Design of Fault-Tolerant Interconnection Networks with Minimal Rerouting Hops

最少額外鏈結數的重新繞路容錯互聯網路之設計

Ching-Wen Chen
Department of Computer Science and Information Engineering
Chaoyang University of Technology
chingwen @mail.cyut.edu.tw

Shih-Chang Fu
Department of Computer Science and Information Engineering
Chaoyang University of Technology
s9167605@mail.cyut.edu.tw

摘要

在互聯網路上對於容錯的設計，經常使用互斥路徑以及動態重新繞路兩種方法。如果在封包傳送之前，已經知道錯誤所在，我們就可以選擇另一條互斥路徑以避開錯誤的交換器或錯誤的鏈結。否則，我們也可以以動態重新繞路的方法來動態的避開錯誤的交換器或錯誤的鏈結。在這篇論文之中，我們著眼於動態重新繞路的問題，並呈現出一個可以容錯的網路。除了能保證一次容錯的能力之外，以重新繞路的方法來找出一條可取代路徑時，也將重新繞路時所增加額外鏈結的數目最小化。此外，為了能容忍多重的錯誤，我們分析重新繞路的方法，並提出另一個能在任一階層發現交換器或鏈結錯誤或忙碌時，都能提供一個可替代路徑的網路。經由與一些重要的動態重新繞路網路之間的比較，我們的成果在容錯能力、重新繞路額外增加的鏈結，以及硬體成本各方面都有比較好的表現。

關鍵詞：Gamma 互聯網路(GIN)，容錯，動態重新繞路，互斥路徑，回溯

Abstract

In the design of fault-tolerant on interconnection networks, disjoint paths and dynamic rerouting are often used to tolerate faults. If a fault is known before the packet is sent, one of the different disjoint paths can be chosen to tolerate the faulty switches or links. Otherwise, a dynamic rerouting scheme is needed to tolerate the faulty switches or links dynamically. In this paper, we address the dynamic rerouting problem and present a network to tolerate faults.

Besides guaranteeing one fault tolerance, the rerouting method minimizes the number of extra links traversed in finding such an alternative path. Furthermore, to tolerate multiple faults, we analysis the dynamic rerouting method and then propose another network to provide an alternative path at each stage when a switch or link is faulty or busy. By the result of comparisons among some important dynamic rerouting networks, our work performs well in fault-tolerant capability, rerouting hops, and hardware cost.

Keywords: Gamma interconnection network (GIN), fault-tolerant, dynamic rerouting, disjoint paths, backtracking

1. Introduction

Interconnection networks are critical to parallel systems because their performance has great impact on system latency and throughput. Multistage interconnection networks (MINs) are well suited to communications among tightly coupled system components, and offer a good balance between cost and performance. In addition, for complex systems, assuring high reliability is a significant task. Thus fault-tolerance is crucial for MINs serving the communication needs of large-scale multi-processor systems [1].

To make MINs fault-tolerant, there are two methods used. One topological method is to provide disjoint paths between any source and destination pair. On the other hand, one routing method is to correct the routing tag in the switch to change the routing path. We call the latter one dynamic rerouting. If the fault is known before the packet is sent, disjoint paths are useful for avoiding the faulty element. Otherwise, the dynamic rerouting method is needed.

To enhance the fault-tolerance capability, 3x3 switches have been used as basic building blocks [1]. For examples, the gamma interconnection network (GIN) [2], augmented data manipulator (ADM), and inverse augmented data manipulator (IADM) [3], Extra Stage Gamma Network [4], CGIN [5], composite banyan [6], B-network [7], [8], PCGIN [9] and Enhanced IADM [3] are in that category. In Gamma networks, there are multiple paths between any source and destination pair except when the source and destination are the same. Extra Stage Gamma Network, CGIN, composite banyan and PCGIN improve the fault-tolerant capability of GIN to provide disjoint paths to tolerate one fault, but B-Network and Enhanced IADM provide dynamic rerouting. B-Network cannot guarantee one fault tolerance. Enhanced IADM modifies IADM by adding 2 links to each stage at stages 1 to n-1. Although Enhanced IADM can tolerate one fault with dynamic rerouting, it requires 5x5's crossbar switch elements.

In this paper, we address the dynamic rerouting problem to present a minimal rerouting hops network, which provides the capability of one fault-tolerance and traverses a minimal number of extra links to find an alternative path. In our design, the network provides two selections to the destination at each stage. According to our selection algorithm, there are two selections at next stage, too. If a faulty link or switch is encountered, the other selection (link) can be taken to the destination. Besides providing the fault-tolerant interconnection networks with minimal rerouting hops, we also discuss how to use the network in heavy traffic environment. That is, we modify the network can tolerate one fault at each stage with minimal rerouting hops. Finally, we make comparisons among several important dynamic rerouting networks.

The remainder of this paper is organized as follows. In Section 2, we introduce the Gamma interconnection network (GIN). In Section 3, we present our design of fault-tolerant interconnection networks with minimal rerouting hops. In Section 4, we discuss the advanced network that can tolerate one fault at each stage. In Section 5, we compare our methods and those using in the dynamic re-routable networks in terms of hardware cost, extra rerouting hop count, and one fault-tolerance. Finally, Section 6 concludes this work.

2. Gamma Interconnection Network

In this section, we introduce the topology and the properties in Gamma networks.

2.1 Topology

A GIN of size N=2^n consists of n+1 stages labeled from 0 to n, and each stage involves N switches [2]. Basically, switches of sizes 1x3 and 3x1 are coupled with the first and last stages respectively. Moreover, each switch located at intermediate stages is a 3x3 crossbar. And each j-th switch number j at stage i has three output links connecting to switches at stage (i+1) based on the plus-minus-2^i function, that is, the j-th switch at stage i has three output links to switches [(j – 2^i) mod N], j, and [(j + 2^i) mod N] at the subsequent stage. Figure 1 illustrates a GIN network with size 8.

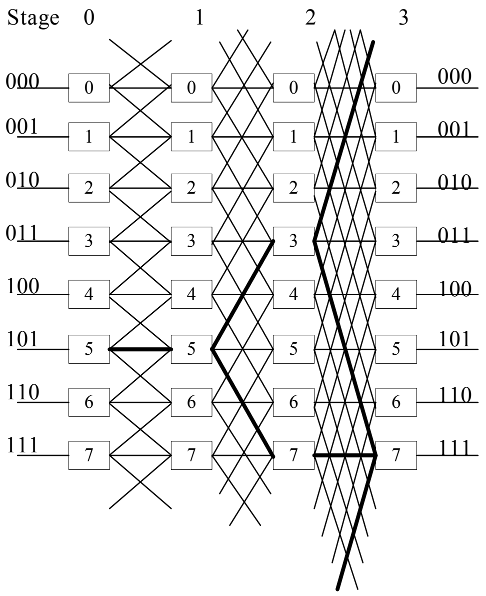

Figure 1. Gamma Interconnection Network with N=8 and its three paths between the source S=5 and the destination T=7

2.2 Multiple paths

In GIN, an n-digit tag determines the path connecting the source to its destination. Each tag digit can be 1, 0, or $\bar{1}$. An n-digit tag D represents the difference between a destination T and a source S, i.e., D = T – S (mod N) = ($d_0d_1...d_{n-1}$), where d_0 means the least signification bit. Digit d_i is used at stage i in such a way that the lower connection is taken when d_i is equal to 1, the straight connection is taken when d_i is 0, or the upper connection is taken when d_i = $\bar{1}$. The GIN makes use of the binary fully redundant number system to represent each tag. A non-zero tag D has multiple representations, that is, there are multiple paths between a source S and a destination T if $S \neq T$. For example, when N=8, a source node S is 5, and a destination node T is 7, the tag D can be 010 or $0\bar{1}1$, or $0\bar{1}\bar{1}$, and the three paths are shown in Figure 1.

2.3 Problems

Gamma networks provide distance tag routing and multiple paths, but there is a lack of mechanism to guarantee one fault-tolerant. There exists only single path in Gamma networks when the source and the destination are the same. In addition, although Gamma networks provide multiple paths when a source is not equal to a destination, these paths cannot guarantee one fault-tolerant.

3. The fault-tolerant interconnection networks with minimal rerouting hops

In this study, we assume that the switch connected to the faulty link or switch has the faulty information. Besides, the switch has the ability to generate rerouting tags by modifying the routing tag. In the following, in Section 3.1, the topology of the fault-tolerant interconnection networks with minimal rerouting hops is presented. In Section 3.2, the routing and rerouting methods are described. Moreover, we prove that the network can tolerate one fault and traverses the minimal number of links when a faulty element is encountered.

3.1 Topology

The fault-tolerant interconnection network with minimal rerouting hops, of size N=2n, consists of n+1 stages labeled from 0 to n and each stage involves N switches. The network is the same as GIN except adding one more link for each switch at stage 0. The added link in the switch j of stage 0 connects to the switch (j-2 mod N) of stage 1. The switches at the first two stages are of size 1x4 and 4x3 and the switches at final stage are of size 3x1. The size of other switches is 3x3. Figure 2 shows the network with size N equal to 8.

3.2 Routing and Dynamic Rerouting

The routing algorithms in the network are presented in two conditions, routing without faults and rerouting after encountering a fault. In this section, the routing algorithms and fault-tolerant capability are demonstrated.

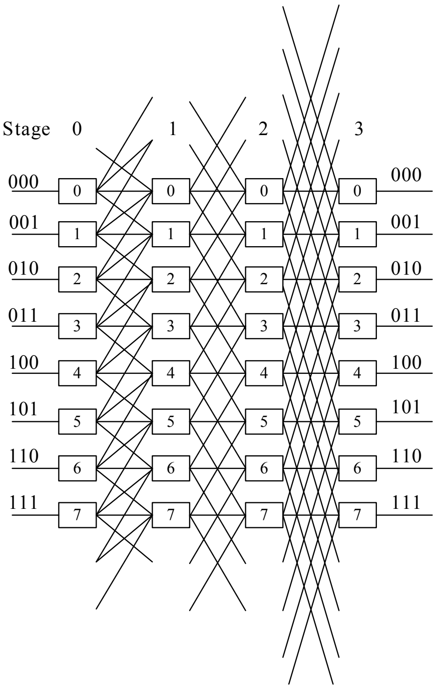

Figure 2. The fault-tolerant interconnection networks with minimal rerouting hops with size N=8

3.2.1 Routing without faults

The routing method uses distance tag routing. Because there are four conditions at the stage 0, two bits are necessary to send a packet to stage 1. At first, the distance D between the source and the destination is computed and is represented to n+1 bits, $(d_{00}d_{01}d_1...d_{n-1})$, by Algorithm 1. The tag generated by Algorithm 1 is the form of 1 and $\bar{1}$ only except the first two bits $d_{00}d_{01}$. At stage 0, the four conditions (00, 01, 10, 11) corresponding to the four links are shown in Figure 3. About other stages, when d_i is 1 (0), the downward non-straight (straight) link at the stage i is taken to the stage i+1 for $i \geqq 1$. Example 1 illustrates the routing path when the source S=1 and the destination T=4.

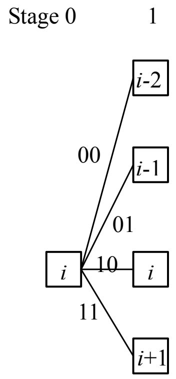

Figure 3. The four outgoing links and the routing bits from stage 0 to stage 1

Algorithm 1: Generating the routing tag by giving *S* and *T*

```
Input the source $S$ and destination $T$
Output the routing tag $d_{00}d_{01}d_1...d_{n-1}$
Begin
/* $t_0t_1...t_{n-1}$ mean the binary representation of the distance between the destination and the source */
Compute the distance of $(T-S)$ Mod $N$ $(=t_0t_1...t_{n-1})$
Let $D = d_{00}d_{01}d_1...d_{n-1}$
If ($t_0$=1) {
If($t_1$=1) $d_{00}d_{01}$=11;
Else If($t_1$=0) $d_{00}d_{01}$=01;
}
Else If($t_0$=0){
If($t_1$=1) $d_{00}d_{01}$=10;
Else If($t_1$=0) $d_{00}d_{01}$=00;
}
Set $d_1$=1;
While (Check $t_i$ from $t_1$ to $t_{n-1}$) {
  If($t_i$ = 0)
Set $d_{i-1}d_i$ to $\bar{1}1$;
        Else
Set $d_i$ to 1
     }
Output $d_{00}d_{01}d_1...d_{n-1}$;
End                                          □
```

Example 1: If the source S=1, the destination T=4, the routing condition is shown in Figure 4.

Solution:
S=1 and T=4, so $t_0t_1t_2$=110 (t_0 is the least signification bit)

By Algorithm 1, we get $d_{00}d_{01}d_1d_2 = 11\bar{1}1$.
The first two bits 11 are used to route a packet to switch 2 of stage 1.

From stage 1 to stage 3, the rest two bits $\bar{1}1$ are used at stage 1 and 2 to route the packet to switch 0 of stage 2 and then arrives at the destination by a downward non-straight link. □

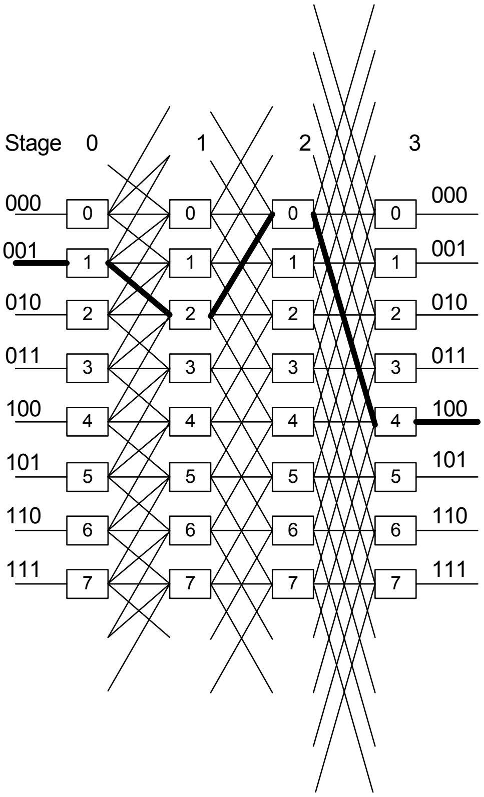

Figure 4. The path from the source S=1 and the destination T=4 in the fault-tolerant interconnection networks with minimal rerouting hops.

Theorem 1: The routing tag generated by Algorithm 1 can route packets to the destination.

Proof: Because the original distance tag $t_0t_1...t_{n-1}$ can route a packet to the destination, we will check the distance between the routing path generated by Algorithm 1 and the original path to prove the routing path can route packets to the destination.

Let the source be S, the destination be T and the routing tag generated by Algorithm 1, be $d_{00}d_{01}d_1...d_{n-1}$. We let the path generated by Algorithm 1 be path *NP* and the original path is *OP*.

From stage 0 to stage 1, the switch number at stage 1 of the path *NP* is (S-2/S-1/S/S+1) mod N if the original bits t_0t_1=00/10/01/11. Therefore, d_1 is set to 1. If the original routing tag $t_0t_1...t_{n-1}$ is applied and t_0t_1=00/10/01/11, the vertical distance of *OP* from stage 0 to stage 2 is 0/1/2/3. In contract, the vertical distance of *NP* from stage 0 to stage 1 is –2/-1/0/1.

If t_2 is 1, d_1d_2 is 11 by Algorithm 1. Otherwise, d_1d_2 is $\bar{1}1$. For the previous case, the vertical downward distance of *NP* from stage 0 to stage 2 is 0/1/2/3. The *OP* and *NP* traverse the same switch at stage 2. For the other case, $d_1d_2 = \bar{1}1$.

Because $\bar{1}$ is applied at stage 1 to stage 2, the vertical downward distance of *NP* from stage 0 to stage 2 is -4/-3/-2/-1. The vertical distance between *OP* and *NP* at stage 2 is 4/4/4/4. As a result, the distance between *NP* and *OP* at stage 2 is 0/0/0/0 ($2^2/2^2/2^2/2^2$) when t_2 is 1(0).

If t_3 to t_{i-1} are all 0s but t_i is 1, the routing tag $d_2d_{3...}d_{i-1}$ by Algorithm 1 is $\bar{1}\,\bar{1}$...1. The distance between *OP* and *NP* at stage *i*-1 is $2^{i-1}/2^{i-1}/2^{i-1}/2^{i-1}$. In addition, from stage *i*-1 to stage *i*, the bit of d_{i-1} is applied, and then the distance between the two paths at stage *i* is 0/0/0/0 because t_{i-1} is 0 and d_{i-1} is 1.

At final stage, if t_{n-1} is 1, the two paths reach the same switch at stage *n*. If t_{n-1} is 0, the distance between the two paths at stage *n*-1 is $2^{n-1}/2^{n-1}/2^{n-1}/2^{n-1}$. In such a case, the vertical distance at stage *n* is 0/0/0/0 because $d_{n-1}=\bar{1}$ is applied and t_{n-1} is 0. The vertical distance between *NP* and *OP* is $(2^{n-1}+2^{n-1})$ Mod 2^n.

As a result, the routing tag generated by Algorithm 1 can route packets from source to the destination. □

The path generated by Algorithm 1 can arrive at the destination by only non-straight links from stage 1 to stage *n*. If a non-straight link is taken, the other non-straight link can also route packets to the destination because the both non-straight links at the same switches are redundant. The implication is that when a non-straight link is traversed and encounters a faulty link or switch, the other non-straight link at the same switch can be taken to the next stage. In next section, we discuss the dynamic rerouting condition when meeting faults.

3.2.2 Dynamic Rerouting with Faults

In this section, we present the rerouting ability when a faulty element is encountered. Because the two non-straight links at the same switch are redundant from stage 1 to stage *n*-1, the routing tag must be re-computed by the switch if the other non-straight link is applied. In Algorithm 2, the method of generating rerouting tag is described.

Algorithm 2: Generating rerouting tag in the switch at the stage *i*

```
Input: Original routing tag $D= d_{00}d_{01}d_1...d_{n-1}$
Output: Rerouting tag $R= r_{00}r_{01}r_1...r_{n-1}$
Begin
  If ($i$=0)
{
     If($d_{00}d_{01}$=00)  { $r_{00}r_{01}$=10; $r_1...r_{n-1}$= ($d_1...d_{n-1}$)-($1_10_2...0_{n-1}$);}
     Else If($d_{00}d_{01}$=01)  { $r_{00}r_{01}$=11; $r_1...r_{n-1}$= ($d_1...d_{n-1}$)-($1_10_2...0_{n-1}$);}
     Else If($d_{00}d_{01}$=10)  { $r_{00}r_{01}$=00; $r_1...r_{n-1}$= ($d_1...d_{n-1}$)+($1_10_2...0_{n-1}$);}
     Else  { $r_{00}r_{01}$=01; $r_1...r_{n-1}$= ($d_1...d_{n-1}$)+($1_10_2...0_{n-1}$);}
}
  Else
  {
     If ($d_i$ = 1)  $r_{00}r_{01}r_1...r_{n-1}$= ($d_{00}d_{01}d_1...\bar{1}_i d_{i+1}...d_{n-1}$)+($0_{00}0_{01}0_1...1_{i+1}...0_{n-1}$);
     Else  $r_{00}r_{01}r_1...r_{n-1}$= ($d_{00}d_{01}d_1...1_i d_{i+1}...d_{n-1}$)-($0_{00}0_{01}0_1...1_{i+1}...0_{n-1}$);
     }
End
```

The operator for + and – is described as follows:

	0	1	$\bar{1}$
+1	1	0 If $i\neq n$-1, next bit+1.	0
-1	$\bar{1}$	0	0 If $i\neq n$-1, next bit -1.

□

Algorithm 1 provides a path that takes only non-straight links from stage 1 to stage *n*-1 and Algorithm 2 generates the rerouting tag when a fault occurs. By Algorithms 1 and 2, a packet can tolerate one fault by dynamic rerouting to the destination. We give an example to illustrate this case and then we prove that the rerouting tag generated by Algorithm 2 can deliver a packet to the destination.

Example 2: This example as shown in Figure 2 illustrates the routing and rerouting path when the source *S*=1, the destination *T*=4 and the downward non-straight link connecting the switch number 2 of stage 1 to the switch number 0 of stage 2 is faulty (or the switch number 0 at stage 2 is faulty).

Solution: By Algorithm 1, we get *D*=11$\bar{1}$1.

Because the downward non-straight link connecting the switch number 2 of stage 1 to the switch number 0 of stage 2 is faulty (or the switch number 0 at stage 2 is faulty), the switch at stage 1 modifies the routing tag to 1110 and then the packet takes the downward non-straight link at stage 1 to the switch number 4 at stage 2 because d_1=1.

By rerouting tag, the straight link at stage 2 is taken to the destination because d_2=0. □

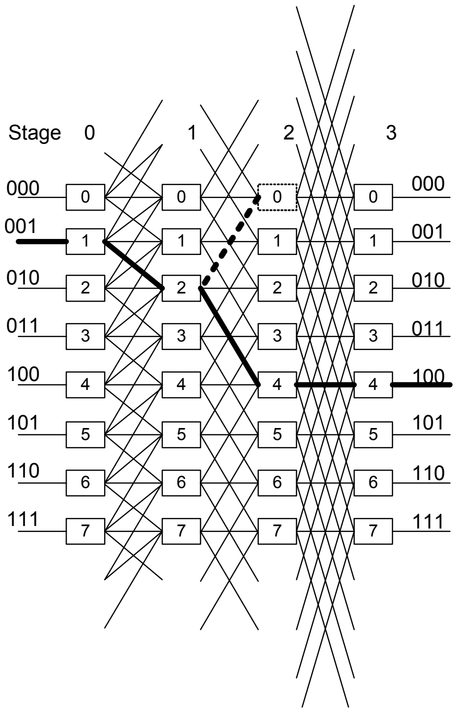

Figure 5. The dynamic rerouting condition from the source S=1 and the destination T=4 in the fault-tolerant interconnection networks with minimal rerouting hops. The dash line or block means the faulty element.

Lemma 1: The rerouting tag generated by Algorithm 2 can make the packet arrive at the destination.

Proof: By Theorem 1, the routing tag can route a packet to the destination. The switch modifies the rerouting tag by changing the routing bit from $1/\bar{1}$ to $\bar{1}/1$ and then adding/substracting 2^{i+1} to the routing tag if the packet meets the faulty element at stage i. We assume that the packet encounters a faulty element from stage i to stage i+1 and d_i is $1/\bar{1}$. Therefore, the alternative path takes the other non-straight link to the stage i+1 and the vertical distance between the original path and the new path at stage i+1 is $\pm 2^{i+1}(=2^i+2^i)$. In Algorithm 2, the distance 2^{i+1} is added or subtracted to generate the rerouting tag. Moreover, the adding/subtracting 2^{i+1} will not affect d_0 to d_i. That is, the different bits between routing tag and rerouting tag only may be in d_{i+1} to d_{n-1}. The vertical distance between the alternative path at stage i+1 and the destination is $r_{i+1}2^{i+1}$ $r_{i+2}2^{i+2}$ $...r_{n-1}2^{n-1}$ where $r_0r_1...r_{n-1}$ is the rerouting tag. By the definition of distance tag routing, the packet can arrive at the destination by applying r_{i+1} r_{i+2} $...r_{n-1}$ from stage i+1 to stage n-1. As a result, the packet can arrive at the destination. □

Theorem 2: The fault-tolerant interconnection networks with minimal rerouting hops can tolerate one faulty switch or link through dynamic rerouting.

Proof: By Theorem 1, a packet can be routed to destination using non-straight links only, and by Lemma 1, if a packet encounters a faulty switch or link between stages 0 to n, the rerouting method can make the packet find the alternative link right away. Hence, the fault-tolerant interconnection networks with minimal rerouting hops can tolerate one fault by dynamic rerouting.

□

Lemma 2: The dynamic rerouting method traverses the minimal number of links to find an alternative path.

Proof: The assumption is that the switch has the fault information if the connected switches or links are faulty. According to the result of Lemma 1, if a faulty element is encountered, the other alternative link at the same switch can be taken to the next stage right now. The dynamic rerouting method traverses the minimal number of links to find an alternative path. □

4. Discussion

In this section, we discuss the fault-tolerant interconnection networks with minimal rerouting hops and how to improve the fault-tolerant capability in such design. Although we have proved that the network can guarantee one fault-tolerance by dynamic rerouting, we investigate whether the network can be used in a heavy traffic environment in advanced. That is, if the network can provide an alternative path at each stage, a packet can take the other path to avoid meeting a busy switch or link.

When a non-straight link is taken to the next stage to the destination, the other non-straight link can also deliver the packet to the destination. By choosing the proper one of the two non-straight links to next stage, there are still two non-straight links at the next stage can be taken to the destination. Algorithm 1 presents the choosing method of the two non-straight links.

When a packet meets a busy (or faulty) element at stage i+1, the switch at stage i takes the alternative link to stage i+1 to avoid meeting the busy element and then the straight link at stage i+1 is taken to stage i+2. As a result, if the packet meets a busy element at stag i+2 again, there is no alternative path. In order to solve this problem, one more link can be added to each switch from stage 1 to stage n-1 like switches at stage 0. The extra link connects the switch j to (j-2^{i+1}) mod N switch between stage i and stage

i+1 shown at the Figure 6. In such a design, the switch has two choices for distance tag bit 1 or 0. If a packet meets a busy or faulty element, the other alternative link can be taken to the next stage. Figure 6 shows the routing condition that the two switches at stage 1 and stage 2 are faulty and the source is 2 and the destination is 2.

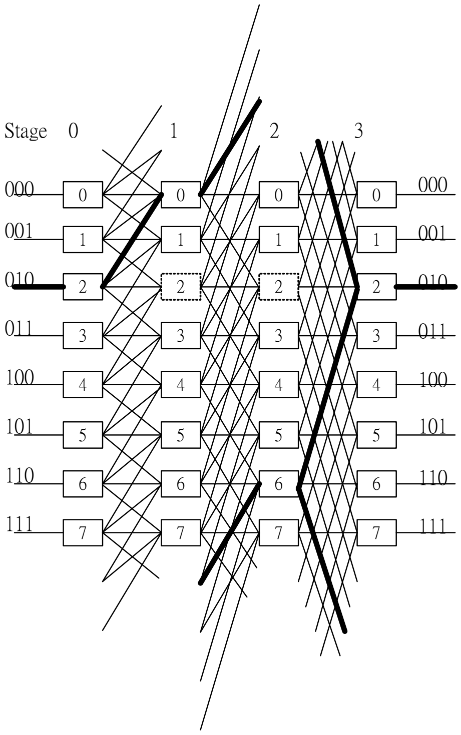

Figure 6. Adding one more link to each switch from stage 1 to stage i-1 to tolerate one faulty (or busy) element at each stage.

5. Comparison

In this section, we compare the hardware cost, fault-tolerance capability and rerouting penalty in terms of number of links of rerouting hops at stage i of some fault tolerant dynamic rerouting methods. These methods include our method, and those used in B-network and Enhanced IADM. Table 1 shows the characteristics of these networks.

In terms of hardware cost, Enhanced IADM is higher than the others because it uses 5x5 switch elements. In our design, the hardware cost is a little high than Gamma network and B-network because one more link is added in the switches at stage 0. For fault-tolerance capability, B-network and Gamma network cannot guarantee one fault tolerance. In contract, our design and Enhanced IADM provide one fault-tolerant capability. If a packet encounters a fault, B-network takes 2 extra links to find an alternative path to next stage if the alternative path exists but the others take 0 links rerouting overhead.

In general, our dynamic rerouting network performs well in fault tolerance capability, hardware cost and zero rerouting overheads in finding an alternative path to the next stage.

6. Conclusion

In this paper, we present the design of fault-tolerant interconnection networks with minimal rerouting hops. Instead of disjoint paths, we use the dynamic rerouting method to tolerate faults. To guarantee one fault-tolerance, Algorithm 1 shows a path that can have two choices at each stage for routing a packet to the next stage. If a faulty element is encountered, the alternative link is taken to the next stage right now. As a result, the network is minimal rerouting penalty.

In order to make the network useful in a heavy traffic environment, we modify the network to have the capability to tolerate one fault at each stage. However, the new network will cost high in hardware cost if 0-rerouting hops are necessary.

Finally, after comparing some dynamic rerouting networks, our design presents a good result in one fault-tolerance guarantee and minimal rerouting hops.

Table 1. The comparison of our network with B-Network and Enhanced IADM.

Network or Design Methods	**Single-Fault Tolerant**	**Hardware Cost (Total switch crossing points)**	**Rerouting Penalty**
Gamma network	No	$9N\log N$ - $2N$	0 or N.A.
B-network	No	$9N\log N$ - $2N$	2 links or N.A.
Enhanced IADM	Yes	$25N\log N$-$27N$	0 link
Our network	Yes	$9N\log N + N$	0 link

7. Acknowledgements

This research was supported by the National Science Council, Taiwan, Republic of China, under Grant No. NSC-91-2218-E-324-006-.

8. References

[1] Adams G. B. III, Agrawal D. P., and Siegel H. J., "A Survey and Comparison of fault-tolerant Multistage Interconnection Networks," IEEE Transactions on Computer Vol. 20, No. 6, June 1987, pp. 14-27.

[2] Parker D.S. and Raghavendra C.S., "The Gamma Network," IEEE Transactions on Computers, Vol. C-33, April 1984, pp.367-373.

[3] 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.

[4] Yoon K. and Hegazy W., "The Extra Stage Gamma Network," IEEE Transactions on Computers, Vol. 37, No. 11, November 1988, pp.1445-1450.

[5] Chuang P. J. "CGIN: A Fault Tolerant Modified Gamma Interconnection Network," IEEE Transactions on Parallel and Distributed Systems. Vol. 7, No. 12, December 1996, pp. 1301-1306.

[6] Seo S. W. and Feng T. Y., "The Composite Banyan Network," IEEE Transactions on Parallel and Distributed Systems, Vol. 6, No. 10, October 1995, pp.1043-1054

[7] Lee K. Y. and Yoon H. " The B-network: A Multistage Interconnection Network with Backward Links," IEEE Transactions on Computers Vol. 39, No. 7, July 1990, pp. 966-969.

[8] Lau F. C. M and Poon W. C., "Throughput Analysis of B-Networks" IEEE Transaction on Computers, Vol. 47 No. 47 April 1998, pp.482-485.

[9] Chen C. W., Lu N. P., Chen T. F. and Chung C. P., "Fault-tolerant gamma interconnection networks by chaining," IEE Proceedings of Computers and Digital Techniques, vol. 147 no. 2 pp. 75-80, March 2000.