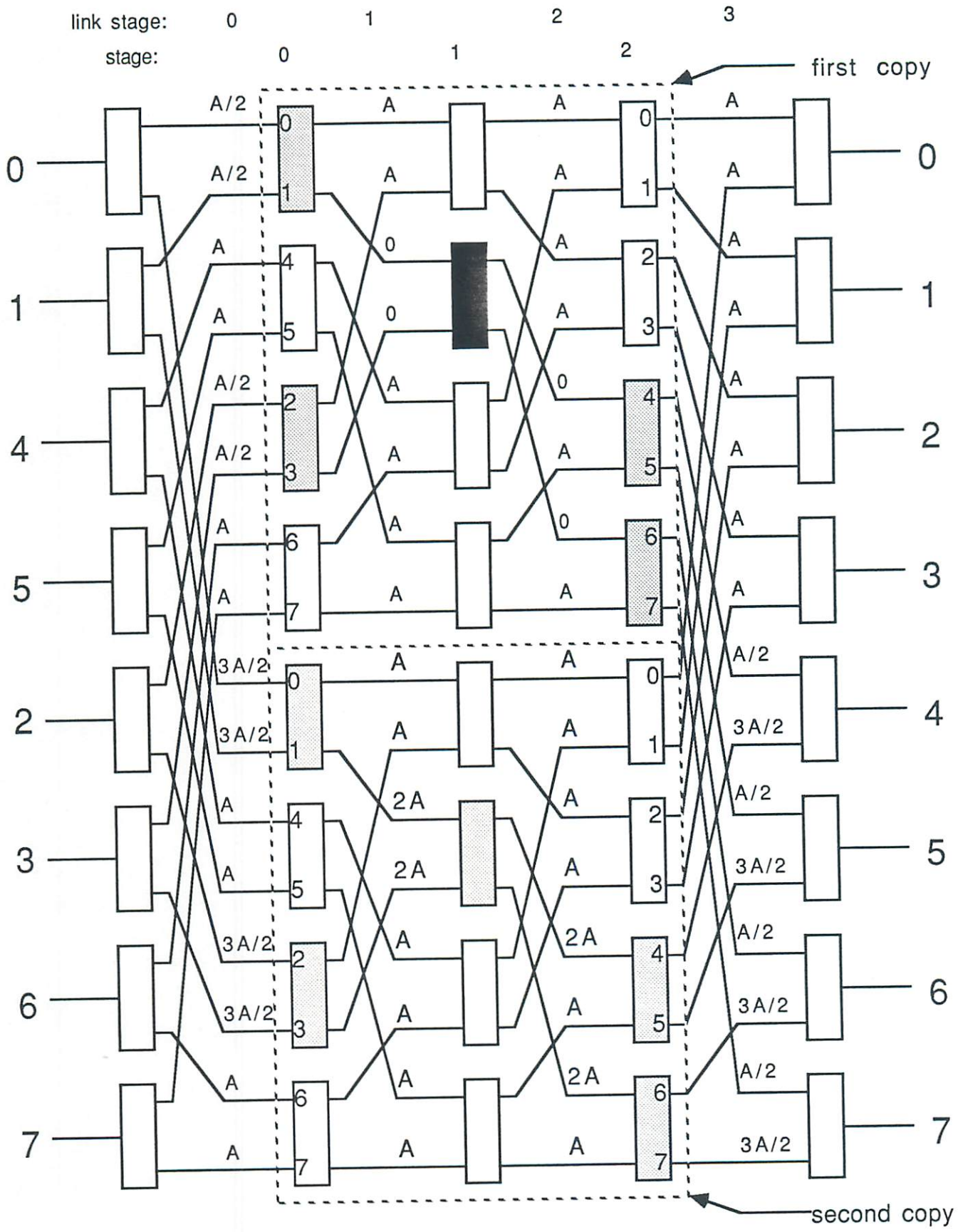

Figure 1. A Two Copy MIN with size 8

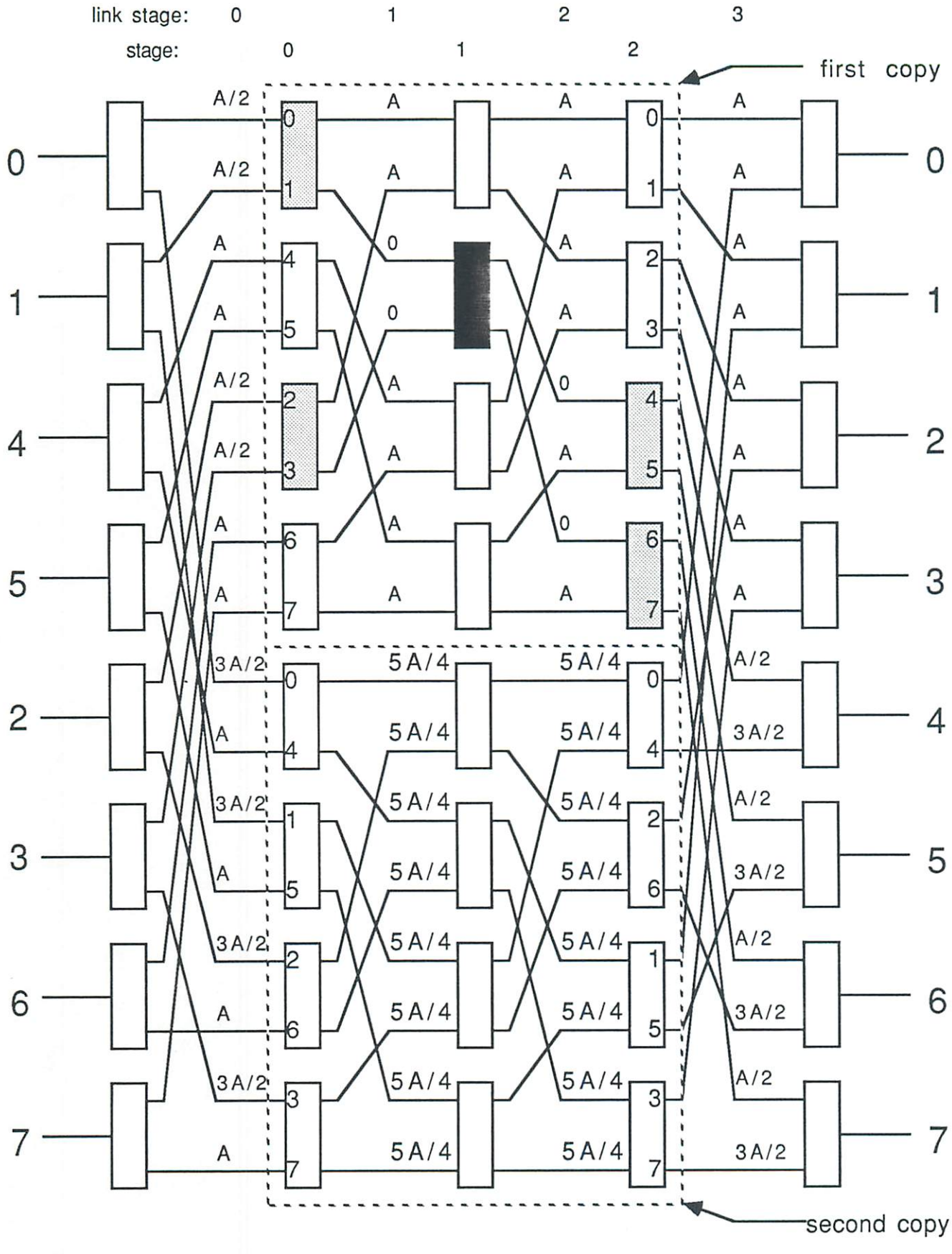

Figure 2. A Dual MIN with size 8

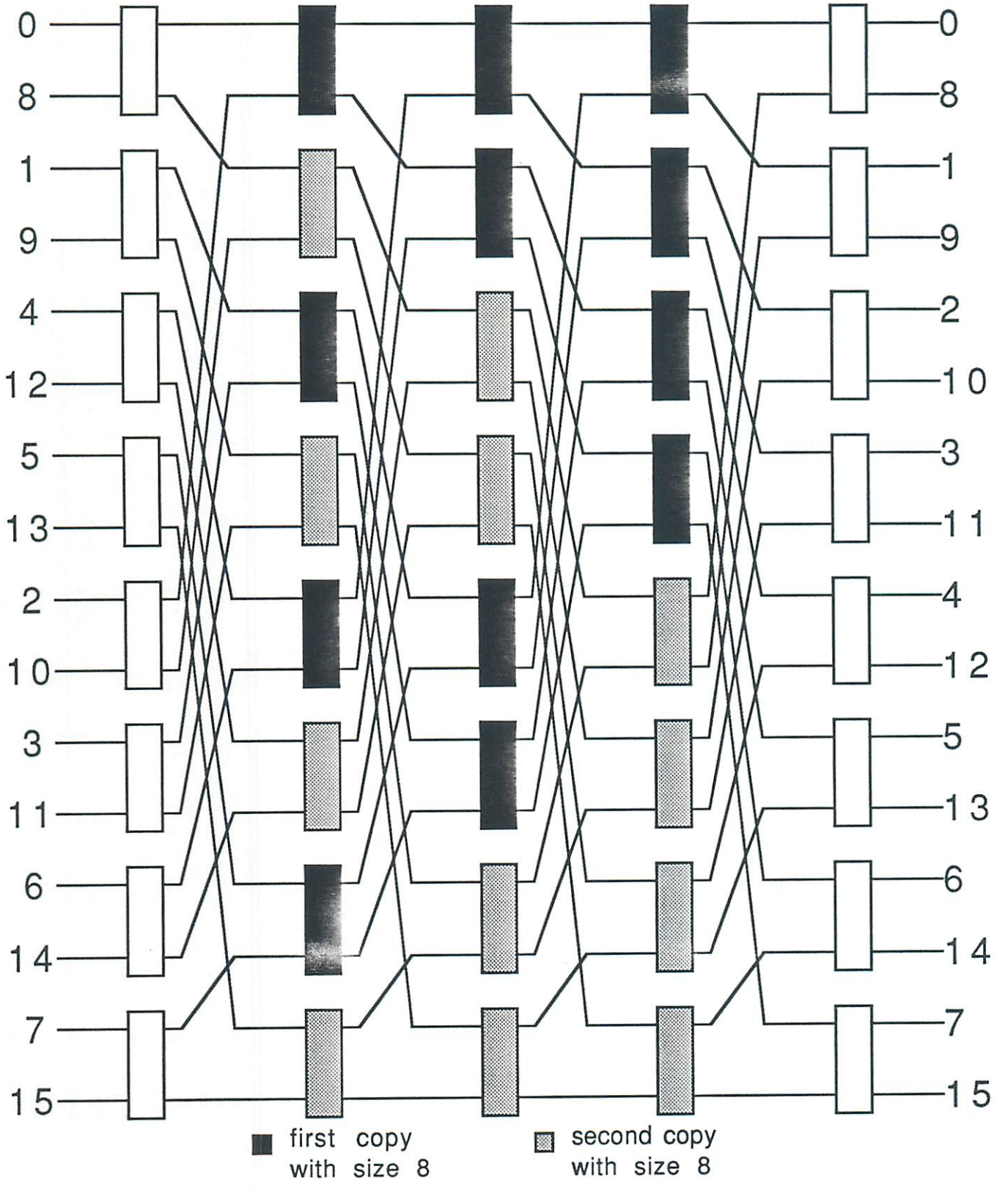

Figure 3. An Extra Stage MIN with size 16

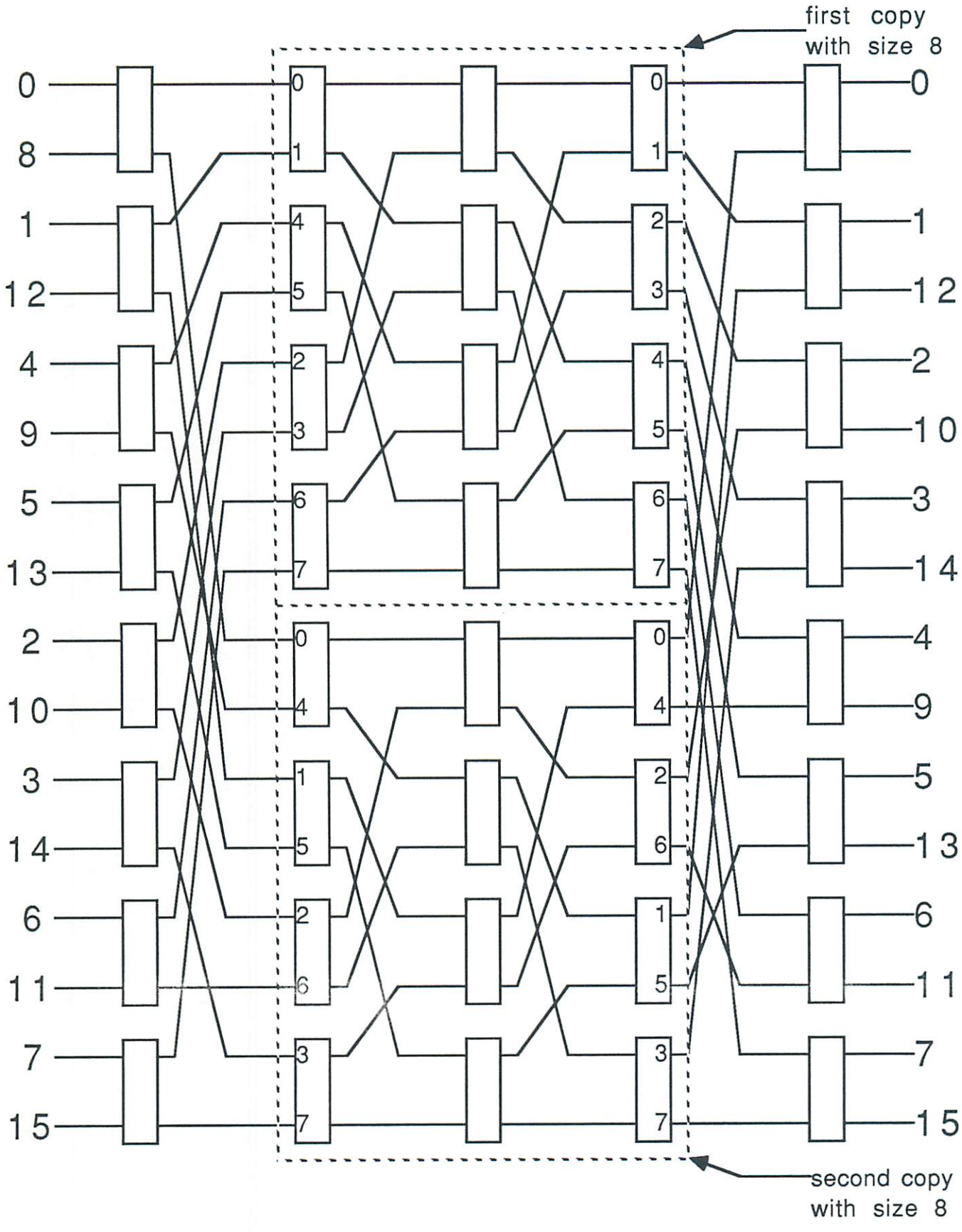

Figure 4. A Dual Extra Stage MIN with size 16

Figure 5. Normalized Bandwidth vs. Failure stage for Two Copy MIN

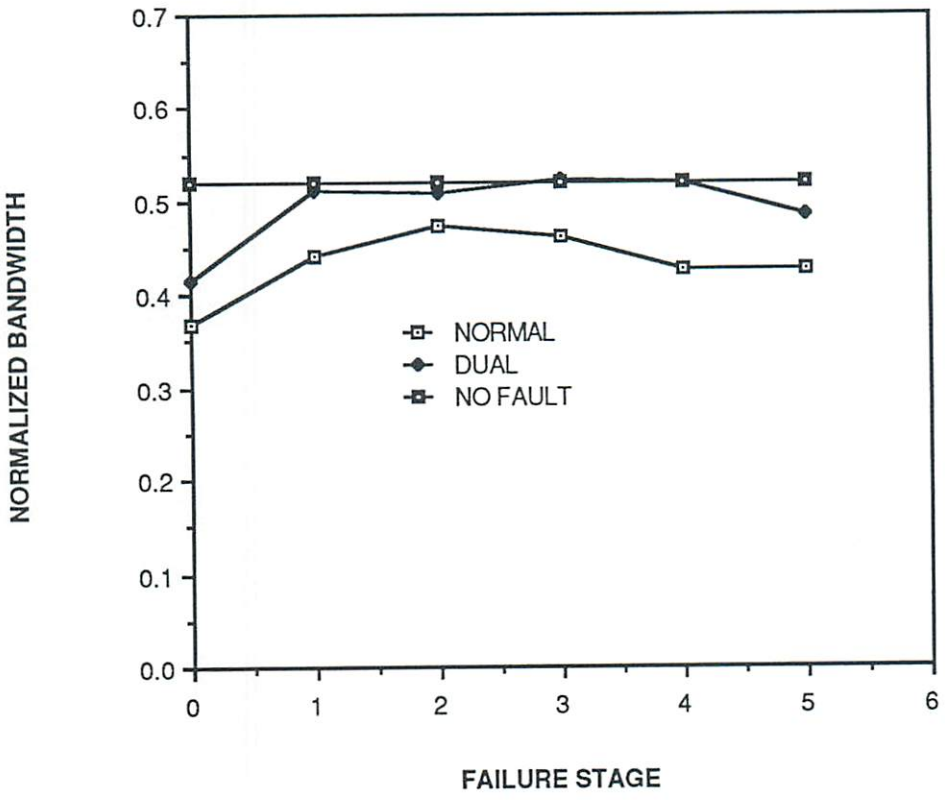

Figure 6. Normalized Bandwidth vs. number of Processors for Two Copy MIN

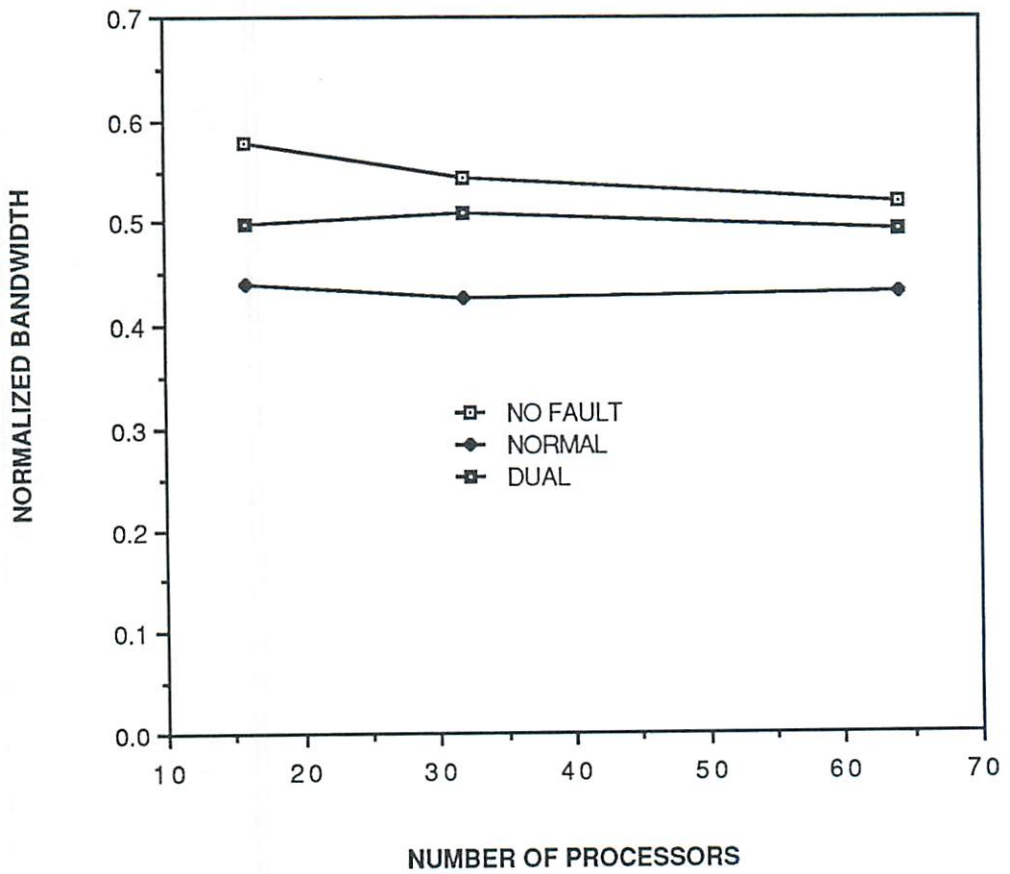

Figure 7. Normalized Bandwith vs. Buffer size for Two Copy MIN

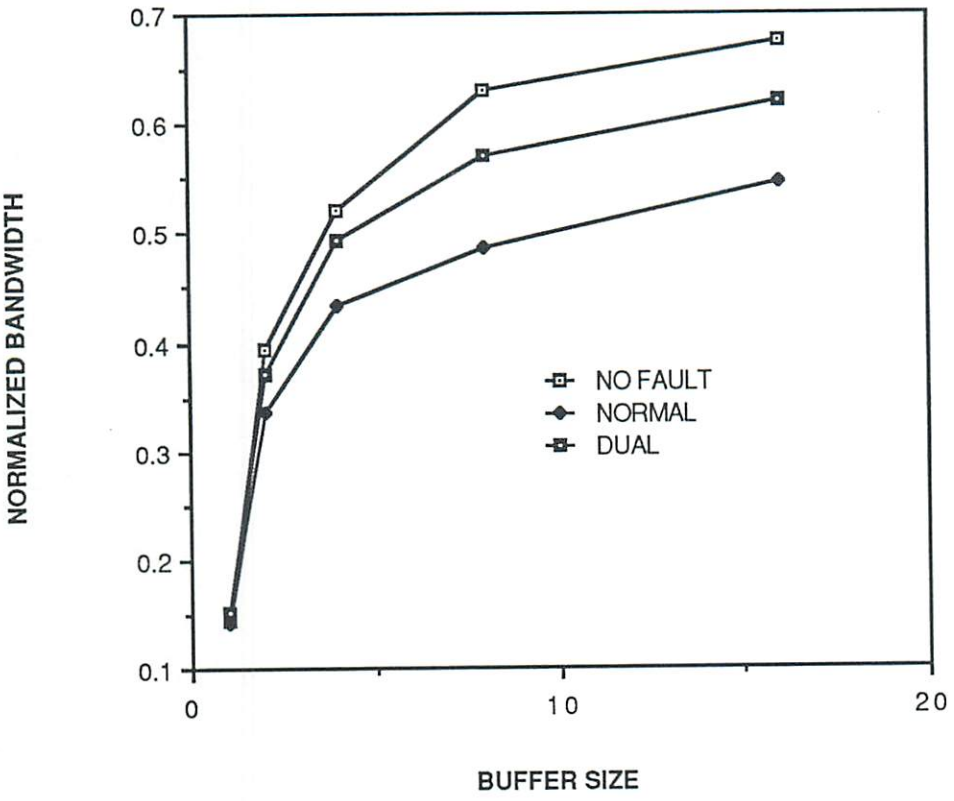

Figure 8. Normalized Bandwidth vs. Failure Stage for Extra Stage MIN

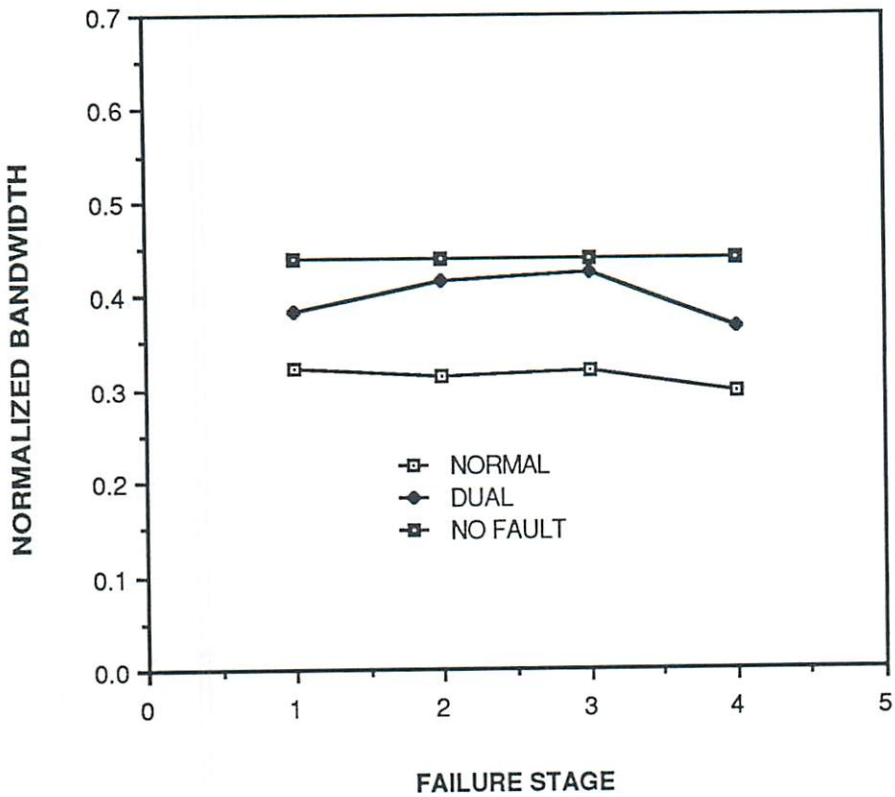

Figure 9. Network Reliability for 5 Stage MIN's

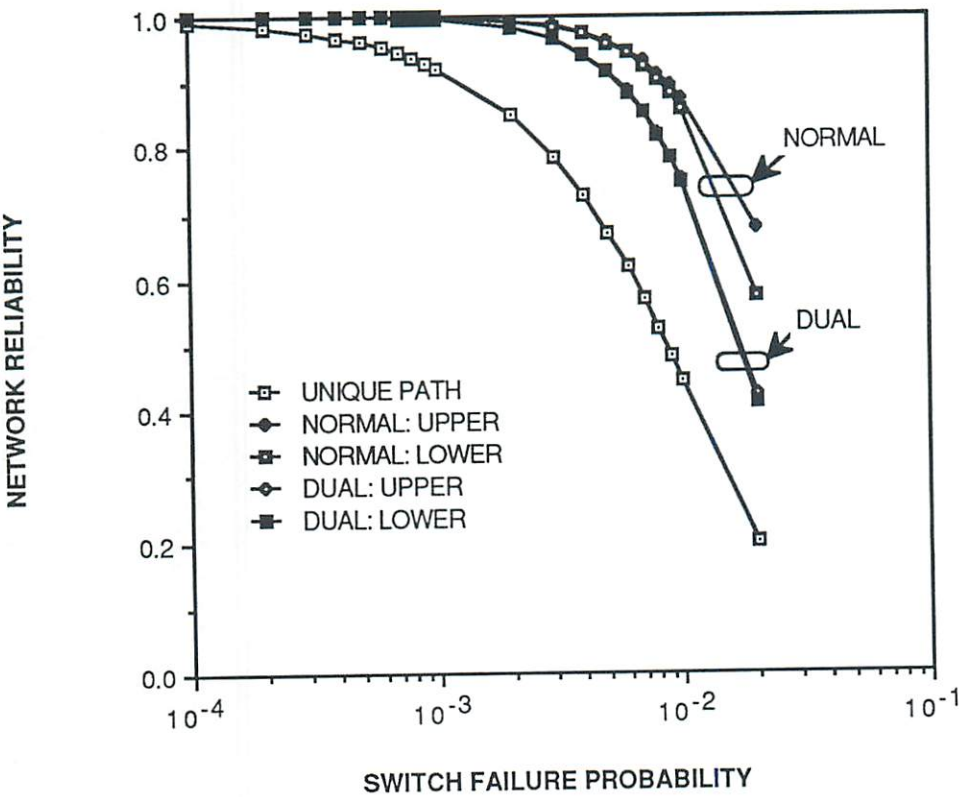

Figure 10. Network Reliability for 10 Stage MIN's

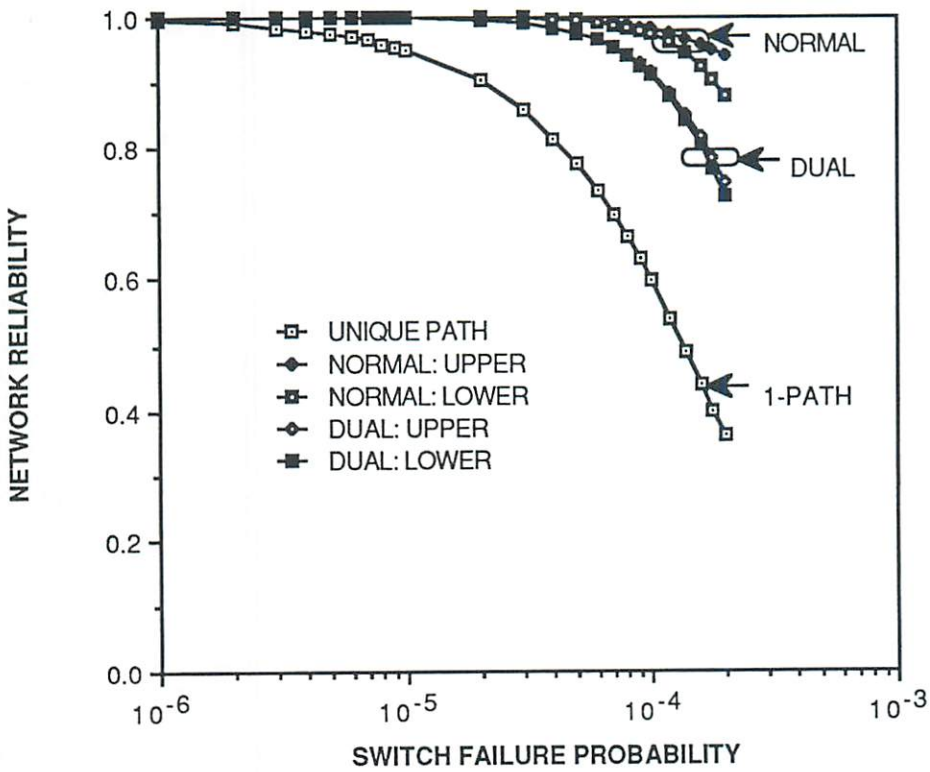

Figure 11. Network Reliability for 10+1 Extra Stage MIN's