ELSEVIER

The Journal of Systems and Software 58 (2001) 23–31

The Journal of Systems and Software

www.elsevier.com/locate/jss

Fault-tolerant gamma interconnection network without backtracking

Ching-Wen Chen [a,*], Chung-Ping Chung [b]

[a] Department of Computer Science and Information Engineering, National Chiao Tung University, 1001 Ta Hsueh Road, Hsinchu 30050, Taiwan ROC

[b] Computer and Communications Research Laboratories, Industrial Technology Research Institute, Chutung, Hsinchu 310, Taiwan ROC

Received 7 December 1999; received in revised form 1 May 2000; accepted 22 June 2000

Abstract

A no-backtracking gamma interconnection network (NBGIN) is a modified gamma interconnection network (GIN) that can tolerate one switch or link fault without backward packet transmission. Basically, this can be achieved by providing two alternative paths at all intermediate nodes during the progression of routing through carefully designed algorithm. In this work, a destination tag routing named as *NB* function is designed. With this *NB* function, a packet can find an alternate path as soon as it encounters a switch or link fault in NBGIN. The major advantage of NBGIN over gamma-induced networks is that there is no rerouting penalty under a fault. In addition, NBGIN processes the characteristics of one-fault tolerance, destination tag routing, and dynamic rerouting. Furthermore, the routing and rerouting path can be simply determined by the destination tag only instead of being computed with the routing tag. © 2001 Elsevier Science Inc. All rights reserved.

Keywords: Gamma interconnection network; Distance tag routing; Destination tag routing; Disjoint paths; Backtracking

1. Introduction

Interconnection networks are critical to parallel systems because their performances are closely related to network latency and throughput. Multistage interconnection networks are well suitable for communication among tightly coupled system components and can offer a good balance between cost and performance. For complex systems, assuring high reliability is a challenging task. Thus fault tolerance is crucial for MINs to serve the communication needs to large-scale multiprocessor systems (Adams et al., 1987).

To enhance the fault-tolerance capability, 3×3 SEs have been used as basic building blocks. For example, gamma interconnection network (GIN) (Parker and Raghavendra, 1984), augmented data manipulator (ADM), inverse augmented data manipulator (IADM) (McMillen and Siegel, 1984) are included in that category. (Gamma and IADM are topology equivalent but different in control algorithm.) In Gamma network, there are multiple paths between any source and destination pair except on the condition that the source and destination are the same. To improve the fault-tolerant capability of GIN, several schemes have been introduced such as Extra Stage Gamma Network (Yoon and Hegazy, 1988), PM22I interconnection network (Lee and Yoon, 1989), CGIN (Chuang, 1996), composite banyan (Seo and Feng, 1995), B-network (Lee and Yoon, 1990), PCGIN, and FCGIN (Chen et al., 2000). Extra Stage Gamma Network (Yoon and Hegazy, 1988) and PM22I (Lee and Yoon, 1989) provide multiple paths between any source and destination pairs by adding one more stage to GIN, but the delay time and routing conflicts can be increased. CGIN (Chuang, 1996) is a revised GIN by providing at least two disjoint paths, however, its routing algorithm is complicated. When a fault occurs, the system must use the backtracking scheme to remedy the fault. Composite banyan (Seo and Feng, 1995) is composed of two banyan networks (Goke and Lipovski, 1973; Wu and Feng, 1980), one of them is a normal banyan network and the other is an inverse banyan network. Because of their symmetric topology, composite banyan network shows two disjoint paths. B-network (Lee and Yoon, 1990) uses a redundant link at each stage in GIN as a backward link. When a fault exists in the routing path, the packet will go backward to the previous stage to find the alternate path. But if a link fault occurs

* Corresponding author. Tel.: +86-35712121 ext. 54734; fax: +86-35724176.
E-mail addresses: cwchen@csie.nctu.edu.tw (C.-W. Chen), cpchung@csie.nctu.edu.tw, cpchung@ccl.itri.org.tw (C.-P. Chung), .

0164-1212/01/$ - see front matter
PII: S0164-1212(01)00025-5

between stage 0 and stage 1, no alternate path can be found. PCGIN (Chen et al., 2000) also provides two disjoint paths between any source and destination pair by using destination tag routing functions to generate two disjoint paths rather than distance tag routing. FCGIN (Chen et al., 2000) uses chaining links (Tzeng et al., 1988) between two switches at the same stage to provide not only disjoint paths but also strong reroutability (Chen et al., 2000; Tzeng et al., 1988) to tolerate faults, but the fault penalty still exists. Although these networks mentioned above can provide the capacity to tolerate a fault, the penalties of tolerating a fault are still high.

The rerouting schemes of the networks discussed so far need rerouting packets via backtracking (Rau et al., 1992), with the exception of B-network (Lee and Yoon, 1990) and FCGIN (Chen et al., 2000). With backtracking methodology, packets backtrack along the path traversing to find an alternate path. Such routing schemes will degrade system performance and increase hardware cost. Instead of backtracking, B-network uses backward links to tolerate faults. But B-network cannot guarantee one-fault tolerant; moreover, fault penalties still exist. The situations inspire us to design a network such that packets are capable of tolerating a fault without penalty.

In this paper, we propose a new multipath network called No-backtracking Gamma interconnection network (NBGIN). The NBGIN can tolerate one link or switch fault without backtracking. And then, we demonstrate a destination tag routing algorithm, which can route a packet to destination without backtracking under a fault. The remainder of this paper is organized as follows. In Section 2, we introduce the Gamma interconnection network, destination tag routing functions in GIN, and the properties of GIN. In Section 3, we present the NBGIN, a destination routing function, and the routing algorithm. Section 4 describes the detailed comparisons of some fault-tolerant GINs. Finally, Section 5 concludes this paper.

2. Gamma interconnection networks

2.1. Topology

A GIN of size $N = 2^n$ consisting of $n+1$ stages is labeled from 0 to n and each stage involves N switches (Parker and Raghavendra, 1984). Basically, switches of sizes 1×3 and 3×1 are coupled with the first and last stages, respectively. Moreover, each switch located at intermediate stages is a 3×3 crossbar. And each 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 jth switch at stage i has three output links to switches $[(j - 2^i) \bmod N], j,$ and $[(j + 2^i) \bmod N]$ at the consecutive stage. Fig. 1 illustrates a GIN network with size 8.

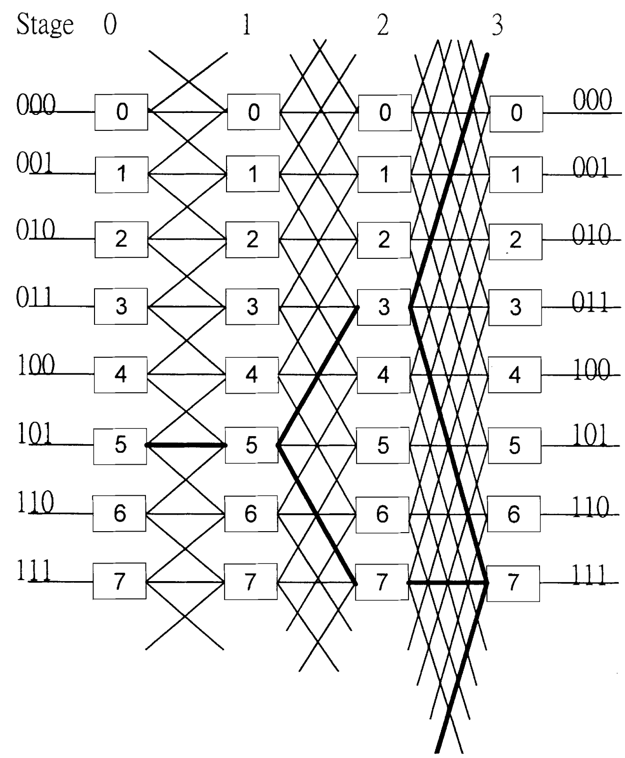

Fig. 1. Gamma interconnection network with $N = 8$ and its three paths between nodes 5 and 7.

In GIN, an n-digit tag determines the path connecting the source to its destination. Each tag digit can be 1, 0, or $\overline{1}$. An n-digit tag T represents the difference between D and S, i.e., $T = D - S(\bmod N)$. Digit t_i is used at stage i in such a way that the lower connection is taken when t_i is equal to 1, and the straight connection is taken when t_i is 0 where the distance $T = t_0, t_1 \cdots t_{n-1}$. The GIN makes use of the binary fully redundant number system to represent each tag. A non-zero tag T has multiple representations, that is, there are multiple paths between source S and destination D, when $S \neq D$. For example, when $N = 8$, source node S is 5, and destination node D is 7, the tag T can be 010 or $0\overline{1}\,1$, or $0\overline{1}\overline{1}$ as shown in Fig. 1.

2.2. Destination tag routing in GIN

The routing algorithm discussed above is the distance tag routing: the sender must compute the tag before the packet is sent. To avoid preprocessing control overhead, we propose two destination tag routing functions $carry(j, d_i)$ and $borrow(j, d_i)$ where j means the switch number at stage i, and $d = d_0 d_1 d_2 \cdots d_{n-1}$ means the destination. Initially, we set the switches to be either even or odd, that is, a switch $j = j_0 j_1 j_2 \cdots j_{n-1}$ is even (odd) at stage i if j_i is 0(1).

Definition 1.

$$carry(j, d_i) = \begin{cases} j + 2^i & \text{if } j \text{ is an even(odd) switch and } d_i = 1(0), \\ j & \text{otherwise,} \end{cases}$$

$$borrow(j, d_i) = \begin{cases} j - 2^i & \text{if } j \text{ is an even(odd) and } d_i = 1(0), \\ j & \text{otherwise.} \end{cases}$$

By Definition 1, a packet is routed to the same switch number to the consecutive stage, when j_i is equal to d_i regardless of which function is applied. *Carry* function has behavior of moving toward either straight or down direction, but *borrow* function goes up or straight. By Definition 1, when the two functions proceed non-straight link, it shows that the switch j at stage i is even (odd), and $d_i = 1$ (0). So the choice of progression to next link can be done by destination tag only instead of precomputed tag. The behavior of switching element of *carry* and *borrow* functions is depicted in Fig. 2.

Lemma 1. *The functions $carry(j, d_i)$ and $borrow(j, d_i)$ are destination tag routing functions.*

Proof. We only prove the *carry* part. The borrow part can be proven similarly. By Definition 1, as *carry* routing function is applied at stage $i, 0 \leqq i \leqq n-1$, the packet in the switch $j = j_0 j_1 j_2 \cdots j_{n-1}$ at stage i will reach to switch $j_0 j_1 \cdots j_{i-1} d_i p_{i+1} \cdots p_{n-1}$ at stage $i+1$, where the destination is $d_0 d_1 d_2 \cdots d_{n-1}$ and p_i is 0 or $\overline{1}$. Given $S = s_0 s_1 \cdots s_{n-1}$ and $D = d_0 d_1 d_2 \cdots d_{n-1}$, the routing path will be $s_0 s_1 \cdots s_{n-1} \rightarrow d_0 p_1 \cdots p_{n-1} \rightarrow d_0 d_1 q_2 \cdots q_{n-1} \rightarrow \cdots \rightarrow d_0 d_1 d_2 \cdots d_{n-1}$ regardless of either carry or borrow function is applied. □

From Lemma 1, the switches traversed by carry or borrow at stage i have the same first i bits as those of the destination's. Once the two paths generated by carry and borrow go through the first non-straight links, they will not meet until the destination is arrived. That is, we want to demonstrate that there are at least two disjoint paths in GIN when $(S - D) \bmod 2 = 1$.

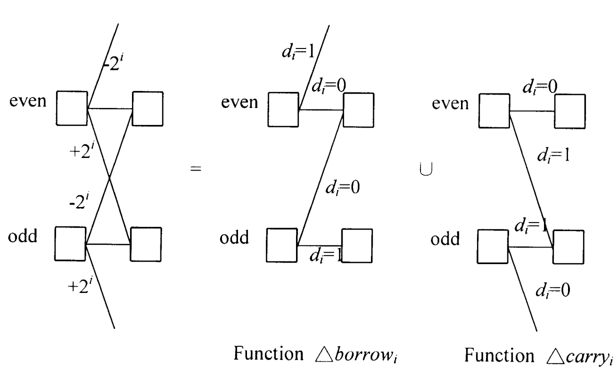

Fig. 2. Switching by $\triangle borrow_i$ and $\triangle carry_i$ functions at stage i.

Theorem 1. *When the difference between any source S and destination D is odd, there are at least two disjoint paths in GIN.*

Proof. When the difference of source S and destination D is odd, two paths generated by *carry* and *borrow* functions will take non-straight links from stage 0 to 1. Let *v_sum* be the vertical distance traversed from source node to current node and be positive. For example, if a path is $1 \rightarrow 0 \rightarrow 6 \rightarrow 2$, then *v_sum* will be $1 + 2 + 4 = 7$. However at stage $n-1$, these two paths still cannot meet because the *v_sums*' sum of the two paths does not exceed 2^n. The sum of these two paths' *v_sum* at stage $n-1$ in the worst case scenario is shown as below:

$$2(2^0 + 2^1 + 2^2 + \cdots 2^{n-2}) = 2^n - 2 < 2^n.$$

These two paths will reach to destination based on Lemma 1, therefore any two paths generated by *carry* and *borrow* functions are disjoint. □

To perform carry or borrow routing functions, we set the switches to be either even or odd depending on their initial location. When a fault occurs, the backtracking method will be started. If the straight link to next stage is the route to be taken and there is a fault on it (or the switch of next stage is faulty), the packet will backtrack along the path traversed to find a non-straight link. When non-straight connecting stages i and $i+1$ are found, the packet goes back to stage i and changes the state of the switch at stage i to alternate state. So the other non-straight link is taken to stage $i+1$ to be the alternate link. For example, if the source node is 0 and the destination node is 1 with size = 8 in GIN, the paths are $0 \rightarrow 1 \rightarrow 1 \rightarrow 1$ and $0 \rightarrow 7 \rightarrow 5 \rightarrow 1$ by applying carry and borrow functions, respectively. Assuming that the link to switch 1 at stage 2 is faulty and the carry function is used. The packet will backtrack to stage 0 to find a non-straight link and the state of the switch will change to borrow. As a result of changing its state, the other non-straight link can be taken as an alternate link. In other words, the path will be $0 \rightarrow 1 \rightarrow (x)1, 0 \leftarrow 1$ (means backtracking penalty), and $0 \rightarrow 7 \rightarrow 1 \rightarrow 1$ as shown in Fig. 3.

2.3. A new routing function in GIN

From Theorem 1, it can be seen that when the difference between source and destination is odd, GIN can tolerate one fault. But the proposed routing schemes need to use backtracking or rerouting packets from the source node. However, such routing schemes will degrade system performance. This situation motivates us to determine whether a packet can be routed without backtracking.

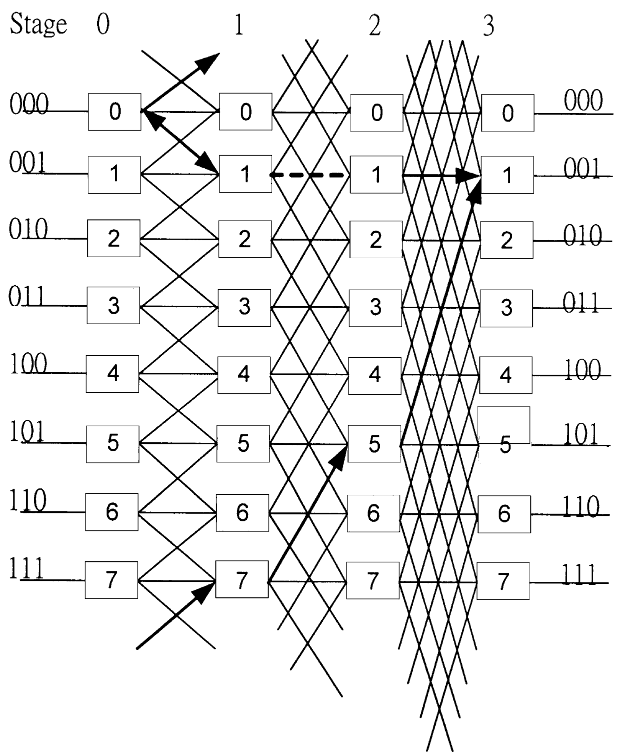

Fig. 3. Backtracking routing under a fault in GIN with $N = 8$ between nodes 0 and 1. The dash line means a fault link.

Before introducing a new routing function, let us review distance tag routing in GIN. At first, we calculate the routing tag $t = t_0t_1t_2\cdots t_{n-1}$ obtained by $D - S$, where t_i is 0, 1 or $\bar{1}$. At stage i, the down (up) link is taken if t_i is 1 ($\bar{1}$), and the straight link is applied if t_i is 0. In distance tag routing, a non-straight link is taken when the tag bit is 1 or $\bar{1}$. If a number being represented can consist of only 1 or $\bar{1}$, the routing path will be all non-straight links. In Theorem 2, we want to show that the representation of $t = (D - S)$ having mode N can consist of only 1 or $\bar{1}$, where $t \bmod 2 = 1$.

Theorem 2. *An odd distance tag t can be represented to $t_0t_1\cdots t_{n-1}$, where t_i is 1 or $\bar{1}$ and $0 \leqq t \leqq N - 1$.*

Proof. When $t = 1$, $t\bar{1}\bar{1}\cdots\bar{1}1$, or $t\bar{1}\bar{1}\cdots\bar{1}$.

Suppose $t = 2j - 1$ is valid. That is, $t = t_0t_1\cdots t_{n-1}$, where t_i is 1 or $\bar{1}$.

Given $t \leqq N - 3$, because when t is equal to $N - 1, t = 11\cdots 1$.

Consider $(2j - 1) + 2$.

Because $(2j - 1) \leqq N - 3$, there is at least one $\bar{1}$ in the representation $t_0t_1\cdots t_{n-1}$ of $2j - 1$. Consider the following situations:

(a) If t_0 is $\bar{1}, 2j + 1 = \bar{t_0}t_1 \ldots t_{n-1}$.

(b) If t_0 is 1, let t_0 change to $\bar{1}$, and consider t_1.

The consideration of t_1 is similar to (a) and (b), and then $t_2, t_3\cdots t_i$ may be traced till $\bar{1}$ is found. However the tracing will be terminated because there exists at least one $\bar{1}$ in $2j - 1$.

Any odd number t can be represented by $t_0t_1\ldots t_{n-1}$ where t_i is 1 or $\bar{1}$ and $0 \leqq t \leqq N - 1$. □

From Theorem 2, we can realize that when the difference between any source and destination pair is odd, the routing path can always go through non-straight links. And in the backtracking scheme, if a packet takes a non-straight link to next stage and a fault occurs (may be on the link or the switch of next stage), the packet can take the other non-straight link to next stage as an alternate link without backtracking. In other words, an alternate path can be found out immediately without backtracking if the routing path is composed of non-straight links only. In Algorithm 1, we show how a binary number is transferred to represent to only 1 and $\bar{1}$.

Algorithm 1.

Input: Source node S and Destination node D
Output: Distance tag $T : t_0t_1\cdots t_{n-1}$ (t_i is 1 or $\bar{1}$)
Begin
  $b = (D - S) \bmod N$;
  for $i = 0$ to $n - 1$
    if $b_i \neq 0$
      $t_i = b_i$;
    else
      $t_{i-1}t_i = \bar{1}\,1$;
  end for End

So far, the no-backtracking routing scheme discussed is based on distance tag routing method. That is, the routing tag must be computed first before a packet is sent. Let us consider a destination tag routing function without the need of backtracking in GIN.

Definition 2. Let $d = d_0d_1\cdots d_{n-1}$ be the destination and $j = j_0j_1\cdots j_{n-1}$ be the switch at stage i.

When $0 \leqq i \leqq n - 2$

$$NB(j, d_id_{i+1}) = \begin{cases} j - 2^i & \text{if } j_ij_{i+1} \text{ is } 01(10) \text{ and } d_id_{i+1} = 11(01), \\ & \text{or if } j_ij_{i+1} \text{ is } 00(11) \text{ and } d_id_{i+1} = 10(00), \\ j & \text{if } j_i \text{ is an } 0(1) \text{ and } d_i = 0(1), \\ j + 2^i & \text{if } j_ij_{i+1} \text{ is } 00(11) \text{ and } d_id_{i+1} = 11(01), \\ & \text{or if } j_ij_{i+1} \text{ is } 01(10) \text{ and } d_id_{i+1} = 10(00). \end{cases}$$

And when $i = n - 1$

$$NB(j, d_{n-1}) = \begin{cases} j - 2^{i-1} \text{ or } j + 2^i \\ \quad \text{if } j_i \text{ is } 0(1) \text{ and } d_{n-1} = 1(0), \\ j \quad \text{if } j_i \text{ is } 0(1) \text{ and } d_n = 0(1). \end{cases}$$

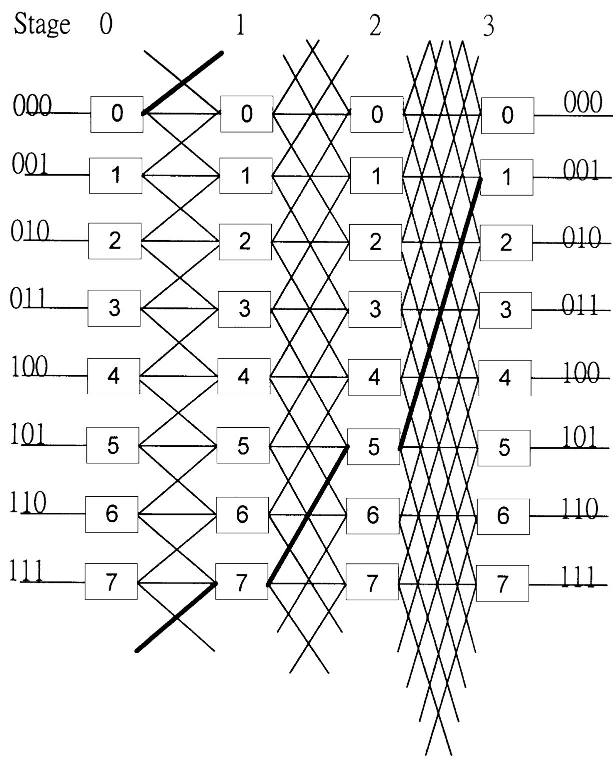

Fig. 4. The path moved by applying NB routing function in GIN with $N = 8, S = 0$ and $D = 1$.

In Definition 2, when a packet is routed from stage $n-1$ to n by a non-straight link, any of these two non-straight links can be chosen because they are redundant. The purpose of Definition 2 is to make the $(i+1)$th bit of the switch at stage $i+1$ to be different from d_{i+1} as a packet is routed from stage i to $i+1$. That is, *NB* function will always take non-straight links from source to destination when the difference between source and destination is odd. For example, as illustrated in Fig. 4, when $S = 0$ and $D = 1$ with $N = 8$ in GIN, the path will be $0 \rightarrow 7 \rightarrow 5 \rightarrow 1$. The *NB* function is implemented as follows: At first, the link to switch 7 is taken because $j_0 j_1 = 00$ and $d_0 d_1 = 10$ are at stage 0. At stage 1, the up link to switch 5 is taken as a result of $j_1 j_2 = 11$ and $d_1 d_2 = 00$. Finally, one of the non-straight links to switch 1 is taken because d_{n-1} is different from j_{n-1} at stage 2.

Lemma 2. *NB function is a destination tag routing function and it routes a packet to its destination only by non-straight links as the difference between the source S and the destination D is odd.*

Proof. Because $(S-D) \bmod 2 = 1$, the proper non-straight link is taken by *NB* function from stage 0 to 1. And then, the bit j_1 of the switch number j at stage 1 will be different from d_1 according to Definition 2. In addition, the non-straight link is still taken to stage 2, and *NB* function will keep the current bit different from d_2. Generally speaking, *NB* function will always reach the odd (even) switch at stage i and the d_i is 0 (1). Eventually, the packet can be routed to destination by destination tag and non-straight links when $(S-D) \bmod 2 = 1$. □

3. No-backtracking gamma interconnection network

There are two parts in GIN routing, one is $(S-D) \bmod 2 = 1$, and the other is $(S-D) \bmod 2 = 0$, where S and D designate source and destination, respectively. So far, we have only discussed the first part of GIN routing. In this section, no-backtracking gamma interconnection network (NBGIN) obtained by modifying GIN is presented. NBGIN can tolerate one fault without paying backtracking penalty for any source and destination pair.

3.1. Topology

An NBGIN of size $N = 2^n$ is similar to GIN, but at stage 0 switch $2i$ and switch $2i+1$ are coupled into one 2×4 switch, and other stages are kept the same as GIN. Fig. 5 shows the NBGIN with size = 8. The naming scheme at stage 0 is as follows. A switch of j at stage 0 is an even switch if $j \bmod 2 = 0$, otherwise it is an odd switch. And the four associated links are named as 10, 01, 11, and 00 if the switch is even. Whereas, the links

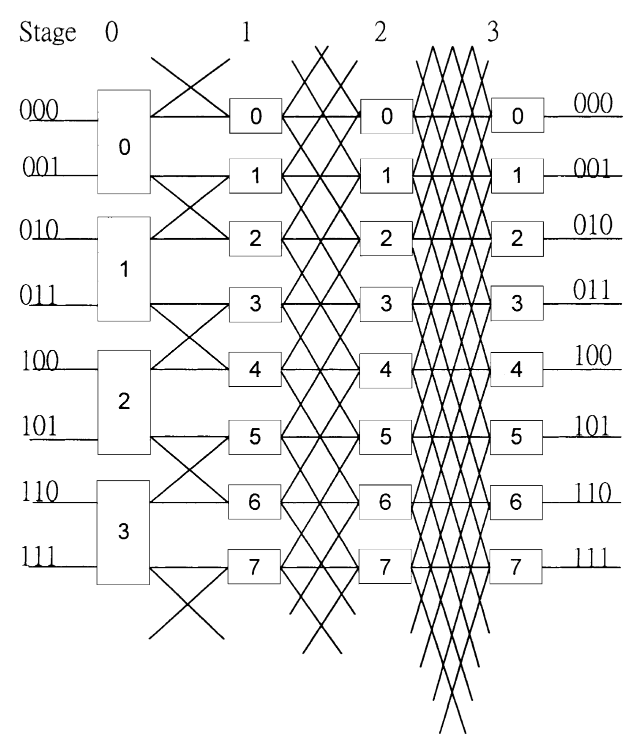

Fig. 5. No-backtracking gamma interconnection network with $N = 8$.

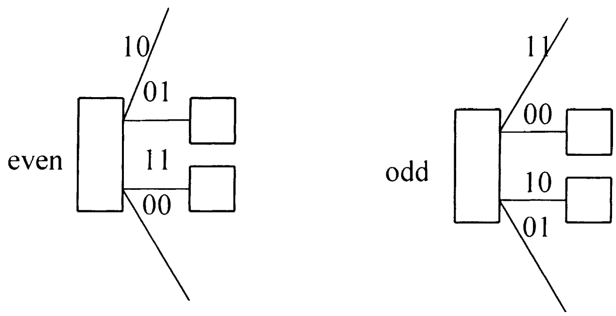

Fig. 6. The behavior of *NB* routing function at stage 0 in NBGIN.

are named as 11, 00, 10, and 01 if the switch is odd as shown in Fig. 6.

3.2. A new destination tag routing function in NBGIN

Here we present a destination tag routing function *NB* that can route a packet by non-straight links only from source to destination in NBGIN. Let the destination $D = d_0d_1 \cdots d_{n-1}$, and the modified *NB* destination tag routing function is shown in Definition 3. Fig. 6 exhibits the behavior of *NB* routing function at stage 0 in NBGIN.

Definition 3. At stage 0

$$NB(j, d_0d_1) = \begin{cases} 2j-1 & \text{if } j_0 = 0(1) \text{ and } d_0d_1 = 10(11), \\ 2j & \text{if } j_0 = 0(1) \text{ and } d_0d_1 = 01(00), \\ 2j+1 & \text{if } j_0 = 0(1) \text{ and } d_0d_1 = 11(10), \\ 2j+2 & \text{if } j_0 = 0(1) \text{ and } d_0d_1 = 00(01). \end{cases}$$

Additional stages are the same as Definition 2.

According to Definition 3, a packet can be routed to a switch j at stage 1 where the bit j_1 is different from d_1. There remain two more choices to reach stage 2 by non-straight links. By choosing the proper non-straight link from Definition 2, the packet will have two non-straight links going to stage 3 and so on. The behavior from stage 1 to the destination is similar to that described in Lemma 2 by *NB* destination tag routing function.

Lemma 3. *NB routing function can route a packet only through non-straight links from stage* 1 *to stage n in NBGIN between any source-destination pair.*

Proof. From Definition 3, we can conclude that when a packet is routed to switch $j = j_0j_1 \cdots j_{n-1}$ at stage 1 by *NB* routing function in NBGIN, the current bit j_1 is different from the bit d_1, where $d = d_0d_1 \cdots d_{n-1}$ is the destination. And by Lemma 2, the packet will be routed to switch $q = q_0q_1q_2 \cdots q_{n-1}$ of stage 2 by a non-straight link such that q_2 is different from d_2. Generally speaking, from stage $i-1$ to stage i, *NB* function will take the switch $j = j_0j_1 \cdots j_{n-1}$ at stage i , where j_i is different from the bit d_i. So the routing path from stage 1 to stage n always takes non-straight links from any source to any destination. □

From Lemma 3, an alternate link can be found immediately from stage 1 to stage n if there is a fault. Let us consider the situation from stage 0 to 1. Because four links are connected to stage 1 from one switch, there are two links to be chosen from when d_0 is 0 or 1. If there is a link fault between stage 0 and 1 or a switching fault at stage 1, the alternate link is the choice.

Theorem 3. *NBGIN can tolerate one fault except stage* 0 *and n without backtracking.*

Proof. It can be proved with two parts: the first part is that a switch fault occurs at stage 1 or a link fault between stages 0 and 1, and the second part is that a fault occurs between stages 1 to n.

When a switch fault at stage 1 or a link fault between stage 0 and 1 occurs, there is an alternate link to be used to connect another switch at stage 1 because there are two links for $d_0 = 0$ or 1. In addition, if there is no fault from the first part, there are two choices going to stage 2. By Lemma 3, the packet can be routed to destination only by non-straight links. That is, if there is a fault, the other non-straight link is the alternate link to tolerate the fault.

By *NB* routing function, we can route a packet to destination without backtracking in NBGIN. □

From Theorem 3, we can see that a packet can be routed to destination without backtracking. Moreover, when a fault occurs, an alternate link (the other non-straight link) can be found right away without paying any penalties due to links fault.

3.3. Routing and rerouting schemes

Algorithm 2 shows the routing algorithm of NBGIN. If there is no fault in NBGIN, by Lemma 3, the packet can be routed to destination by applying *NB* routing function. When a fault occurs, an alternate path (the other non-straight link at the same stage) can be found immediately. That is, when the non-straight link or the switch at stage i is faulty between stage $i-1$ and i , the other non-straight link can be taken to stage i . Then, the straight link will be taken from stage i to $i+1$ after the alternate link from stage $i-1$ to i had been progressed. If a non-straight link is taken between stage $i+1$ and destination, *NB* function will choose the proper non-straight link in order to tolerate one more fault. But if the path from stage $i+1$ to destination consists of straight links only, no more faults can be tolerated. Therefore, it is concluded that NBGIN is one-fault tolerant and multiple faults robust.

Algorithm 2.

Input: destination tag $d_0d_1d_2\cdots d_{n-1}$, the source node $j = j_0j_1j_2\cdots j_{n-1}$

begin

$j = \mathrm{int}(j/2)$;

for $(i = 0; i < n; i++)\{$

if $(i = 0)\{$

if(the switch is even){

switch(d_0d_1)

case:10

the first link to switch $2j - 1$ is taken;

case:01

the second link to switch $2j$ is taken;

case:11

the third link to switch $2j + 1$ is taken;

case:00

the fourth link to switch $2j + 2$ is taken;

}

if(the switch is odd){

switch(d_0d_1)

case:11

the first link to switch $2j - 1$ is taken;

case:00

the second link to switch $2j$ is taken;

case:10

the third link to switch $2j + 1$ is taken;

case:01

the fourth link to switch $2j + 2$ is taken;

}

/* for example, the alternate link of first link is third link */

if the chosen link is faulty then

the alternate link is (the chosen link +2) mod 4

}

else{

if ($j_ij_{i+1} = 00$ and $d_id_{i+1} = 10$) or ($j_ij_{i+1} = 01$ and $d_id_{i+1} = 11$) or ($j_ij_{i+1} = 10$ and $d_id_{i+1} = 01$) or ($j_ij_{i+1} = 11$ and $d_id_{i+1} = 00$)

the link to switch $j - 2^i$ is taken to stage $i + 1$

else if ($j_i = 0$ and $d_i = 0$) or ($j_i = 1$ and $d_i = 1$)

the straight link to switch j is taken to stage $i + 1$

else

the other non-straight link to switch $j - 2^i$ is taken to stage $i + 1$

}

if the non-straight link is faulty

the other non-straight link is alternate to next stage

else

print the network have more than two faults, and then *exit*

j is set to a new switch number at stage $i + 1$

end

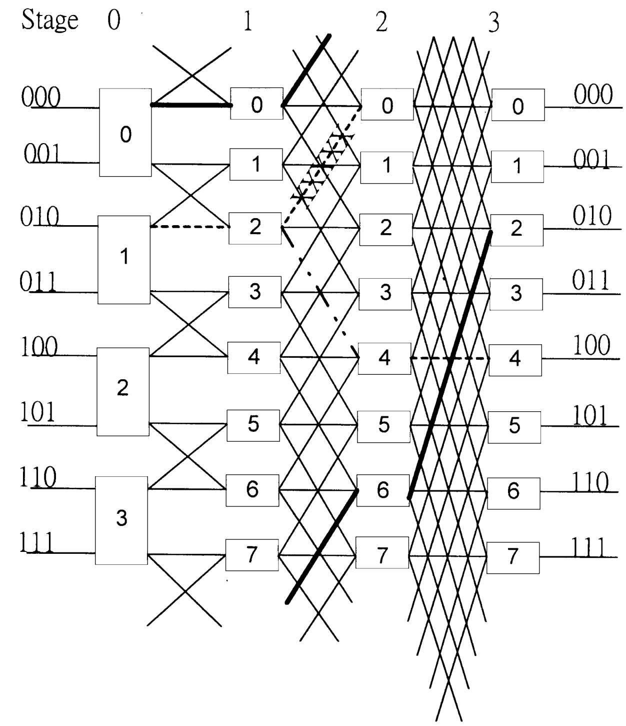

Fig. 7. The routing path with $S = 0$ and $D = 2$ without faults shown by bold lines, and the fault-tolerant path with $S = 2$ and $D = 4$ shown by dash lines in NBGIN with $N = 8$.

Fig. 7 illustrates a routing example of NBGIN with $N = 8$. The bold line shows a fault-free path from $S = 0$ to $D = 2$. The dotted lines move the routing path from $S = 2$ to $D = 4$, where the link to switch 0 of stage 2 is faulty. At first, these two separate links to switch 2 and switch 4 can be taken randomly, and we take the link to switch 2 by *NB* function. Thus, the link to switch 0 of stage 2 should be used. However, the link between switch 2 of stage 1 and switch 0 of stage 2 is faulty. To bypass the fault, an alternate link from switch 2 of stage 1 to switch 4 of stage 2 is taken, and finally, the straight link is taken to reach the destination.

4. Comparison and analysis

To demonstrate the supremacy of NBGIN, we have compared NBGIN with GIN and the other four gamma-induced interconnection networks, CGIN, B-network, PCGIN, and FCGIN. Table 1 shows the characteristics of these networks.

The hardware costs for these networks are almost equal except at the stage 0 or n. If a link fault occurs between stages 0 and 1, the B-network has no alternate path. In CGIN (Chuang, 1996) and PCGIN (Chen et al., 2000), there are at least two disjoint paths between any source and destination pair, and the routing schemes are distance tag and destination tag routing, respectively.

Table 1
The comparison of GIN, CGIN, B-network, PCGIN, FCGIN, and NBGIN

Network	Fault-tolerance method	Single-fault tolerant	Routing method	Rerouting link penalty
GIN	Multiple paths	No	Distance tag routing	–
Cyclic GIN	Backtracking	Yes	Distance tag routing	$\frac{n-1}{3}$
B-network	Dynamic alternating path	No	Destination tag routing	2 links
PCGIN	Backtracking	Yes	Destination tag routing	$\frac{1}{n}\left(\frac{13}{42}n+\frac{79}{147}-\frac{17}{49}\left(\frac{-3}{4}\right)^{n-1}\right)$
FCGIN	Dynamic alternating path	Yes	Destination tag routing	1
NBGIN	Dynamic alternating path	Yes	Destination tag routing	0

And FCGIN also uses destination tag scheme. As for the capability of fault tolerance, CGIN, PCGIN, FCGIN, and NBGIN are single-fault tolerant from stage 0 to n, whereas B-network is one-fault tolerant between stage 1 and n. Once a fault occurs, CGIN and PCGIN use the backtracking mechanism to find alternate paths; but B-network, FCGIN and NBGIN can find an alternate path without backtracking. In B-network, if there is a fault between stages i and $i+1$, the backward link to stage $i-1$ is taken first and then followed by a non-straight link to stage i (Lau and Poon, 1998). If the link to stage i is faulty, the backward link to stage $i-2$ will be taken to find the alternate path. Rather than using backward links, FCGIN uses chaining link to tolerate faults. In NBGIN, when a fault occurs, the other non-straight link will be taken to reach the next stage, with the penalty being 0. However, the penalty is 2 under a fault in B-network and it is 1 in FCGIN.

In CGIN, when a fault occurs, the packet is backtracking to stage 0 and the rerouting tag is used as a routing tag. As a result, the average number of backtracked links is

$$\frac{1}{3n} * \sum_{i=0}^{n-1} 2i = \frac{n-1}{3}.$$

In PCGIN (Chen et al., 2000), the probability is 0.5 for either a fault to occur at a straight link or a switch to occur at a straight link or a switch to be reached via a straight link; and the probability is 1/3 for straight link fault to occur. Hence, there is a 16.67% probability to backtrack, but at stage $n-1$ the probability is 50% because no redundant link is available. Let the backtracking penalty at stage i be $T(i)$:

$$T(i) = \frac{1}{2} * \left(1 + \frac{1}{2}(T(i-1)+1)+1\right)$$
$$= \frac{3}{4} + \frac{1}{4}T(i-1), \quad T(0)=1, \quad T(1)=5/2$$
$$T(n) = \frac{13}{7} - \frac{6}{7}\left(\frac{-3}{4}\right)^{n}.$$

Average penalty is

$$\frac{1}{n}\left(\frac{1}{6}\sum_{i=0}^{n-2} T(i) + \frac{1}{2}T(n-1)\right)$$
$$= \frac{1}{n}\left(\frac{13}{42}n. + \frac{79}{147} - \frac{17}{49}\left(\frac{-3}{4}\right)^{n-1}\right) \quad \text{for } n \geqq 2.$$

From Table 1, it can be seen that the penalty in NBGIN is zero under a fault; in contrast, others need more than 2 links to accomplish the same result. In addition, with respect to routing method, NBGIN and B-network use destination tag routing; however, others need distance tag routing. Regarding the capability of fault tolerance, CGIN and NBGIN permit one-fault tolerant and are robust with multiple faults, but GIN and B-network are robust with multiple faults only. In terms of hardware cost, NBGIN needs very little at stage 0 which composes of $N/2$ 2×4 switches instead of N 1×3 switches. However, backtracking needs more hardware expenses, because it must provide a bi-direction network as the worst case happens.

5. Conclusion remarks

A new multipath multistage interconnection network called NBGIN is proposed in this paper. NBGIN allows one-fault tolerant; and its advantages over existing fault-tolerant GINs are dynamic rerouting plus low rerouting penalties under a fault. In addition, we also present a destination tag routing function called *NB* routing function. By applying *NB* function, the alternate path can be found immediately when a fault occurs in NBGIN. Finally, we compare and summarize the features of six networks in terms of cost, fault tolerability and average penalties. To sum up, NBGIN represents an interconnection network of which it has capabilities of performing one-fault tolerant, low fault penalty, dynamic rerouting and destination tag routing.

References

Adams III, G.B., Agrawal, D.P., Siegel, H.J., 1987. A survey and comparison of fault-tolerant multistage interconnection networks. IEEE Transactions on Computer 20 (6), 14–27.

Chen, C.W., Lu, N.P., Chen, T.F., Chung, C.P., 2000. Fault-tolerant gamma interconnection networks by chaining. IEE Proceedings of Computers and Digital Techniques 147 (2), 75–80.

Chuang, P.J., 1996. CGIN: a fault tolerant modified gamma interconnection network. IEEE Transactions on Parallel and Distributed Systems 7 (12), 1301–1306.

Goke, L.R., Lipovski, G.J., 1973. Banyan networks for partitioning multiprocessing systems. In: Proceedings of the First Annual Computer Architecture Conference. pp. 21–28.

Lee, K.Y., Yoon, H., 1989. The PM22I interconnection network. IEEE Transaction on Computers 38 (2), 302–307.

Lee, K.Y., Yoon, H., 1990. The B-network: a multistage interconnection network with backward links. IEEE Transactions on Computers 39 (7), 966–969.

Lau, F.C.M., Poon, W.C., 1998. Throughput analysis of B-networks. IEEE Transaction on Computers 47 (47), 482–485.

McMillen, R.J., Siegel, H.J., 1984. Routing schemes for augmented data manipulator network in an MIMD system. IEEE Transactions on Computers 33, 367–373.

Parker, D.S., Raghavendra, C.S., 1984. The gamma network. IEEE Transactions on Computers C-33, 367–373.

Rau, D., Fortes, J.A.B., Siegel, H.J., 1992. Destination tag routing techniques based on a state model for the IADM network. IEEE Transactions on Computers 41 (3), 274–285.

Seo, S.W., Feng, T.Y., 1995. The composite banyan network. IEEE Transactions on Parallel and Distributed Systems 6 (10), 1043–1054.

Tzeng, N.F., Yew, P.C., Zhu, C.Q., 1988. Realizing fault-tolerant interconnection networks via chaining. IEEE Transaction on Computers C-37(4), 458–462.

Wu, C.L., Feng, T., 1980. On a class of multistage interconnection networks. IEEE Transactions on Computers 29, 694–702.

Yoon, K., Hegazy, W., 1988. The extra stage gamma network. IEEE Transactions on Computers 37 (11), 1445–1450.

Ching-Wen received the B.E. degree in information engineering and computer science from Feng Chia University, Taiwan, Republic of China in 1993, and the M.E. degree in Department of Computer Science from National Tsing Hwa University, Taiwan, Republic of China in 1995. Currently he is pursuing the Ph.D. degree in computer science and information engineering at the National Chiao Tung University. His research interests include computer architecture, interconnection network and parallel processing.

Chung-Ping received the B.E. degree from the National Cheng-Kung University, Taiwan, Republic of China in 1976, and the M.E. and Ph.D. degrees from the Texas A&M University in 1981 and 1986, respectively, all in electrical engineering. He was a lecturer in electrical engineering at the Texas A&M University while working towards the Ph.D. degree. Since 1986 he was been with computer science and information engineering at the National Chiao Tung University, Taiwan, Republic of China, where he is a professor. From 1991 to 1992, he was a visiting associate professor of computer science at Michigan State University. Currently, he is on leave and served as the director of Advanced Technology Center, Computer and Communications Research Laboratories, Industrial Technology Research Institute (CCL, ITRI), ROC, and then the consultant of CCL, ITRI. His research interests include computer architecture, parallel processing, and parallel compiler design.