Computer Science Department Technical Report
University of California
Los Angeles, CA 90024-1596

THE PERFORMANCE OF A FAULTY MULTISTAGE INTERCONNECTION NETWORK WITH DIVERTING SWITCHES AND BYPASS LINKS

Tomas Lang
Lance Kurisaki

January 1990
CSD-900003

The Performance of a Faulty Multistage Interconnection Network with Diverting Switches and Bypass Links

Tomas Lang and Lance Kurisaki
UCLA Computer Science Department
January 1990

Abstract -- Multistage interconnection networks (MINs) are used in multiprocessor systems to connect processors with other processors or memory modules. Performance studies have indicated that blocking MINs provide low delay for light to moderate traffic loads under uniform traffic conditions when no faults exist in the network. When faults do occur, multipath and multipass schemes allow network operation to continue.

In this paper, we present simulation results which show that overall network performance is significantly affected by the location of a fault, and that the primary influence is the nonuniformity of the traffic distortions produced by the fault. Diverting switches and alternate path networks are shown to give better performance than blocking switches, especially when the input traffic produces additional nonuniform traffic spots. To further improve performance in the presence of faults, we propose modifications to the basic network.

Key words: Multistage networks, fault tolerance, nonuniform traffic, diverting switches, performance analysis.

Introduction

A variety of interconnection networks for large scale parallel processing systems have been proposed. Multistage interconnection networks (MINs) comprise an important subset of these, and are recognized as a cost-effective means to provide communication in multiprocessor systems, as opposed to crossbar switches which have good performance but are prohibitively expensive, with $O(N^2)$ cost (where N is the number of input and output terminals), and the timeshared buses which are inexpensive but have unacceptable throughput when the number of processors connected to them is large [KuRe85]. Performance studies have indicated that under uniform traffic conditions, blocking buffered MINs provide low delay for light to moderate traffic loads [KrSn88]. However, significant degradation occurs when nonuniform traffic spots (NUTS) exist [LaKu88]. In that case, diverting switches and networks with alternate paths provide improved performance.

The basic MINs, such as the Omega network in Figure 1, possess the property of full access, which implies that data from any source node can reach any destination node in a single pass through the network. Moreover, there exists a unique path between each source and destination pair. This unique path property, which facilitates simple and efficient routing algorithms, is undesirable for fault tolerance since the loss of a single switching element disrupts the communication between many source and destination pairs, as shown in Figure 1. The reliability of these networks remains a significant problem.

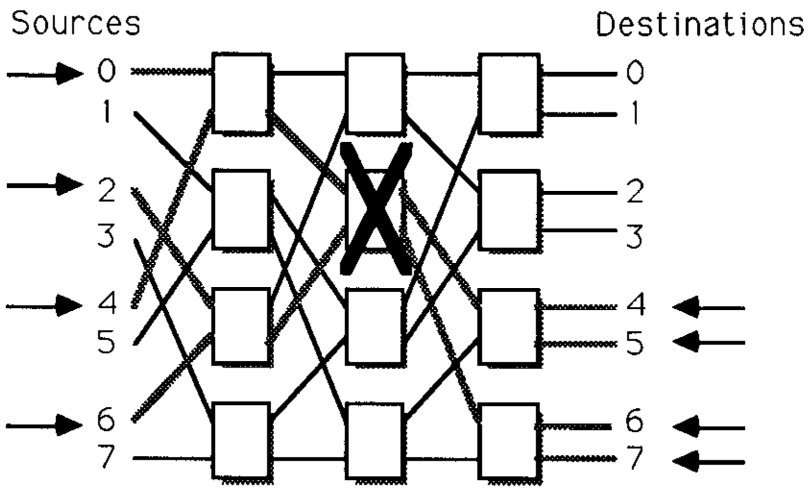


Figure 1. Loss of access caused by a faulty switch.

Techniques developed to deal with faults can be described as multipath, multipass, or a combination of both. Mulitpath schemes augment the basic network with additional links and/or additional switches to provide alternative source-to-destination paths. When a switch or link fault is detected, another path is taken. A survey and comparison of these types of networks can be found in [AdAg87], although no performance measures in the presence of faults are given.

Rerouting or selection of alternate paths in a multipath MIN can be either static or dynamic [KuRe85]. If a multipath MIN allows rerouting to be done only at the source or some fixed points in the network where the alternate paths fork, then the rerouting is static. The Modified Omega network [PaLa83], extra stage cube [SiMc81], and Indra [RaVa84] have only static rerouting capabilities. In contrast, multipath MINs with dynamic rerouting have a fork at every stage of the network. Thus rerouting decisions can be made by the switches at any stage as faulty or congested switches are encountered. Examples of dynamically reroutable networks include the IADM with half links [McSi82], the ACN and MDN [ReKu84].

One particular multipath dynamic rerouting network, called the Augmented Shuffle-Exchange Network (ASEN) and shown in Figure 7, has been studied by [KuRe89]. The performance was analyzed in the presence of link, switch, and loop faults by analytical modeling techniques. According to their model, the location of a fault had little effect on the overall performance of the network, but did have a significant impact on some parts of the system.

The simulation results we present here show that overall network performance **is** affected by fault location. The discrepancy in the two results can be traced to a difference in some basic assumptions of the network behavior. A more detailed discussion is given later.

In a multipass network, packets are allowed to traverse the network a finite number of times when there is no direct source-to-destination path [VaRa88]. When a fault is detected by a predecessor switch, packets are diverted to the wrong output port. Only switches which are predecessor to a fault generate diverted packets -- all other switches block when congestion occurs. Packets which arrive at incorrect destinations make a second pass through the network. This requires that the network have wrapped-around connections so that destination nodes can submit packets. Multipass schemes in general require less additional hardware multipath networks. The performance of multipass networks in the presence of faults, however, has not previously been analyzed.

In this paper, we present simulation results which show that overall network performance is significantly affected by the location of a fault, and that the primary influence is the nonuniformity of the traffic distortions produced by the fault. The degradation in performance is most severe when the fault lies near the destination nodes. Diverting switches and alternate path networks are shown to give better performance than blocking switches, especially when the input traffic produces additional nonuniform traffic spots. To further improve performance in the presence of faults, we propose modifications to the basic network. The input and output stages of the network can be relocated when a fault occurs, effectively changing the location of the fault. This requires additional buses and multiplexers, and a network with cylindrical wrapped-around organization. A new type of alternate path connection, called bypass links, is introduced to help when the fault is near to the destination nodes, and the performance improvement is shown.

Performance Measures

Performance measures for interconnection networks can be either global or local. Global measures attempt to capture the overall performance of the network. One common global performance measure, which we will use, is average delay versus average relative throughput for varying network loads.

Sometimes, however, global measures fail to capture important characteristics of system performance. For example, some processors may receive service well below the average. This is especially significant when processors cooperate on a parallel computation, in which periodic synchronization is required. In such a case, the speed of the computation will be limited by the slowest processor. Therefore, good average performance is not sufficient -- we need to consider the performance of the individual processors. As one such local performance measure, we use the distribution of relative throughput seen by the different processors. To compare the effects of the different switch types, we consider the distributions at a point which produces relatively low average delay (twice the minimum delay) and roughly equivalent throughput. Ideally, we would like a smooth distribution, in which all processors receive the same network service. We will see, however, that faults in the network affect different parts of the network to a varying degree.

The Fault Model

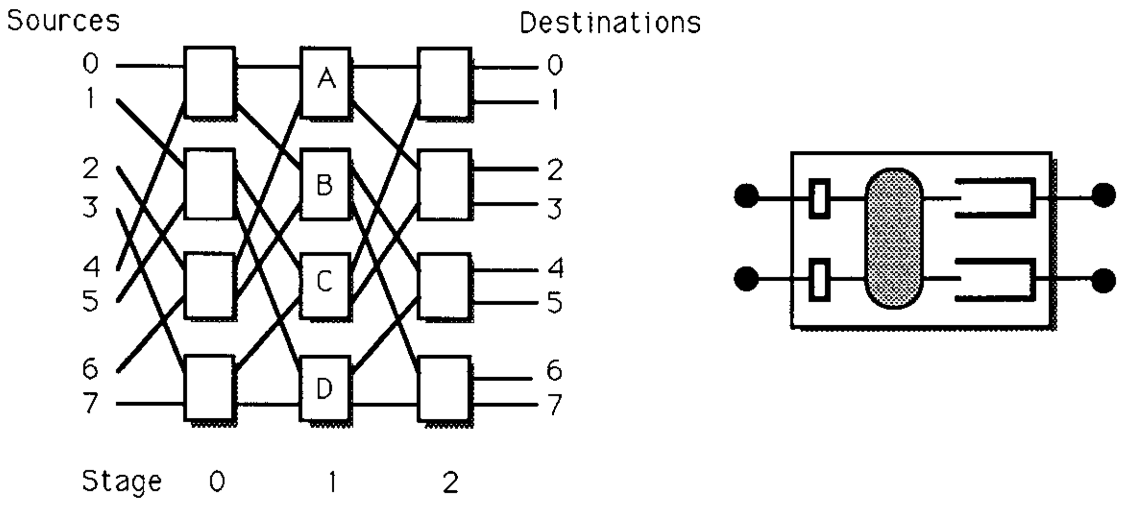
In our fault model, we consider only the failure of an individual switching element. A faulty switch is considered totally unusable and no packets can be routed through it. Faults in individual links are not considered, as switch faults can be used to conservatively approximate link faults by treating the link as part of a faulty switch. We assume that switch elements are able to detect faults in successors by either applying test inputs [FeWu81], or self-checking logic techniques [BrKu89].

We assume that faults are permanent, no provision for repair is made, and only a **single fault** exists in the network. While transient faults have not been simulated, there is no reason why they cannot be integrated into the model, with normal operation resuming if the fault vanishes. By assuming permanent faults, we arrive at a lower bound on the performance.

The *fault tolerance criterion* is the condition that must be met for the network to be said to have tolerated a given fault [AdAg87]. In our case, we use **dynamic full access** under single faults, which is the ability for any input node to communicate with any output node in a finite number of passes through the network.

Multistage Network Structure and Operation

To make this paper relatively self-contained, we now repeat the brief description of the structure and operation of the multistage network, emphasising the assumptions we make in [LaKu88]. A more detailed discussion can be found in [SIEG85]. The type of multistage interconnection network we are considering has $N = 2^n$ inputs (sources) and outputs (destinations), both labeled from 0 to N - 1. It consists of n stages of N/2 2×2 switches, as shown in Figure 2a for N = 8. Switches have input register with FIFO buffers at the outputs (Figure 2b). Details regarding alternative buffer organizations and policies can be found in [LaKu88], and are not considered here. The outputs of stage-i switches are connected to the inputs of stage-(i+1) switches, with the network inputs going to stage-0 switches and the network outputs coming from the stage-(n-1) switches.



Figures 2 a and b. An Omega Network and Switch Element

Several specific multistage networks have been proposed, differing in the interconnection pattern between stages. Since their characteristics, in terms of method of operation and performance, are similar for all these different topologies, we consider here the Omega network [LAWR75], which has been extensively studied in [ChLa81]. In this network, the interconnection pattern between stages is the perfect shuffle permutation, as shown in Figure 2a.

The routing of packets in the network is unique since there is a single path between specific source-destination pairs. The control of routing is done using a destination tag that is associated with each packet. At stage i, the routing depends on the (n-i-1)th bit of the tag (bit 0 is the least significant bit); if the bit has value 0(1), route to the upper(lower) output port of the switch.

The operation of the network is synchronous and pipelined. In its basic form, each switch has one register per output and each cycle one packet is transferred from an output register in a switch of stage i to the corresponding output register of a switch of stage i+1. This implies that the packets are all of the same size. If the packet is large, the above mentioned cycle can be divided into several subcycles and a part of the packet be transferred per subcycle. However, we will not be concerned with this subdivision and will use the packet as the basic unit of transfer and the time of this transfer as the unit of time (one cycle).

Since the output register can receive only one packet per cycle, there is a conflict when both packets entering a switch in a cycle have to be routed to the same output. One solution is to have a buffer for each output and to store the additional packet in such a buffer. Of course, these buffers are finite so it is necessary to have an operational policy when the buffer is full. The basic scheme used is a **blocking** policy in which the predecessor switches do not send packets to a full buffer. To support this policy it is necessary to have signals from a switch to its predecessors indicating that the corresponding input register is full and cannot receive another packet (Figure 3).

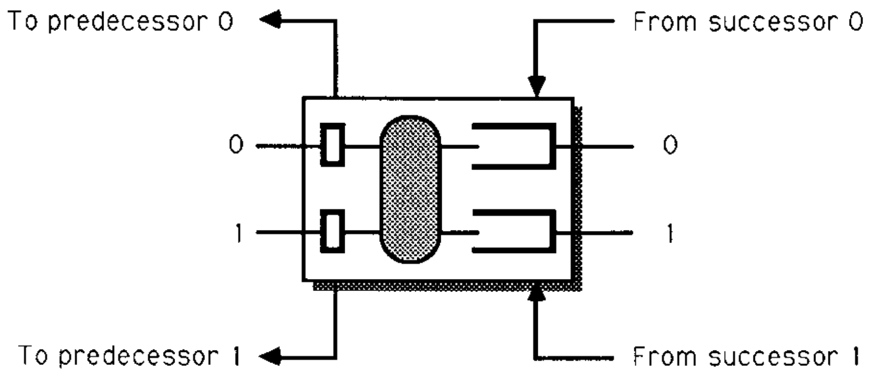


Figure 3. A Blocking Switch with Full Signals

To control contention it is also possible to use **diverting** switches [LaKu88], as shown in Figure 4. In this case, contention is resolved by diverting a conflicting packet to the wrong destination port. Since only one source-destination path exists, the diverted packet will eventually reach an incorrect destination, and must be resent from there. It is important to note that packets are sent to the next stage every cycle (if the source buffer is not empty), so there will be at least one space in each output buffer. Since two buffer spaces are freed by exiting packets and a maximum of two packets can enter each cycle, no blocking occurs and no full signals are necessary. At most one packet will be diverted per switch cycle.

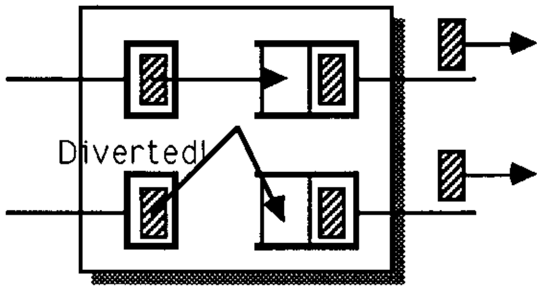
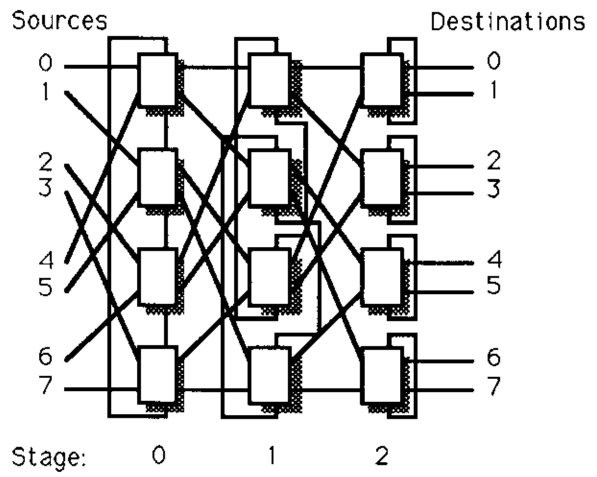


Figure 4. A Diverting Switch

In [LaKu88], it was shown that for traffic patterns that produce NUTS (nonuniform traffic spots), diverting switches produce a larger throughput than blocking switches for a given delay in a fault-free Omega network.

To further reduce the contention it is possible to add intrastage links which produce **alternate paths** for the packets [KuRe85, TzYe85] (Figure 5). The intrastage links connect switches which span the same set of destination nodes. Consequently, overflow packets can be dynamically rerouted through the additional links and are able to reach their desired destination in a single pass through the network. This, of course, requires an augmented 3×3 switch with additional control. We combine this alternate path configuration with diverting control to obtain the best throughput [LaKu88].



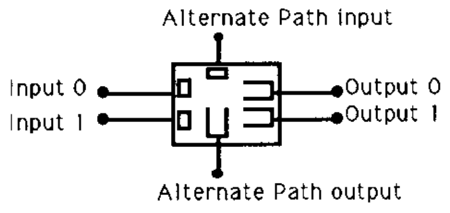


Figure 5. An Alternate Path Network and Augmented Switch

Modifications to accommodate faults

Packets can avoid a fault in the network by diverting and taking an additional pass, by taking an alternate path if one exists, or a combination of both. The basic switch network control schemes are unable to accommodate faults, and must be modified. For example, to avoid the fault in the network with blocking switches, packets can be diverted to the other functional port in the two switches which are immediate predecessors of the fault [VaRa88]. The other switches in the network block as they would normally, and diversions are only generated immediately before the fault. Diverted packets must be resent from their intermediate destination.

In the basic network with diverting switches, no full signals are required since blocking never occurs. When faults exist in the network, however, this is no longer true since the two switches which are immediate predecessors of the fault can accept two packets and send a maximum of one per cycle (Figure 6). To avoid buffer overflow in this situation, blocking signals are necessary. A switch will block an input port if the packet in the corresponding input register cannot be transferred to an output queue in the current cycle. When no fault exists in the network, there will always be space for packets in the output queues, and the switch will not block.

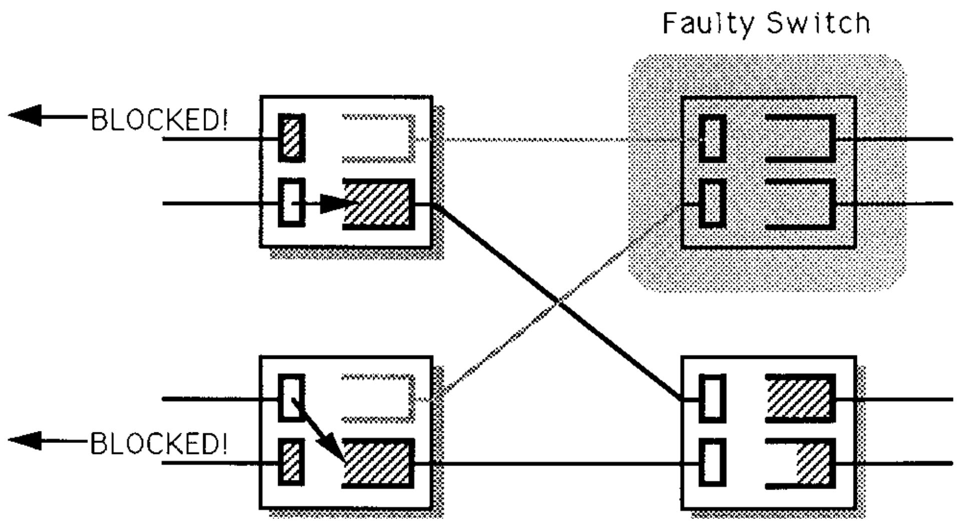


Figure 6. Diverting Switches Require Blocking Signals

The necessary modifications to the alternate path network are the same as for diverting switches. That is, blocking signals are required, and blocking occurs when the corresponding input register cannot be emptied in the current cycle. This results in a combination multipath-multipass network.

A number of algorithms can be used to route packets once they are diverted [LaKu88]. We use a "modified complement" diverting policy, in which a diverted packet is routed to a random output port at every stage until the last, where it is then routed using the bit complement of the source tag. This policy has the advantages of avoiding the original point of diversion in the second pass, and evenly distributing the diverted packets among destination nodes.

Faults in the first and last stages of the network can be handled with the addition of multiplexers and demultiplexers to provide multiple network connections to source and destination nodes. One example of such a network is the ASEN [KuRe89], as shown in Figure 7. Destination nodes have two network connections through the demultiplexers in the last stage. However, the demultiplexers are *coupled* with their predecessor switch, so that a fault in a demultiplexer is considered a fault of a next to last stage switch element.

Similarly, source nodes have two network connections through multiplexers in the first stage. When a failure of a multiplexer occurs, all of the traffic from four sources is routed through the alternate multiplexer and its corresponding switch in the next stage. This has an equivalent effect to a second stage fault in the basic network without input multiplexers.

Consequently, failures in the first and last stages have effects similar to failures in the second and next to last stages, and need not be considered separately.

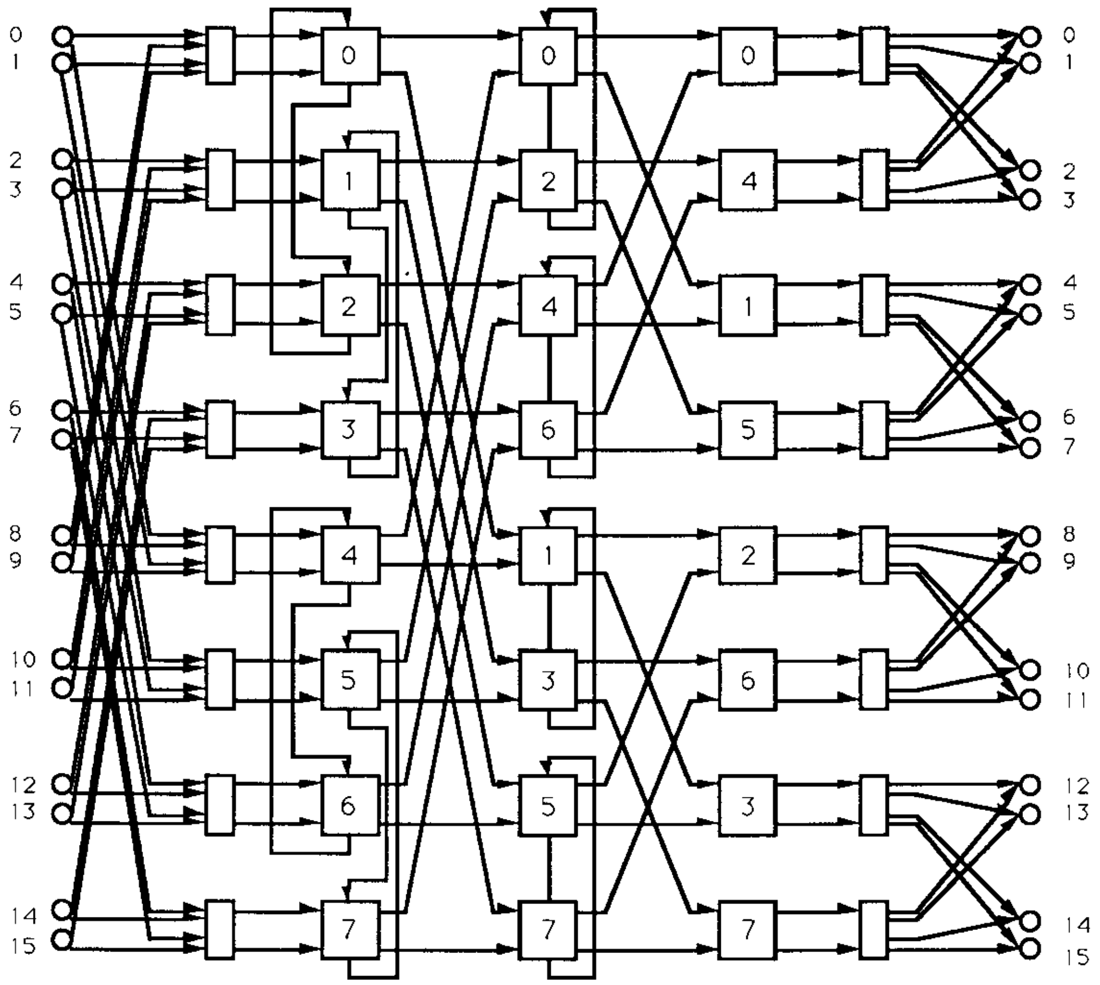


Figure 7. The ASEN (Note: drawn as a Delta network)

Performance Evaluation by Simulation

To study the performance of the network under fault conditions, we performed simulations. For this, we built a network simulator using as a basis SIMON, a general-purpose discrete event multiprocessor simulator developed at the University of Utah [FUJI86].

We evaluate the steady-state behavior of the system, that is, we begin collecting network statistics after a long enough time period so that the transient behavior has passed.

The fundamental parameters for the network are its size and the size of the buffers. We have found that the relative performance of the network remains essentially the same for different values of these parameters. Consequently, we report our results for a network of size 64×64 and buffers of size 2. This buffer size is convenient because it produces a relatively small delay. These parameters are the same as in [LaKu88], where more detail can be found. With 2×2 switches, the minimum delay from input to output will be 6 cycles.

Simulations are performed for networks of three types of switches (blocking, diverting, and alternate paths) with a single switch fault in one stage of the network. As previously mentioned, global network performance is measured by average delay versus average relative throughput for a network load ranging from 10% to 100%. As a local performance measure, we use the distribution of relative throughput for the different processors at an operating point which produces relatively low average delay (twice the minimum) and roughly equivalent throughput.

The performance of the network is measured under uniform and nonuniform traffic patterns. As the nonuniform pattern, we choose the even-to-first odd-to-second (EFOS) traffic pattern, in which the even sources send (uniformly) to the first half of the destinations, and the odd sources send to the second half, as shown in Figure 8. This pattern serves to illustrate a case in which each source accesses a subset of destinations, and has been shown to produce degradation in a blocking network [LaKu88].

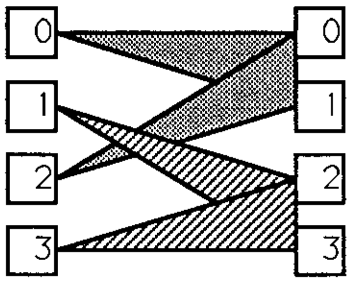


Figure 8. The EFOS Traffic Pattern

Under uniform traffic, a fault in any switch in a particular stage has an equivalent effect on performance. Packets in the EFOS traffic pattern, however, are not evenly spread through the network. For example, in stage 0 (the input stage) all packets from even sources are routed to switch output port 0, while all packets from odd sources are routed to switch output port 1.

Since switches in stage 0 are connected to either two even sources or two odd sources, only one output port of a switch will be used, and only half of the switches in the next stage will carry traffic. Since a fault in a switch which carries little traffic will have a minimal effect on overall performance, we simulate the worst case in which faults in the EFOS traffic network occur in switches which carry a heavy load.

Simulation Results

Figures 9 through 11 show the average delay versus average relative throughput for networks with blocking, diverting, and alternate path switches under uniform and EFOS traffic patterns.

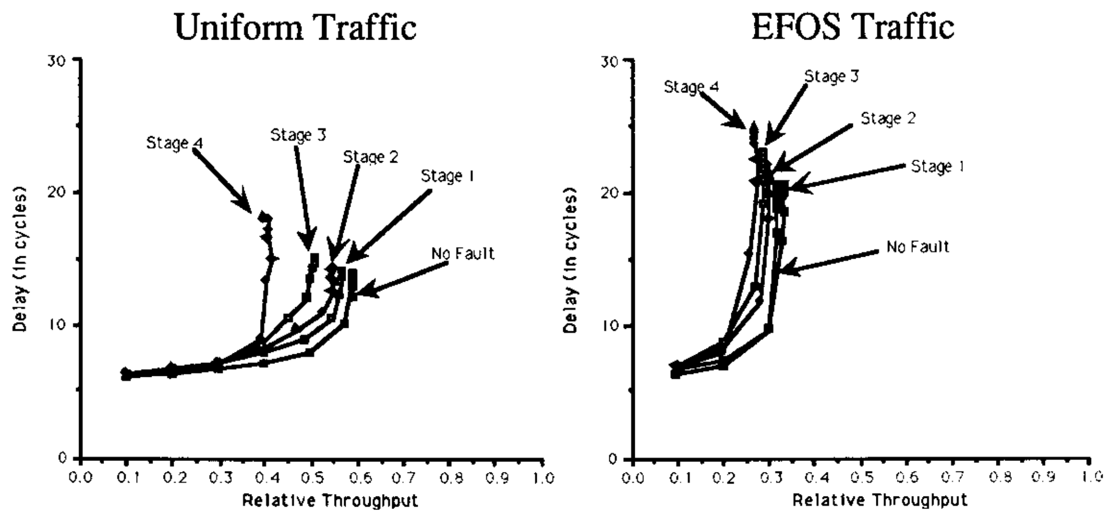


Figure 9. Delay vs. throughput (blocking switches)

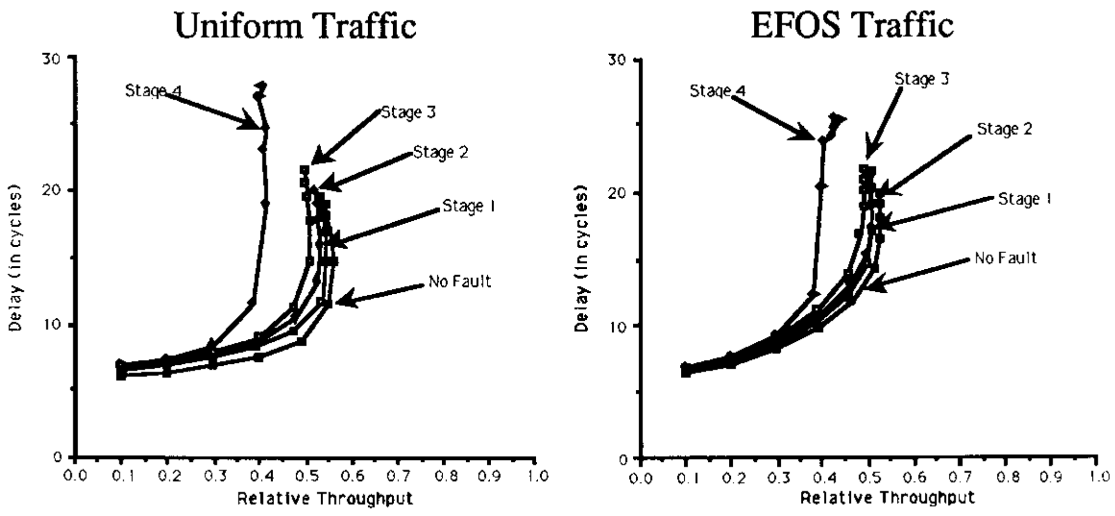


Figure 10. Delay vs. throughput (diverting switches)

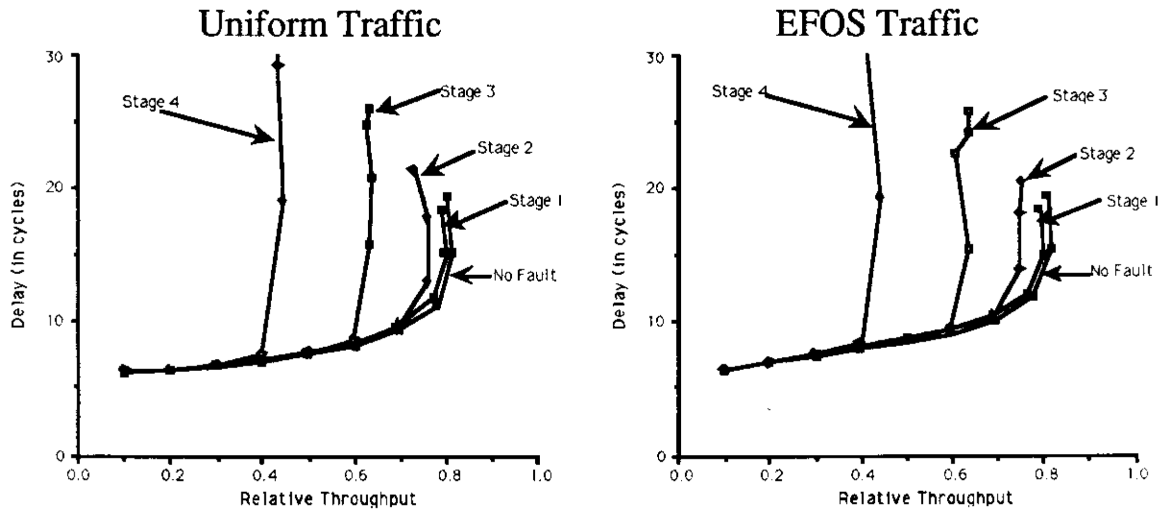


Figure 11. Delay vs. throughput (alternate path network)

The simulation results clearly show that global performance is affected by the location (stage) of a fault. Degradation becomes severe as the fault is moved closer to the destination nodes. This effect is apparent for all three types of switches, and for uniform and nonuniform traffic patterns.

Blocking and diverting switches produce roughly the same average throughput for uniform traffic in the presence of faults. However, the distribution of throughput in the blocking network is less uniform than the distribution in the diverting network. For example, Figure 12 shows the distribution under uniform traffic with a fault in stage 3. The network with diverting switches produces a more desirable distribution.

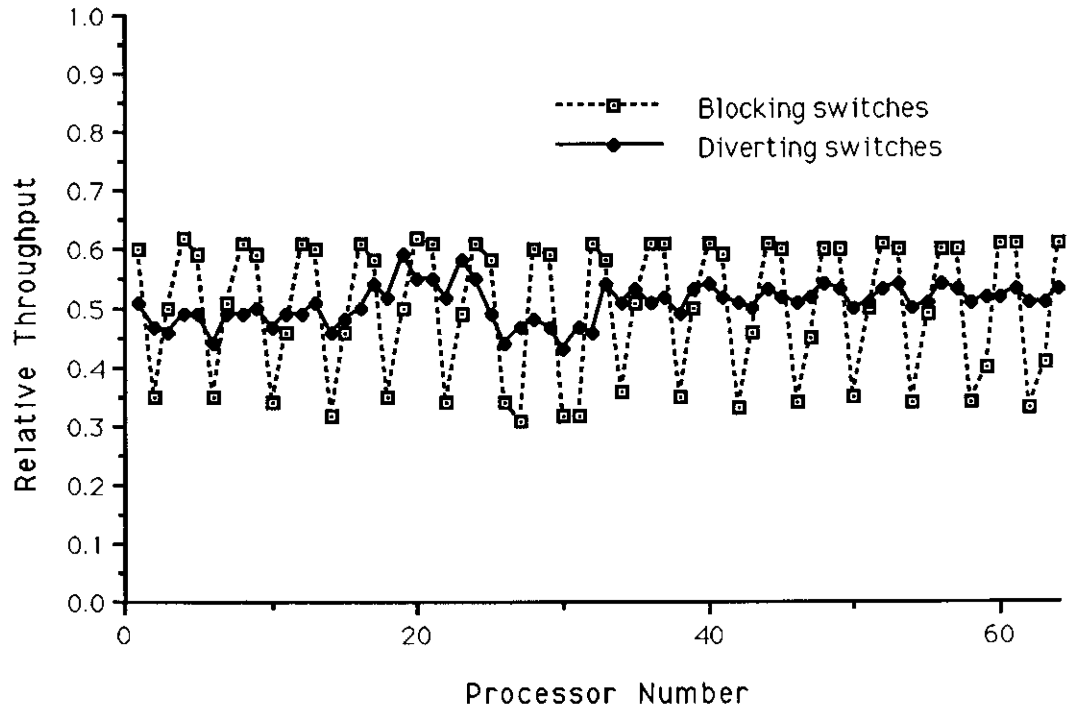


Figure 12. Distribution of throughput for blocking and diverting switches

Under nonuniform EFOS traffic, blocking switches produce lower average performance than diverting switches for all fault locations.

The alternate path network gives significantly better average throughput than the other networks under uniform and nonuniform traffic for most fault locations. When the fault is in stage 4, however, the maximum average throughput rate is reduced to roughly the same level for all switch types. Moreover, it is interesting to note that the alternate path network also shows severe degradation with a fault in stage 3, unlike the networks with blocking and diverting switches.

Analysis of Results

We now analyze the results of the simulations. In particular, we explain the reasons for the dependence of performance on fault location.

When the fault lies near the source nodes, only a few sources will be significantly affected, and therefore there is little impact on global performance. Our simulation results confirm this intuitive hypothesis. However, the severe global degradation that results when the fault lies near the destination nodes is somewhat unexpected, and requires closer investigation.

We will use the following terminology introduced in [ReKu84] and [AGRA83]. All switches in stage i which are on paths to the same set of destination nodes are said to be in a **conjugate subset**. The number of switches in a conjugate subset is 2^{n-i-1}. In Figure 2a, the two conjugate subsets in stage 1 are {A, C} and {B, D}. Two switches in stage i are **output buddies** if their output links are connected to the same switches in stage i + 1; that is, they have the same successor switches. Two switches in stage i are **input buddies** if their input links are connected to the same switches in stage i - 1; that is, they have the same predecessor switches. In Figure 2a, switches A and C are output buddies, while switches A and B are input buddies. A switch has two successors: a **0-successor** connected to its output port 0, and a **1-successor** connected to its output port 1. An analogous relationship exists with the 0- and 1-predecessors of a switch.

A fault in the network degrades performance by two effects: (1) congestion in the tree of predecessors to the fault, and (2) the nonuniformity of the traffic distortions produced by the fault. The first effect is the most obvious one, and is due to the fact that the faulty switch's input buddy must carry the additional traffic previously flowing through the fault, as shown in Figure 13. This overload results in a tree of saturated queues rooted at the input buddy and extending back to a subset of sources. These sources are forced to issue requests at a reduced rate because of the tree saturation.

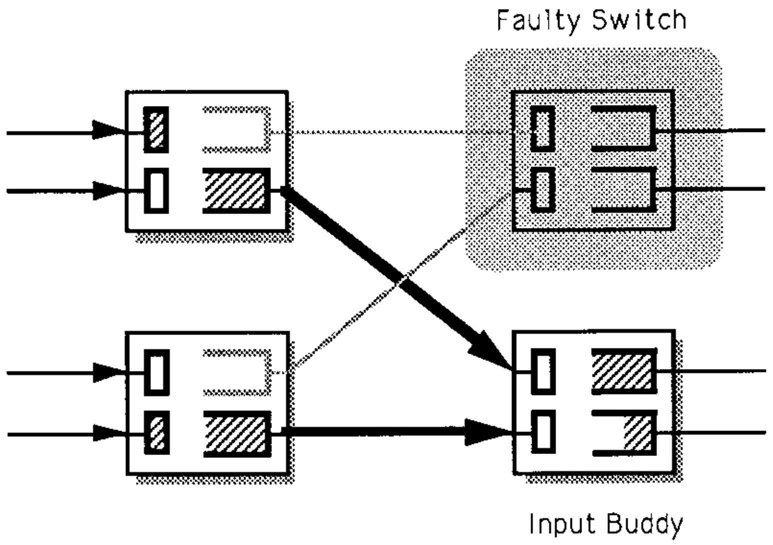


Figure 13. Fault's input buddy must carry additional traffic.

In a network of diverting switches, congestion still exists at the input buddy. As the tree of saturated queues builds, however, packets are diverted away from the congestion earlier than in the blocking network. Tree saturation is reduced, although more packets are diverted. In addition, packets that are diverted earlier will be distributed among more destination nodes.

This effect alone, however, does not account for the global degradation that occurs. When the fault is near the sources, only a few sources are severely affected. When the fault is near the destinations, many sources are affected, but each only to a small degree. In either case, average performance will not be significantly affected.

A fault generates a significant number of diversions in all three types of networks. These diversions are resent from their intermediate destination node, and **must** pass through the conjugate subset, which is now reduced by one, to reach their intended destination. Diverting switches and alternate paths will not help since a diverted packet will again attempt to traverse the conjugate subset on its next pass, and no alternate paths between conjugate subsets exist.

This nonuniformity in the traffic distortions is the dominant effect of a fault, and is most severe when the fault is near the destination nodes. When the fault is in stage n - 2, for example, the conjugate subset contains only two switches, and the remaining non-faulty member must carry **all** of the traffic bound for four destination nodes. Figure 14 shows that a fault in stage 2 of a 16×16 network forces all packets bound for sources 8 through 11 to be routed through the one remaining member of the conjugate subset of the faulty switch. This creates a severe "hot spot" at the internal switch and results in tree saturation. The simulations show that with uniform traffic, all switch types degrade to the same level of average throughput with a fault in stage n - 2. The degradation in earlier stages is less since conjugate subsets are larger.

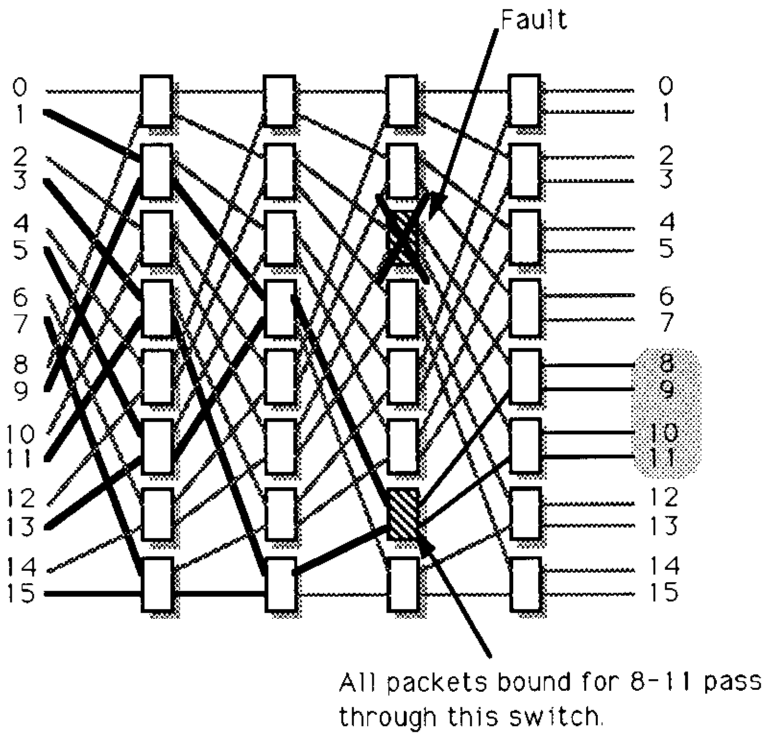


Figure 14. Congestion at the conjugate subset of the fault.

A fault in an alternate path network has the additional effect of eliminating a link in one of the intra-stage loops, which connect switches in the same conjugate subset. As previously noted, the conjugate subset of a faulty switch carries an additional load due to the rerouted traffic. Therefore, losing an alternate path connection in this group has a significant effect. Congestion occurs in the switch whose alternate path output feeds the faulty switch, since it has three (heavily utilized) inputs and only two functioning outputs, as shown in Figure 15. The tree saturation that results explains the relatively severe degradation seen in the alternate path network simulations with a stage 3 fault, which is not present in the blocking and diverting switch networks.

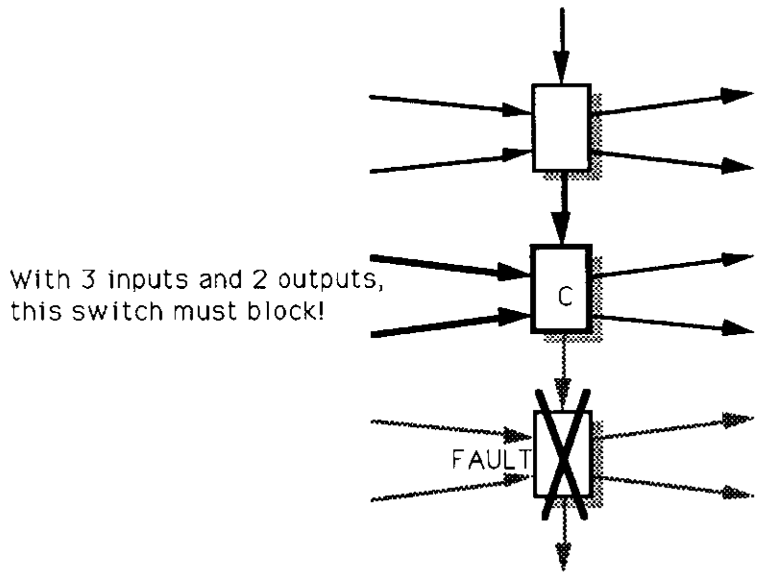


Figure 15. Loss of alternate path link causes congestion.

Previous research [KuRe89] in a similar study concluded that global performance was not significantly affected by the location of a fault in a circuit switched multistage network. However, those results were based on an analytical model in which the discarded traffic's destination tag was (uniformly) regenerated before the packet was resent. We have shown that the nonuniformity of the traffic distortions produced by a fault has a significant effect on performance, and cannot be disregarded.

Some Solutions

An effective fault tolerant system must not only be able to continue operation in the presence of faults, but must also provide an adequate level of service. These simulations have shown that severe local and global degradation can occur in both multipass and multipath MINs when faults exist. We now propose some solutions to improve performance in such cases.

Mobile Input/Output Stages

The network in our simulations suffered severe degradation when the fault is near the destination nodes. One way to avoid this situation is to move the fault when it occurs in an undesirable location. This can be accomplished in the Omega network because each stage is interconnected by the same perfect shuffle permutation. Figure 16 shows the top view of the cylindrical network. When a fault occurs in an undesirable stage, we can move the input and output stages across the cylinder with busses and multiplexers to effectively change the location of the fault. In addition, faults in the input and output stages can also be handled in this manner.

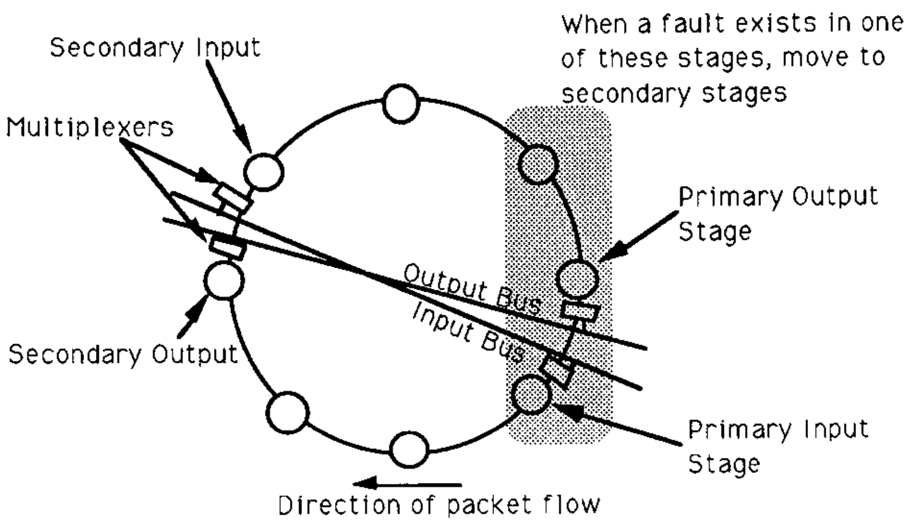


Figure 16. Move the Input/Output stages when faults occur.

In the alternate path network, the intrastage connection pattern is unique in each stage. Moving the input and output stages in this case would require additional reconfiguration of those links. However, limiting the input and output stages to two possible locations means that only two configurations of intrastage links are needed, and network complexity might not be severely increased.

Bypass Links

The intrastage links in the last stage of the alternate path network do not provide any alternate paths, and are essentially not utilized. By changing the connection pattern of those links, we can use them to "correct" packets diverted due to a fault in the next to last stage (n - 2). We call links in such a configuration "bypass" links.

The connection pattern of the bypass links can be described in the following manner and is shown in Figure 17 for an 8×8 network: For every switch S in stage n - 2 and its input buddy B, connect the 0-successor of S and the 0-successor of B in a closed loop, and the 1-successor of S and the 1-successor of B in another closed loop. Figure 16 shows an example in which source 1 sends a packet to destination 3. A fault exists in stage 1 of the network. Although the packet is diverted in the first stage, it is still able to reach the intended destination in a single pass by using the bypass links.

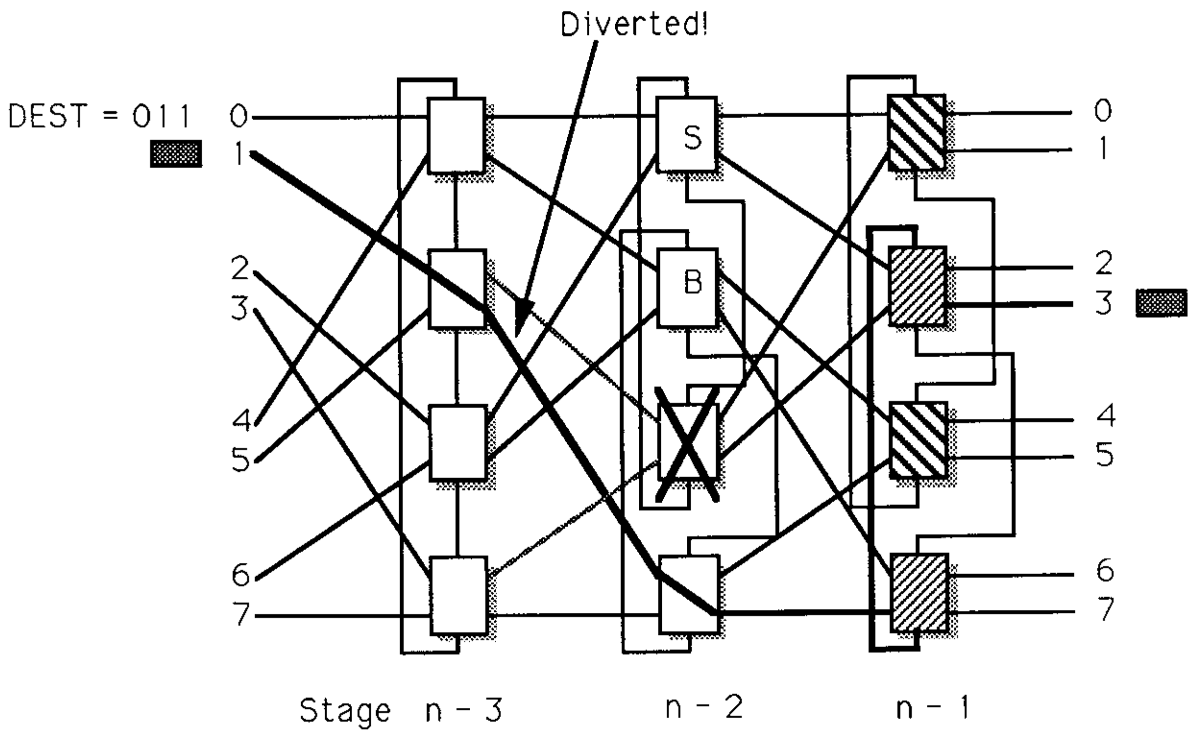


Figure 17. An alternate path network with bypass links in the last stage.

In order for the bypass links to be effective, we use "direct diverting", in which the original destination tag is used for routing after a diversion occurs [LaKu88].

A packet is eligible to use the bypass output link in stage n -1 under the following conditions:

1. The packet has been diverted only once.
2. The diversion occurred in stage n - 3.
3. The packet did not come via the bypass input port.

When those conditions are met, the bypass links will correct the diverted packet and allow it to arrive at its intended destination node in one pass. The simulation results in Figure 18 show that bypass links significantly improve the performance of the worst case fault (stage 4).

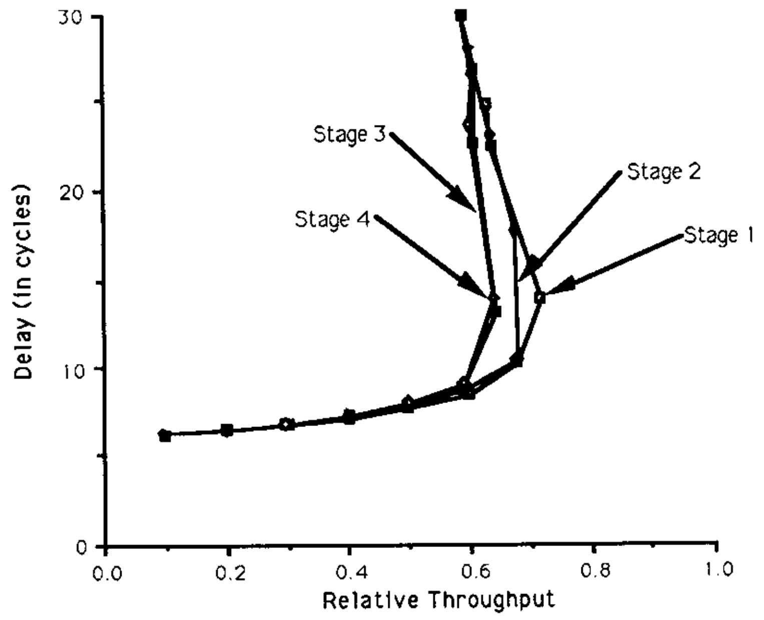


Figure 18. Performance of a network with bypass links (uniform traffic)

Conclusions

Simulation results show that the location of a fault significantly affects the local and global performance of a MIN with blocking, diverting, or alternate path switches under uniform and nonuniform traffic patterns. Global performance (average throughput versus average delay) is substantially degraded in all cases when the fault is near the destination nodes.

Diverting switches produce better average performance than blocking switches when the input traffic is nonuniform. They also provide improved local performance, as measured by the distribution of relative throughput of the sources, under both uniform and nonuniform traffic conditions.

The primary cause of the degradation is the nonuniformity of the traffic distortions produced by a fault. When the fault is in the next to last stage, diverting and alternate paths cannot ease the load on the conjugate subset of the fault, and all three network types degrade to the same level.

We propose to overcome this by moving the input and output stages of the network to change the effective location of the fault when it occurs in a critical stage. In addition, a new type of alternate path connection in the last stage, called bypass links, has been shown to be effective for the worst case faults in the next to last stage.

Bibliography

[AdAg87] Adams, G. B., Agrawal, D. P., Siegel, H. J., "Fault-Tolerant Multistage Interconnection Networks," IEEE Computer, June 1987, pp. 14-27.

[AGRA83] Agrawal, D. P., "Graph Theoretical Analysis and Design of Multistage Interconnection Networks," IEEE Trans. Computers, July 1983, pp. 637-648.

[BrKu89] Brunner, B., Kumar, V. P., Petitpierre, C., "A Self-Checking Switching Element for a Fault-Tolerant Multistage Interconnection Network," submitted for publication.

[ChLa81] Chen, P., Lawrie, D., Yew, P., "Interconnection Networks Using Shuffles," Computer, December 1981, pp. 55-64.

[FeWu81] Feng, T. Y., Wu, C., "Fault Diagnosis for a Class of Multistage Interconnection Networks," IEEE Trans. Computers, Vol C-30, October 1981, pp. 743-758.

[KrSn88] Kruskal, C. P., Snir, M., Weiss, A., "The Distribution of Waiting Times in Clocked Multistage Interconnection Networks," IEEE Trans. on Computers, November 1988, pp. 1337-1352.

[KuRe85] Kumar, V. P., Reddy, S. M., "Design and Analysis of Fault-Tolerant Multistage Interconnection Networks with Low Link Complexity," 12th Annual Computer Architecture Conference, 1985, pp. 376-386.

[KuRe87] Kumar, V. P., Reddy, S. M., "Augment Shuffle-Exchange Multistage Interconnection Networks," IEEE Computer, June 1987, pp. 30-40.

[KuRe89] Kumar, V. P., Reibman, A. L., "Failure Dependent Performance Analysis of a Fault-Tolerant Multistage Interconnection Network", IEEE Transactions on Computers, December 1989, pp. 1703-1713.

[LaKu88] Lang, T., Kurisaki, L., "Nonuniform Traffic Spots (NUTS) in Multistage Interconnection Networks," Proceedings of the 1988 International Conference on Parallel Processing, August 1988, pp. 191-195.

[PaLa83] Padmanabhan, K., Lawrie, D. H., "A Class of Redundant Path Multistage Interconnection Network," IEEE Trans. Computers, C-32, December 1983, pp. 1099-1108.

[PfNo85] Pfister, G. F., Norton, V. A., "Hot Spot Contention and Combining in Multistage Interconnection Networks," IEEE Transactions on Computers, October 1985, pp. 943-948.

[RaVa84] Raghavendra, C. S., Varma, A., "INDRA: A Class of Interconnection Networks with Redundant Paths," 1984 Real Time Systems Symposium, December 1984.

[ReKu84] Reddy, S. M., Kumar, V. P., "On Fault-Tolerant Multistage Interconnection Networks," 1984 International Conference on Parallel Processing, Computer Society Press, 1984, pp. 155-164.

[SiMc81] Siegel, H. J., McMillen, R. J., "The Multistage Cube: A Versatile Interconnection Network," Computer, Vol. 14, No. 12, December 1981, pp. 65-76.

[SIEG85] Siegel, H. J., "Interconnection Networks for Large-Scale Parallel Processing, Theory and Case Studies," Lexington Books, 1985.

[VaRa89] Varma, A., Raghavendra, C. S., Fault-Tolerant Routing in Multistage Interconnection Networks," IEEE Transactions on Computers, March 1989, pp. 385-393.

[VaRa88] Varma, A., Rathi, B. D., "A Fault-Tolerant Routing Scheme for Unique-Path Multistage Interconnection Networks," IBM Research Report RC 13441 (#60126), Watson Research Center, 1/21/88.