TNC Transactions on Networks and Communications
Society for Science and Education
United Kingdom
Volume 7, No. 1
ISSN: 2054 -7420

Fault-Tolerant Adaptive Routing Algorithm for 2D Torus Network

Tsukasa-Pierre Nakao, Yasuyuki Miura, Naohisa Fukase
Shonan Institute of Technology, Kanagawa, JP;
nakaot@center.shonan-it.ac.jp; miu@info.shonan-it.ac.jp; fukasen@center.shonan-it.ac.jp
+81-466-30-0207, 1-1-25, Tsujido-Nishikaigan, Fujisawa, Kanagawa, Japan

ABSTRACT

A 2D torus network is one of the most popular networks for parallel processing. We have researched the North-South First (NSF) routing which is applicable to a 2D torus and combines the north-first (NF) and south-first (SF) methods. We focused on the proposal of a routing algorithm aimed at avoiding congestion of the crowded network. It was superior in congestion tolerance but not in fault tolerance. We have therefore been researching algorithms considering fault tolerance of the NSF method. In this paper we propose an NSF-FT method which is a new routing algorithm with improved fault tolerance. We evaluated the congestion resistance and fault tolerance of the proposed method by dynamic communication performance evaluation by simulation. The software simulation showed that the proposed algorithm has higher performance.

Keywords: Network on Chip; Interconnection Network; Adaptive Routing; Turn Model; Fault Tolerance.

1 Introduction

The interconnection network is an important topic in the field of parallel processing. Parallel computers have processing elements (PEs) that are directly connected to a network such as a *k*-ary *n*-cube. Parallel processing is also performed in a *Network on Chip* (NoC) between PEs located on one chip. Many different interconnection networks for parallel processing have been proposed, and the 2D torus network is one of the most popular networks for parallel processing.

The routing algorithms of interconnection networks are classified into deterministic routing, in which paths are fixed, and adaptive routing [1-11], in which paths are changed to avoid failures or congestion. Because of its tolerance to failures and congestion, adaptive routing has been the topic of a lot of research. Various adaptive routing algorithms have been proposed for *k*-ary *n*-cubes [6]-[10, 12, 13].

[However, these methods require additional hardware for virtual channels comparison with deterministic routing (its name is *Dimension Order Routing*, DOR). For example, Dally et al. proposed two types (dynamic and static) of *Dimension Reversal routing* [7]. Those methods require at least one (dynamic) or two (static) additional virtual channels for achieving adaptive routing. The fully adaptive routing for *k*-ary *n*-cubes can also be based on Duato's method [11], but this method requires one additional virtual channel for the bypass path needed for achieving adaptive routing.

DOI: 10.14738/tnc.71.6032
Publication Date: 09th February 2019
URL: http://dx.doi.org/10.14738/tnc.71.6032

A number of adaptive routing algorithms based on the turn model [14-17] do not need additional virtual channels. However, most of these algorithms cannot be applied to torus networks without change. If an adaptive routing algorithm for a torus network could be realized by modifying the turn model, it would be possible to realize adaptive routing without having to install additional virtual channels [18].

We have previously proposed the North-South First (NSF) algorithm, which is the combination of North First and South First algorithms [19-23]. Since up to now we focused on the proposal of a routing algorithm aimed at avoiding congestion of the interconnection network, the fault tolerance of the NSF algorithm was not sufficient.

We improved the NSF algorithms and propose an improved North-South First method (NSF-IP, NSF-ImProved), which is a fault-tolerant routing algorithm. And we evaluated both its congestion resistance and its fault tolerance in dynamic communication performance by simulation[23]. However, the fault tolerance of NSF-IP is not enough [23]. It is thought that the fault tolerance can be by changing the routing policy when a packet arrives at a faulty PE.

In this paper we propose the *NSF-FT* (NSF- Fault-Tolerant) algorithm to improve the fault tolerance by improving the NSF-IP algorithm. We evaluate its congestion resistance and fault tolerance by software simulation.

2 2D Torus Network

The structure of a 2D torus network is shown in Figure 1. The network has an *N*×*N* two-dimensional structure, and its four edges are connected by wraparound links. It is used in many parallel computers and some interconnection networks.

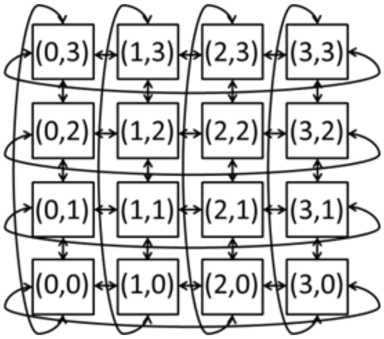


Figure 1: Structure of a 2D-torus network.

Dimension-order routing (DOR) is generally used for deterministic routing on a 2D torus. In DOR, the packet moves on channels in the y direction before moving in the x direction. To avoid deadlocks on a 2D torus, DOR needs two virtual channels (channel-L and channel-H).

- Choose channel-L when starting routing in the y direction.
- When the head of the packet passes through a wraparound link, move the packet to channel-H.
- When the routing in the y direction is completed, move the packet in the x direction; use channel-L regardless of the current channel.
- When the head of a packet passes through a wraparound link, moves the packet to channel-H. Use channel-H until the routing finishes.

The link selection function and channel selection function of dimension-order routing on an N×N torus network are shown in Figures 2 and 3.

```
/ Link Selection Function for Dimension-Order Routing
Link_Select_DOR (cx, cy, dx, dy)
  cx, cy;  // current node  0 ≦ cx, cy ≦ N－1
  dx, dy;  // destination   0 ≦ dx, dy ≦ N－1
{
  if(cy≠dy){                      // dimension Y
     dist_y = (N+dy-cy)%N;
     if(1≦dist_y ≦N/2)           return Y+;
     else                         return Y-;
  }
  else if(cx≠dx){                 // dimension X
     dist_x = (N+dx-cx)%N;
     if(1≦dist_x ≦N/2)           return X+;
     else                         return X-;
  }
  else return OUT;

}
```

Figure 2: Link selection function of the dimension-order routing.

```
// Channel Selection Function for Dimension-Order Routing
Channel_Select_DOR (cd, cc, nd)
  cd;              // current direction    ∈{Y+, Y-, X+, X-}
  cc;              // current channel      ∈{L, H, W}
  nd;              // next direction       ∈{Y+, Y-, X+, X-}
{
  if(cc∈L)  return L;                 // before wrap around
  else if(cc∈W)  return H;            // in wrap around
  else
     if(cd∈{X+,X-} & nd ∈{Y+,Y-})
           return L;     // Y-routing → X-routing
     else   return H;                 // after wrap around
}
```

Figure 3: Channel selection function of the dimension-order routing.

Figures 2 and 3 show the link selection function and channel selection function of DOR on an N×N torus. Here the address of each PE of the torus is shown in terms of its coordinates (*x*,*y*). Moreover, the y-direction channels are written as Y+ and Y−, and the x-direction channels are written as X+ and X−. The four inputs of the link selection function indicate the *x* and *y* coordinates of the present PE, and the *x* and *y* coordinates of the destination PE. The function outputs the link of either X+, X−, Y+, Y− or ``OUT'', which is an output link to a node.

The three inputs of the channel selection function correspond to the current direction, current channel, and direction of the next hop. The current direction and the direction of the next hop have four states, i.e., X+, X−, Y+, and Y−. The current channel has three states, i.e., channel-L (L), channel-H (H), and wraparound channel (W). Although the output has two states (L and H), it unconditionally serves as W when the selected link is a wraparound link.

3 Adaptive Routing of *k*-ary *n*-cube

3.1 Turn Model

The turn model [15] is used by some adaptive routing algorithms [16, 17]. Packet cycles can be prevented by adding a restriction to a path change (turn) of a packet. In the case of a 2D mesh, there are eight kinds of turn, and the various turn model methods put restrictions on two of the eight turns. There is essentially no difference between these methods other than the choice of turn to be restricted. In this paper, we shall incorporate the North First (NF) algorithm and South First (SF) algorithm into one (NSF) and apply it to a 2D torus.

3.2 North First (NF) Routing

The turn model of DOR for a 2D mesh is shown in Figure 4, and the turn model of the NF algorithm is shown in Figure 5. DOR restricts four out of eight turns, whereas the NF algorithm restricts only two, i.e., X− (left, west) → Y+ (upper, north) and the X+ (right, east) →Y+ (upper, north). The South First algorithm, by which the Y− (South) direction is chosen at the beginning of a routing path, is similar.

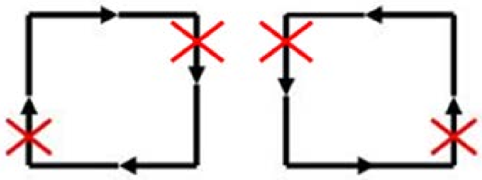


Figure 4: Turn model of dimension-order routing for a 2D-mesh network.

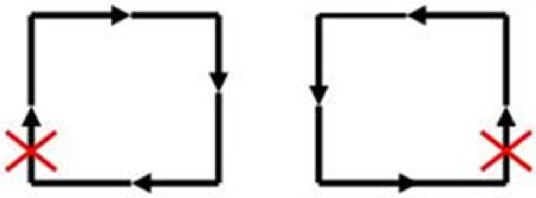


Figure 5: Turn model of north-first (NF) algorithm for 2D-mesh network.

3.3 Application of the Turn Model to a Torus Network

When applying a turn model such as the NF algorithm to a torus network without change, the following differences from the case of a mesh network have to be considered.

(1) In a torus network, when the packet passes through a wraparound channel, a deadlock by cyclic dependency can occur. Therefore, it is necessary to impose an additional restriction.

(2) At least two virtual channels are needed for routing in a torus network. As a result, adaptive routing with higher pliability is attained by applying different turn models to each channel.

An example of a cyclic dependency that occurs in the NF algorithm is shown in Figure 6. Here, packets A–D mutually block a path, causing a deadlock. By contrast, the deadlock does not happen in DOR because packets A and C do not turn in Figure 6. This problem illustrates that in adaptive routing on a torus network it is necessary to take into consideration complicated turn restrictions. Our method deals with this issue by applying the NF and SF algorithms to channel-H and channel-L.

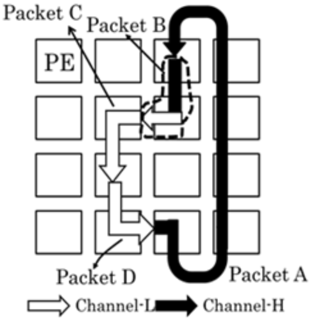


Figure 6: Cyclic in the NF algorithm running on a torus network.

4 NSF Algorithm

When the turn model of NF or SF is applied to a 2D torus directly, cyclic dependency like that shown in Figure 6 occurs because of channel wrap-around. To avoid the deadlock described above, additional restrictions have to be put on the NF and SF algorithms:

1) The SF algorithm does its routing on channel-H. However, a cycle may occur when a path is chosen in which a packet returns to channel-L through channel-H, and for this reason, DOR is carried out instead of the adaptive routing. In DOR, the x-direction channel chosen after a vertical (y-direction) wraparound channel has to be channel-L.

2) The NF algorithm does its routing on channel-L. Because the path of channel-H → channel-L exists after a wraparound channel, the cycle shown in Figure 6 occurs. As shown in Figure 6, though, the cycle can be avoided by adding one more restriction to the other two. Here, three restrictions are put on eight turns, specifically, right → upper, left → upper, and right → lower. This algorithm was named {\it restricted North First (rNF) [19]. The turn model of rNF is shown in Figure 7.

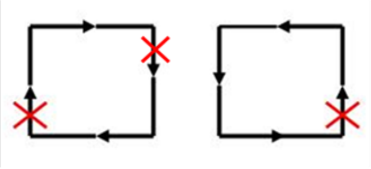


Figure 7: Restricted north-first routing.

From here on, all channels will be described in terms of their dimension $d \in \{X, Y\}$, direction $\delta \in \{+, -\}$, channel type $c \in \{L, W, H\}$, i.e., $(d\delta, c)$. X means X dimension, Y means Y dimension, and L L, W, and H means channel-L, wraparound channel, and channel-H. $(d+, c)$ and $(d-, c)$ will be shown as a set, written as $(d\pm, c)$.

4.1 Routing Algorithm of NSF Routing

In our method, the restricted NF algorithm is carried out in channel-L and the SF algorithm is carried out in channel-H. Since $(Y+, L)$ and $(Y+, H)$ are respectively used in the restricted NF algorithm and SF algorithm, we will study cases in which $(Y+, c)$ is used and not used and cases in which the horizontal and vertical wraparound channels are used and not used.

Figures 8 and 9 show the link selection function and channel selection function of the proposed method on an $N \times N$ torus. As in the case of DOR in Figure 2, the link selection function outputs X+, X−, Y+, Y−, or ``OUT'' (an output link to a node). The proposed method needs the ``current channel'' as an input in addition to the inputs of DOR.

The channel selection policy varies depending on whether $(Y+, c)$ is used or not. If it is used, adaptive routing is carried out only when the wraparound channels are not to be used from that point on. If $(Y+, c)$ is not used, the restricted NF algorithm is carried out from the source PE until the first wraparound channel (or destination PE) is reached.

The algorithm of Figure 8 first determines, in procedure ①, whether the wraparound links of X and Y are used. In this case, the determination is based on the X and Y coordinates of the source and destination PEs as follows:

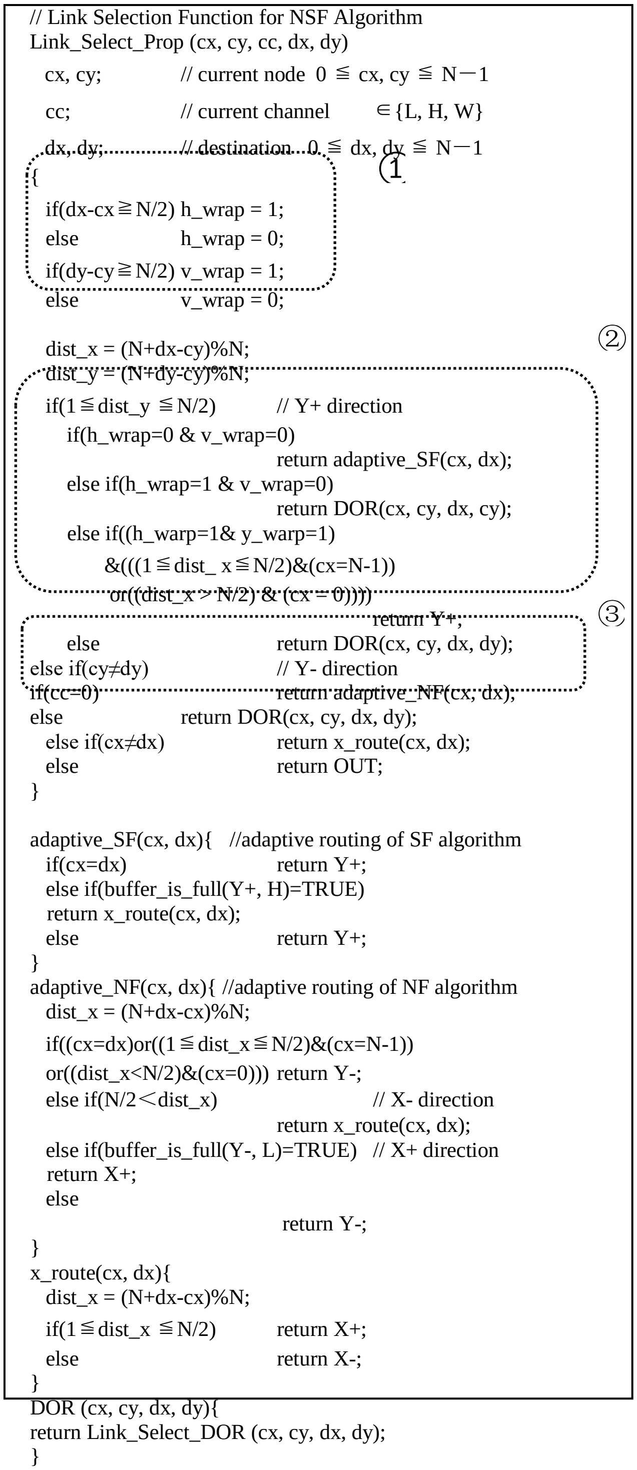
- When the difference between the X coordinates of the current PE and destination PE is less than *N*/2, h_wrap is set to 0 because the wraparound channel of the x direction is not straddled. If not, h_wrap is set to 1.
- When the difference between the Y coordinates of the current PE and destination PE is less than *N*/2, v_wrap is set to 0. If not, v_wrap is set as 1.

Next, the link is chosen on the basis of whether the Y+ channel (channel (Y+,*c*)) is used or not, as follows:

- When $(Y+, c)$ is used, the procedure ② is carried out. In this case, since the restricted NF in channel-L is equivalent to DOR, only the adaptive routing of the SF method in channel-H is carried out. If neither wraparound channel is used in going from the current PE to the destination, the packet can be sent over channel-H and routing can be continued. Thus, adaptive routing can be carried out with the SF method. The only other case in which channel-H may be used is after the packet has passed through a vertical wraparound channel $(Y+, W)$ and is due to pass through a horizontal wraparound channel $(X\pm, W)$. Even in this case, it is thought that adaptative routing using the SF method is possible. However, since it is difficult to prove that is deadlock-free, only the X-directional routing is carried out from $(Y+, W)$ to $(X\pm, W)$ and SF is applied after the packet has passed through $(X\pm, W)$. So the load concentrates in part of the network (near PE(0, 0)).

- When $(Y+, c)$ is not used, the procedure ③ is carried out. Since the SF method in channel-H is equivalent to DOR, only the adaptive routing of the restricted NF method in channel-L is carried out. In this case, the following restriction is added in order to make the order of passage in a wraparound channel into $(Y-, W)$→$(X\pm, W)$.
  - Restricted NF is carried out only when$(Y-, W)$ is not be passed from the current PE to the destination or the next channel is not $(X\pm, W)$. DOR is carried out otherwise.

Besides the three inputs of the channel selection function of DOR in Figure 3, the channel selection function needs four inputs that indicate the *x* and *y* coordinates of the source and destination PEs. These new inputs can be used to judge the possibility of the packet passing through a wraparound channel. Based on the judgment, channel-H is chosen only when the wraparound channel is not to be used and $(Y+, c)$ is to be used. DOR is carried out otherwise. As in the case of DOR, the output has two states, L and H. However, an output unconditionally serves as W when the selected link is a wraparound link.



```
// Link Selection Function for NSF Algorithm
Link_Select_Prop (cx, cy, cc, dx, dy)
  cx, cy;           // current node  0 ≦ cx, cy ≦ N－1
  cc;               // current channel      ∈{L, H, W}
  dx, dy;           // destination  0 ≦ dx, dy ≦ N－1
{
  if(dx-cx≧N/2) h_wrap = 1;
  else          h_wrap = 0;
  if(dy-cy≧N/2) v_wrap = 1;
  else          v_wrap = 0;

  dist_x = (N+dx-cy)%N;
  dist_y = (N+dy-cy)%N;
  if(1≦dist_y ≦N/2)          // Y+ direction
     if(h_wrap=0 & v_wrap=0)
                              return adaptive_SF(cx, dx);
     else if(h_wrap=1 & v_wrap=0)
                              return DOR(cx, cy, dx, cy);
     else if((h_warp=1& y_warp=1)
          &(((1≦dist_ x≦N/2)&(cx=N-1))
          or((dist_x > N/2) & (cx = 0))))
                                          return Y+;
     else                     return DOR(cx, cy, dx, dy);
else if(cy≠dy)                // Y- direction
if(cc=0)                      return adaptive_NF(cx, dx);
else              return DOR(cx, cy, dx, dy);
  else if(cx≠dx)              return x_route(cx, dx);
  else                        return OUT;
}

adaptive_SF(cx, dx){   //adaptive routing of SF algorithm
  if(cx=dx)                   return Y+;
  else if(buffer_is_full(Y+, H)=TRUE)
  return x_route(cx, dx);
  else                        return Y+;
}
adaptive_NF(cx, dx){ //adaptive routing of NF algorithm
  dist_x = (N+dx-cx)%N;
  if((cx=dx)or((1≦dist_x≦N/2)&(cx=N-1))
  or((dist_x<N/2)&(cx=0))) return Y-;
  else if(N/2＜dist_x)                   // X- direction
                              return x_route(cx, dx);
  else if(buffer_is_full(Y-, L)=TRUE)   // X+ direction
  return X+;
  else
                               return Y-;
}
x_route(cx, dx){
  dist_x = (N+dx-cx)%N;
  if(1≦dist_x ≦N/2)          return X+;
  else                        return X-;
}
DOR (cx, cy, dx, dy){
return Link_Select_DOR (cx, cy, dx, dy);
}
```

Figure 8: Link selection function of the proposed algorithm.

```
// Channel Selection Function for NSF Algorithm
Channel_Select (cx, cy, dx, dy , cd, cc, nd)
   cx, cy;  // current node 0 ≦ cx, cy ≦ N－1
    dx, dy;             // destination  0 ≦ dx, dy ≦ N－1
    cd;                 // current direction ∈{Y+, Y-, X+, X-}
    cc;                 // current channel  ∈{L, H, W}
    nd;                 // next direction  ∈{Y+, Y-, X+, X-}
{
  if(dx-cx≧N/2)                    h_wrap = 1;
  else                             h_wrap = 0;
  if(dy-cy≧N/2)                    v_wrap = 1;
  else                             v_wrap = 0;

    dist_y = (N+dy-cy)%N;
  if((1≦dist_y ≦N/2)                          // Y+ direction
          & (h_wrap=0 & v_wrap=0))            return H;
  else                                        // Others
     return DOR_Channel (cd, cc, nd);
}

DOR_Channel (cd, cc, nd){
  return Channel_Select_DOR (cd, cc, nd);
}
```

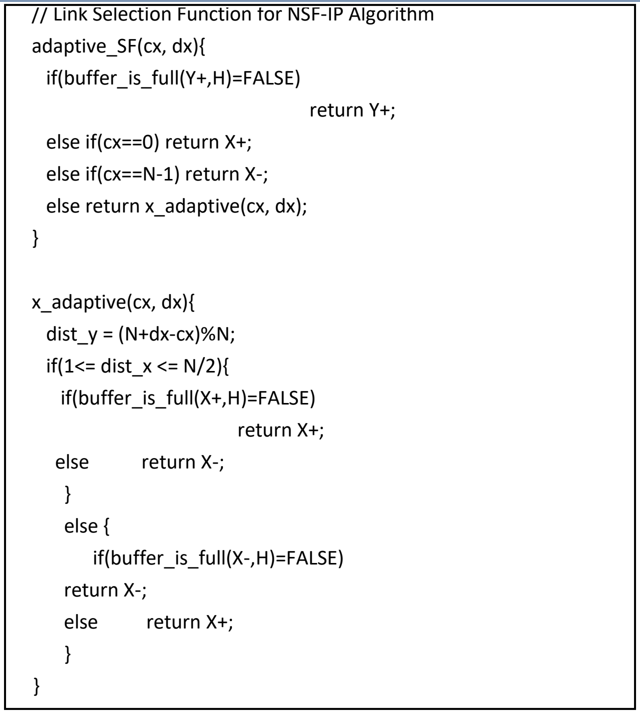
Figure 9: Channel selection function of the NSF algorithm.

4.2 NSF-IP Routing

The NSF-IP method proposed in this paper has improved fault tolerance. In this method, the SF method executed with channel-H is changed to a method that does not necessarily pass the shortest path. The routing algorithms of the SF method part before and after the change are shown in Figures 10 and 11, respectively.

```
// Link Selection Function for NSF Algorithm

adaptive_SF(cx, dx){   //adaptive routing of SF algorithm

     if(cx=dx)                          return Y+;

     else if(buffer_is_full(Y+, H)=TRUE)

     return x_route(cx, dx);

     else                               return Y+;

}
```

Figure 10: SF routing of the link selection function of the NSF algorithm.



```
// Link Selection Function for NSF-IP Algorithm
adaptive_SF(cx, dx){
  if(buffer_is_full(Y+,H)=FALSE)
                                    return Y+;
  else if(cx==0) return X+;
  else if(cx==N-1) return X-;
  else return x_adaptive(cx, dx);
}

x_adaptive(cx, dx){
  dist_y = (N+dx-cx)%N;
  if(1<= dist_x <= N/2){
    if(buffer_is_full(X+,H)=FALSE)
                          return X+;
   else       return X-;
    }
    else {
        if(buffer_is_full(X-,H)=FALSE)
    return X-;
    else       return X+;
    }
}
```

Figure 11: SF routing of the link selection function of the NSF-IP algorithm.

The main difference between before and after improvement is that adaptive routing that selects both the - direction and the + direction is performed in the horizontal direction (X direction). This enables routing avoiding congestion and failure regardless of whether it is the shortest route or not.

4.3 Deadlock Avoidance

A channel dependency graph is drawn in order to prove that the routing algorithm described in the previous section does not cause a deadlock [11, 24, 25]. The channel dependency graph is a directed graph in which nodes (channels) with dependencies are connected by an arrow. Specifically, nodes (channels) with dependencies are pairs of nodes (channels) in which a packet may be directly transmitted and received while routing.

First, the channel dependency graph is drawn. Then, each channel is numbered. If it is proved that the channel numbers are in ascending order (or descending order) in the direction of the arrows of the channel dependency graph, deadlock does not happen. In such a case, the channels are said to have an ordered relation and the corresponding channel will not cause a cyclic dependency.

A routing algorithm based on the turn model generally assigns numbers to the output channels from the PE on the basis of the PE address. As mentioned above, a 2D torus network has two virtual channels. Accordingly, the following 4-dimensional channel numbers CN are given to the 4 links × 2 channels (=8 channels) in each PE of an $N \times N$ torus.

Here, $x\ (0 \leq x \leq N-1)$ and $y\ (0 \leq y \leq N-1)$ are the x and y coordinates of the PE address, $d \in \{Y+, Y-, X+, X-\}$ is the direction of the channel, and $ch \in \{\mathrm{L}, \mathrm{H}, \mathrm{W}\}$ is the type of channel. g_m, c_1, g_s and c_2 denote the main group, first coordinate, sub group, and second coordinate, respectively. These values are numbered as follows:

$$CN(x, y, d, ch) = (g_m, c_1, g_s, c_2) \quad (1)$$

- Main Group g_m

The channel direction order is based on d and ch. Table 1 lists the values of g_m.

Table 1: Values of g_m determined by g_m and ch.

d	ch	g_m
Y+	L,W	0
Y-, X-	L,W	1
Y-	H	
X+	L,W	2
X+, X-, Y+	H	3

- First Coordinate c_1

The value c_1 is based on g_m (see Table 2).

Table 2: Value of c_1 determined by g_m.

g_m	c_1
0	y
1	$N - x$
2	0
3	y

- Sub Group g_s

The sub group g_s value determines the order of channels in the same g_m. g_s is set to 0 at g_m=0 and g_m=2. Table 3 lists the values of g_s at g_m =1 and g_m=3.

Table3: Value of g_s

g_m	d	ch	g_s
1	Y-	L,W	0
	Y-	H	1
	X-	L,W	2
3	X+, X-	H	0
	Y+	H	1

- Second Coordinate c_2

The second coordinate c_2 determines the order of the same d and ch. The c_2 values of , $N - y$, x , and $N - x$ correspond to d values of $Y+, Y-, X+$, and $X-$.

Figure 12 illustrates the channel numbers for each channel. Here, deadlocks can be avoided because channel numbers will be in ascending order through a routing path [20][21].

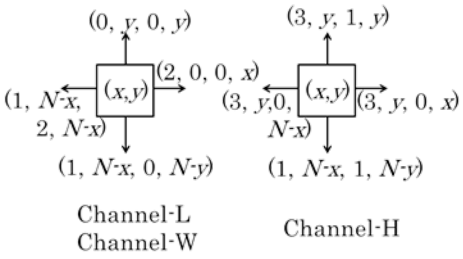


Figure 12: The channel number.

5 NSF- Fault-Tolerant (NSF-FT) Algorithm

The NSF-FT method improves the fault tolerance by moving the head of packet to channel-H when the following condition is satisfied. Examples of avoidance of faulty PE on NSF-FT are shown in Figure 13. Figure 13 shows a situation in which a packet is transferred from the source PE to the destination PE. In the conventional method, channel-L is firstly used and the transfer by path ① is carried out. When there is a faulty point as shown in the Figure 13, since the route of the packet is blocked by the failure point at point α, it is impossible to reach the destination PE. Therefore, when the next hop PE is in a faulty state, the head of the packet is compulsorily moved to channel-H. This enables adaptive routing based on the SF method. As a result, the path of the packet changes as shown in the figure, and the path ② is chosen by using channel-H and packet can avoid the faulty PEs.

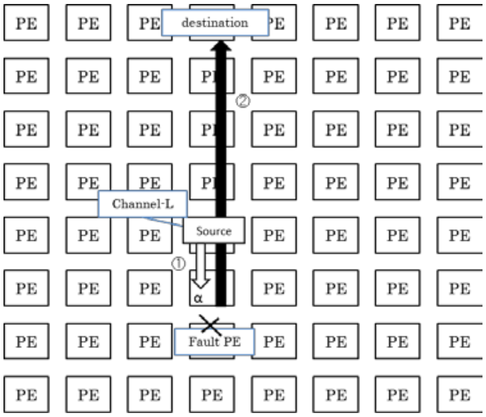


Figure 13: Examples of assumed failure patterns.

Figure 14 shows the link selection function of the NSF-FT algorithm. "Out_NSF_IP" is the selection result of the link selection function according NSF_IP. "Out_NSF_IP_H" is the selection result of the link selection function by SF algorithm in channel-H. In addition, the status (Out_NSF_IP) returns the state of the link selection destination PE by NSF_IP. The value of “Fault” is returned in the case where the PE is faulty. In the algorithm of the figure, firstly the PE of the link selection destination by NSF_IP is confirmed. In the case where the PE is not faulty, the selection result of the link selection function by NSF_IP is selected. If the PE is faulty, routing by the SF algorithm is carried out.

```
// The link Selection Function of the NSF-FT Algorithm.
Link_Select_NSF_FT(cx, cy, cc, dx, dy)
{
  Out_NSF_IP = Link_Select_NSF_IP (cx, cy, cc, dx, dy);
  if(dy>cy)  Out_NSF_IP_H = adaptive_SF(cx, dx)
     else       Out_NSF_IP_H = DOR(cx, cy, cc, dx, dy);

  if(status(Out_NSF_IP) == fault)
    return Out_NSF_IP_H;
  else
    return Out_NSF_IP;
}
```

Figure 14: Link selection function of the NSF-FT algorithm.

Figure 15 shows a channel selection function of NSF-FT. "Ch_NSF_IP" in the figure shows the selection result of the channel selection function by NSF_IP. In the channel selection function shown in the figure, firstly the PE of the link selection destination by NSF_IP is confirmed. In the case where the PE is not faulty, the selection result of the channel selection function by NSF_IP is selected.

If the PE is faulty, channel-H is selected.

```
// The Channel Selection Function of the NSF-FT Algorithm.
Channel_Select_NSF_FT(cx, cy, cc, dx, dy)
{
  Out_NSF_IP = Link_Select_NSF_IP(cx, cy, cc, dx, dy);
  Ch_NSF_IP = Channel_Select(cx, cy, dx, dy, cd, cc, nd);

  if(status(Out_NSF_IP)==fault)
    return H;
  else
    return Ch_NSF_IP;
}
```

Figure 15: Channel selection function of the NSF-FT algorithm.

5.1 Deadlock Avoidance of NSF-FT

NSF-FT is the algorithm that the packet moves from channel-L or channel-W to channel-H. From Figure 12, the case that the channel number has descending order by moving from channel-L to channel-H is only the case that the transfer is from channel $(X+, L)$ or $(X+, W)$ to $(Y-, H)$. In such cases, the channel numbers become from $(2, 0, 0,\ x)$ to $(1,\ N - x,\ 1,\ N - y)$. Therefore, it has to be proven that such case does not occur.

The cases that the transfer from channel $(X+, L)$ or $(X+, W)$ to $(Y-, H)$ are as following three cases:

① The path of $(X+, L) \rightarrow (Y-, L) \rightarrow (Y-, H)$.

② The path from $(X+, L)$ to $(Y-, H)$ via $(X+, W)$ and $(X+, H)$.

③ At $(X+, L)$ or $(X+, W)$, the packet moves to channel-H because of the blocking of faulty PE. In this case, $(Y-, H)$ is used in channel-H.

From the turn model of Figure 7, the turn from right to lower is not permitted in the rNF method in channel-L. Therefore, the transmission from $(X+,L)$ to $(Y-,L)$ such as ① does not occur. Also, the turn from right to lower is not permitted in the SF method in channel-H. Therefore, the transmission from $(X+,H)$ to $(Y-,H)$ such as ② does not occur.

From the above reason, the turn from right to lower is not permitted in every channel of channel-L, channel-W, and channel-H. Therefore, in the case that the destination PE is at lower-right from the current PE, the Y coordinate of the network is aligned first and then $(X+,c)$ are used. In such cases, $(Y-,c)$ cannot be used after $(X+,L)$ or $(X+,W)$, so the transmission such as ③ does not occur. Thus, the transmission from $(X+,L)$ or $(X+,W)$ to $(Y\pm,c)$ such as ③ does not occur.

Moreover, channel numbers seem not to be in ascending order because the value of Sub Group g_s becomes small in the case from $(X-,L)$ or $(X-,W)$ to $(Y-,H)$ (channel number becomes from $(1, N-x,2,N-x)$ to $(1,N-x,1,N-y)$). However, the value of first Coordinate c_1 increases when $(X-,L)$ is transmitted. Therefore, the order relation is maintained.

6 Evaluation of Congestion Resistance

A communication performance was evaluated by a software simulator that transmits 50000 cycles of packets for a 16 × 16 torus network with 256 PEs, and the evaluation results were shown. In the software simulator used in this evaluation, functional modules of routers and processor cores such as crossbar switch, FIFO, multiplexer, demultiplexer, and control unit are faithfully implemented, and is written by C language [26, 27].

Evaluation by Uniform communication pattern was carried out using this simulator. The Uniform communication pattern is a simulation performed randomly for both the start point and the end point when sending a packet.

The dynamic communication performance of an interconnection network is characterized by message latency and network throughput. Message latency refers to the time elapsed from the instant when the first flit is injected into the network from the source to the instant when the last flit of the message is received at the destination. Average transfer time is the average value of the message latency for all packets. Network throughput refers to the maximum amount of information delivered per unit of time through the network. It is the average value of the number of flits that a PE receives in each clock cycle. In the evaluation of dynamic communication performance, flocks of messages are sent in the network to compete for the output channels. Packets are transmitted by the request-probability *r* during *T* clock cycles, and the number of flits that reached at destination PE and their transfer time are recorded. Then the average transfer time and throughput are calculated and plotted with average transfer time on the horizontal axis and throughput on the vertical axis. The process of performance evaluation is carried out with changing the request-probability *r*.

Figure 16 shows the average transfer time for network throughput obtained by simulation. The horizontal axis of the graph is throughput and the vertical axis is average transfer time. As shown in Figure 16, it was revealed that the NSF, NSF-IP, and NSF-FT methods have higher throughput than the DOR. In addition, when the NSF-IP, NSF-FT, and NSF methods were compared, the throughput was almost the same, and no significant performance degradation due to proposed method was observed. In a communication pattern in which the interconnection network is crowded as a whole like random communication, the effect of avoiding congestion is limited by adaptive routing in a method that does not use additional virtual channels.

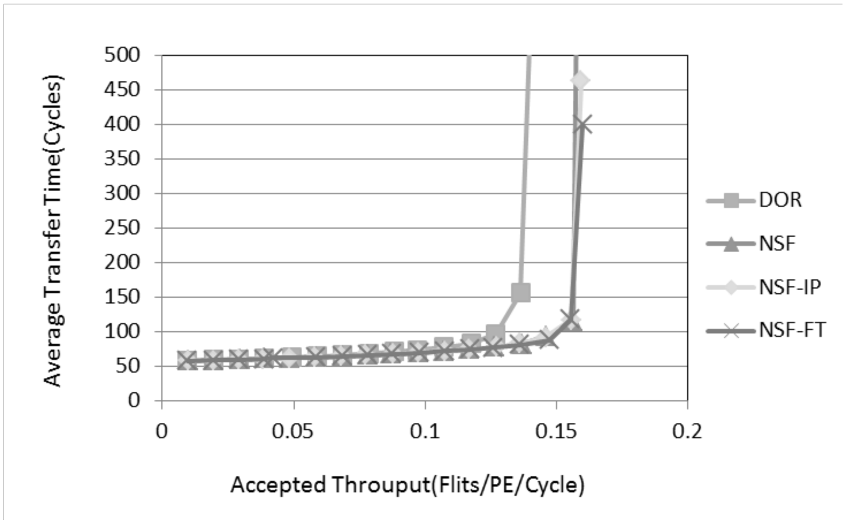


Figure 16: Result for uniform traffic.

7 Evaluation of Fault Tolerance

In the same way as in the previous section, communication performance was evaluated by a software simulator that transmits packets for a 16 × 16 torus network with 256 PEs. On the communication pattern of the evaluation of this section, one “session” is defined as “All PEs in the network send one packet to another PE. In this case, packets blocked by faulty PEs remain in the network. In this evaluation, the communication of 1, 3, and 5 sessions were carried out in the network with some faulty patterns, and the number of non-arrival packets (packets that do not arrive at the destination because they were blocked by a faulty PE) with DOR，NSF，NSF-IP，and NSF-FT were tracked. We tracked the results of 10 times simulation and plotted the average values of the number of non-arrival packets in Tables 4, 5, and 6. How much the proposed algorithm reduces the number of non-arrival packets can be determined by tracking the non-arrival packets. “Loop 1”, “Loop 3”, and “Loop 5” in the following subsections are the communication patterns of 1, 3, and 5 sessions.

7.1 Four PEs Near the Center Are Faulty

Table 4 shows the results of evaluation of “four PEs near the center are faulty PEs”. Since four PEs are set as faulty PEs, $256 - 4 = 252$ packets are sent between PEs per loop. In the “loop 1”, the numbers of non-arrival packets for NSF, NSF-IP, and NSF-FT were lower than that for DOR and the numbers of non-arrival packets for NSF-IP and NSF-FT were a little lower than that for NSF. In the “loop 3” and “loop 5”, the number of non-arrival packets for NSF-FT was slightly lower than that for NSF-IP.

The “loop 3” results are more than ten times larger than the “loop 1” results. This is the result of another packet being blocked in the path that was blocked by the faulty node. As a result, it is thought that such a result was obtained because a large amount of unarrived packets remains in the Interconnection network. The improvement of NSF-IP compared with DOR and NSF is due to removal of constraints of the shortest path, which increased the number of selectable paths. And the reason NSF-FT is slightly better than the other methods is that it avoids faulty points by the method described in Figure 13.

Table 4: Number of non-arrival packets on 4 central points of failure.

Algorithm	Loop 1	Loop 3	Loop 5
NSF-FT	14.8	179.0	639.8
NSF-IP	14.7	188.3	648.8
NSF	16.8	204.9	681.2
DOR	21.1	251.5	742.4

7.2 Four Corner PEs Are Faulty

Table 5 shows the results of evaluation of average non-arrival packets where four PEs at four corners including wraparound link are faulty PEs. In all the loops 1, 3, and 5, the number of non-arrival packets of the proposed method is decreasing, and the effect by implementing the proposed method is shown.

Table 5: Number of non-arrival packets on 4 corner failure.

Algorithm	Loop 1	Loop 3	Loop5
NSF-FT	13.1	182.8	652.5
NSF-IP	15.9	213.4	698.2
NSF	15.7	210.4	693.4
DOR	19.2	248.6	740.9

7.3 PEs Are Randomly Broken

It was assumed that 1, 2, 4, 8, and 16 PEs are randomly broken in a 16 × 16 two-dimensional torus network simulator having 256 PEs, and the number of non-arrival packets were measured using the DOR, NSF, NSF-IP, and NSF-FT algorithms. The average numbers of non-arrival packets in loops 1, 3, and 5 are shown in Figures 17, 18, and 19. The vertical axis of each graph is the average number of non-arrival packets. Table 6 shows the ratio of average number of non-arrival packets in NSF-FT compared to other methods. The value of "DOR → FT" in Table 6 is (number of non-arrival packets for NSF-FT) / (number of non-arrival packets for DOR). Likewise, "IP → FT" and "NSF → FT" are (non-arrival packet number for NSF-FT) / (number of non-arrival packets for NSF-IP) and (number of non-arrival packets for NSF-FT) / Packet number). Shaded areas in the tables are portions where the improvement by NSF-FT are not seen.

Although the dispersion of non-arrival packets in NSF-FT and NSF-IP of 8 faulty PEs in Figure 17 and NSF-FT and NSF-IP of 16 faulty PEs in Figure 18 were large, the number of arrival packets of NSF-FT was improved in all methods in others. The reason the dispersions were large at two points in Figures 17 and 18 is that NSF-FT caused more packets to flow into the channel-H than other methods did. It causes the congestion of channel-H, and then it causes the increase of the number of cycles and the increase of the number of non-arrival packets. From the above, one can infer that NSF-FT improves fault tolerance in the case that the number of faulty PE is not large or in a situation of congestion or moderate congestion.

The average transfer times required for loops 1, 3, and 5 are shown in Figures. 20, 21, and 22. The number of cycles required for transfer is plotted as the vertical axis. Figures 20, 21, and 22 show that the average transfer times were slightly larger in DOR and NSF-FT.

The large average transfer time of DOR is thought to be due to the long waiting time of packets in the network because there are few choices regarding the course of the packet. The reason the average transfer time of NSF-FT becomes longer is the longer transfer path itself. As described in Figure 13 in section 5, NSF-FT tends to lengthen the transfer path because it is a method of changing the rule of link selection by the faulty PE. This can be considered a cause of the above-mentioned dispersion.

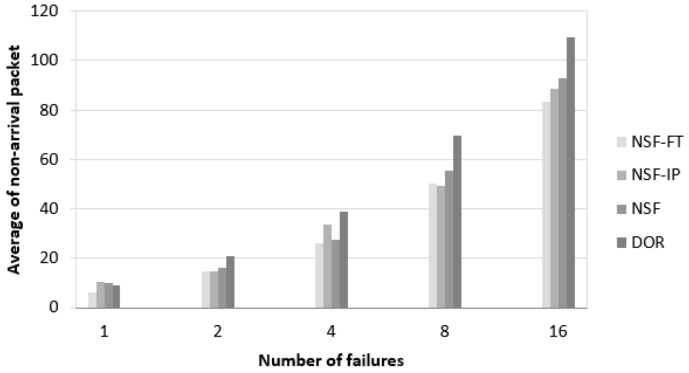


Figure 6: Average of non-arrival packet in loop 1.

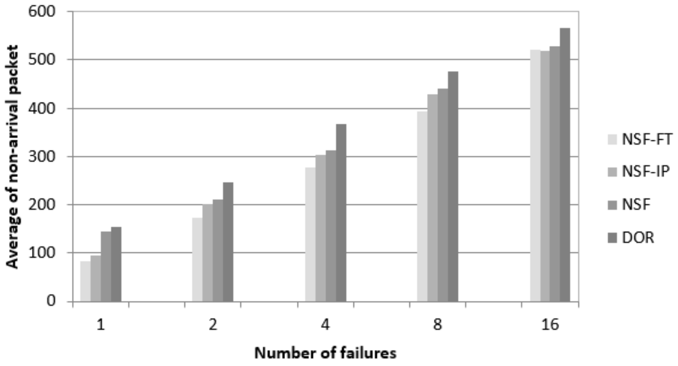


Figure 7: Average of non-arrival packet in loop 3.

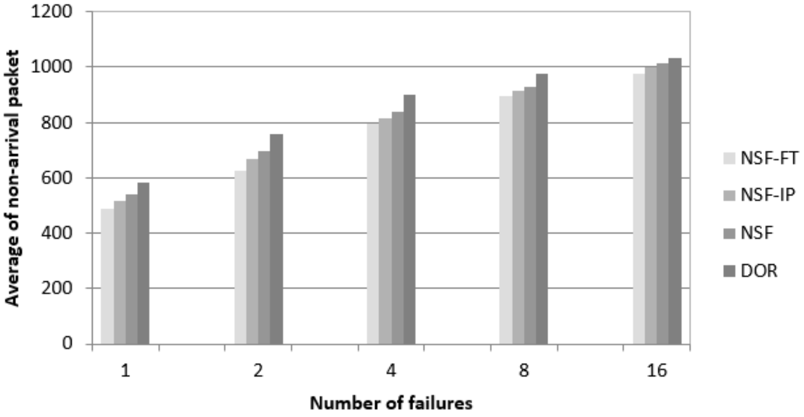
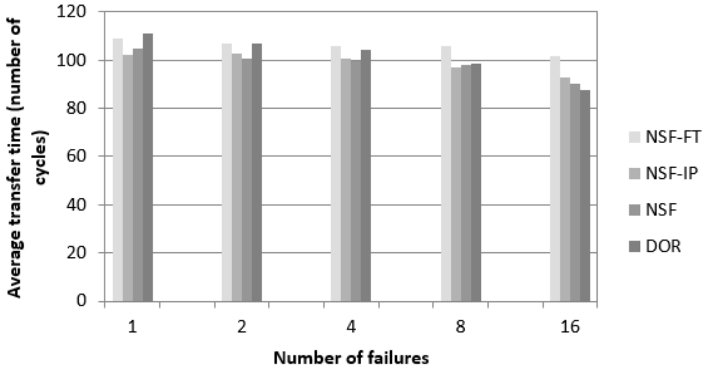


Figure 8: Average of non-arrival packet in loop 5.



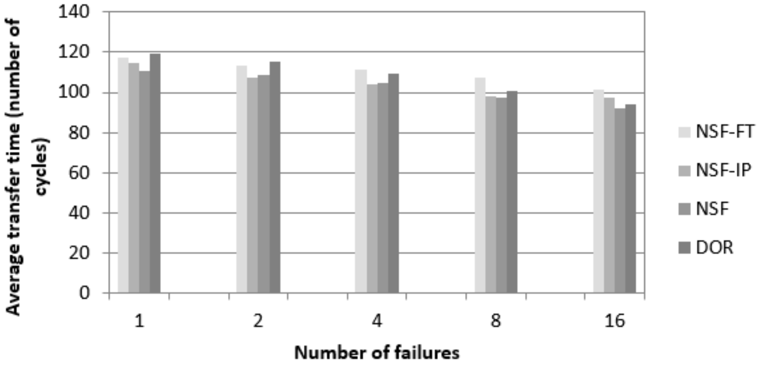


Figure 9: Average transfer time on loop 3.

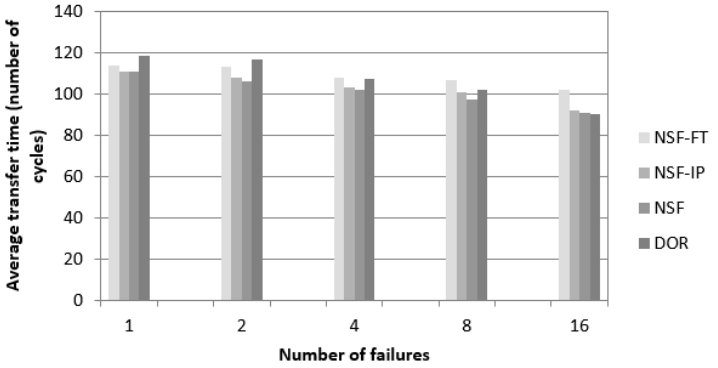


Figure 10: Average transfer time on loop 5.

Table 6: The ratio of NSF-FT average non-arrival packets compared to other methods.

	Number of faulty PEs	Loop 1	Loop 3	Loop 5
DOR→FT	1	69.2%	54.0%	83.2%
	2	70.5%	70.1%	82.2%
	4	67.0%	75.1%	88.6%
	8	72.1%	82.3%	92.2%
	16	76.1%	92.0%	94.3%
IP→FT	1	60.0%	87.2%	94.0%
	2	99.3%	85.9%	93.2%
	4	77.6%	90.8%	97.7%
	8	101.8%	91.7%	98.0%
	16	93.8%	100.1%	97.3%
NSF→FT	1	64.3%	57.4%	89.9%
	2	91.4%	82.5%	89.8%
	4	94.5%	88.4%	94.9%
	8	91.1%	88.9%	96.5%
	16	89.9%	98.8%	96.3%

8 Conclusion

In this paper we proposed the NSF-FT method, which is a routing algorithm that improves the fault-tolerance of NSF method based on the turn model and is applicable to a 2D torus. And its congestion resistance and fault tolerance were evaluated by software simulation.

As for the congestion resistance, the throughput obtained with NSF-FT is better than that obtained with DOR. And even compared with NSF and NSF-IP, there was no throughput reduction due to NSF-FT implementation.

Regarding the evaluation of fault tolerance assuming a random faulty PEs, it is clear that the packet arrival rate in congested situations has been increased by the NSF-FT method. As the result, the fault tolerance was improved.

We are planning to extend the proposed method to the *n* -dimensional torus because it can be applied to any two dimensions in an *n*-dimensional torus network. Specifically, the following algorithm can be considered for the *n*-dimensional torus, whereas the routing order of 0 dimension → 1 dimension→ 2 dimension → ... (N−1)-dimension is carried out in DOR.

- At first, the NSF method and so on are applied for dimensions 0 and 1.
- Next, the NSF method or the like is similarly applied for dimensions 2 and 3.
- And then, the NSF method and so on are applied until dimension N−1.

Furthermore, future work will further improve fault tolerance by combining the proposed method with one in which the rectangular areas of faulty PEs are handled as faulty area [28].

REFERENCES

[1]. J. Y. Ngai and C. L. Seitz, A framework for adaptive routing in multicomputer networks, ACM SIGARCH Computer Architecture News, vol. 19, no. 1, pp. 6–14, 1991.

[2]. T. Schonwald, J. Zimmermann, O. Bringmann, and W. Rosenstiel, Fully Adaptive Fault-Tolerant Routing Algorithm for Network-on-Chip Architectures, Digital System Design Architectures, Methods and Tools, pp. 527–534, 2007.

[3]. M. M. Hafizur Rahman, Yukinori Sato, Yasuyuki Miura, and Yasushi Inoguchi, Dynamic Communication Performance of a Hierarchical 3D-Torus Network, IASTED, In 10th International Conference Parallel and Distributed Computing and Networks (PDCN 2011), 2011.

[4]. Y. Miura and S. Horiguchi, An Adaptive Routing for Hierarchical Interconnection Network TESH, Proc. of the Third International Conference on Parallel And Distributed Computing, Applications and Technologies, pp. 335–342, 2002.

[5]. Y. Miura, Masahiro Kaneko, M. M. Hafizur Rahman, and Shigeyoshi Watanabe, Adaptive Routing Algorithms and Implementation for TESH Network, Communications and Network (CN), vol. 5, no. 1, pp. 34–49, 2013.

[6]. W. J. Dally, Performance Analysis of k-ary n-cube Interconnection Networks, IEEE Trans. on Computers, vol. 39, no. 6, pp. 775–785, 1990.

[7]. W.J. Dally and Hiromichi Aoki, Deadlock-Free Adaptive Routing in Multicomputer Networks Using Virtual Channels, IEEE Trans. on Parallel and Distributed Systems, vol. 4, pp. 466–475, 1993.

[8]. M. P. Merlin and J. P. Schweitzer, Deadlock Avoidance in Store-and-Forward Networks-1: Store and Forward Deadlock, IEEE Trans. On Communications, vol. COM-28, no. 3, pp. 345–354, 1980.

[9]. W. J. Dally and C. L. Seitz. Deadlock-Free Message Routing in Multiprocessor Interconnection Networks. IEEE Trans. on Computers, vol. C-36, no. 5, pp. 547–553, 1987.

[10]. C. S. Yang, Y. M. Tsai, and Y. L. Tsai, Adaptive Routing in k-ary n-cube Multicomputers, Proc. of 1996 International Conference on Parallel and Distributed Systems(ICPADS'96), pp. 404–411, 1996.

[11]. J. Duato, A New Theory of Deadlock-Free Adaptive Routing in Wormhole Networks, IEEE Trans. on Parallel and Distributed Systems, vol. 4, no. 12, pp. 1320–1331, 1993.

[12]. D. H. Linder and J. C. Harden, An adaptive and fault tolerant wormhole routing strategy for k-ary n-cubes, IEEE Trans. on Computers, vol. C-40, no.1, pp. 2–12, 1991.

[13]. R. S. Ramanujam and Bill Lin, Destination-based adaptive routing on 2D mesh networks, 2010 ACM/IEEE Symposium onArchitectures for Networking and Communications Systems (ANCS), pp. 1–12, 2010.

[14]. C. J. Glass and L. M. Ni, Maximally Fully Adaptive Routing in 2D Meshes, Proc. of The 19th International Symposium on Computer Architecture, pp. 278–287, 1992.

[15]. C. J. Glass and L. M. Ni, The Turn Model for Adaptive Routing, Proc. of The 25th Annual International Symposium on Computer Architecture, pp. 441–450, 1998.

[16]. Jie Wu, A Fault-tolerant and Deadlock-free Routing Protocol in 2D Meshes Based on Odd-even Turn Model, IEEE Trans. on Computers, vol. 52, no. 9, pp. 1154–1169, 2003.

[17]. A. Jouraku, M. Koibuchi, and H. Amano, An Effective Design of Deadlock-Free Routing Algorithms Based on 2D Turn Model for Irregular Networks, IEEE Trans. on Parallel and Distributed Systems, vol. 18, no. 3, pp. 320–333, 2007.

[18]. W. J. Dally, Virtual-Channel Flow Control, IEEE Trans. on Parallel and Destributed Systems, vol. 3, no. 2, pp. 194–205, 1992.

[19]. K. Matoyama, Y. Miura, and S. Watanabe, Adaptive routing of the 2-D torus network, Proc. of FIT2009, RC-005, 2009 (in Japanese).

[20]. Y. Miura, K. Shimozono, K. Matoyama, and S. Watanabe, An Adaptive Routing of the 2-D Torus Network Based on Turn Model, Proc. of 4th International Workshop on Advances in Networking and Computing, pp. 587-591,December 2013.

[21]. Yasuyuki Miura, Kentaro Shimozono, Kazuya Matoyama, and Shigeyoshi Watanabe, The Static and Dynamic Performance of an Adaptive Routing Algorithm of 2-D Torus Network Based on Turn Model, Proc. of the 2014 International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA' 14), pp. 114–120, July 2014.

[22]. Yasuyuki Miura, Kentaro Shimozono, Naohisa Fukase, Shigeyoshi Watanabe, and Kazuya Matoyama, An Adaptive Routing Algorithm of 2-D Torus Network Based on Turn Model: The Communication Performance, International Journal of Networking and Computing (IJNC), pp. 223–238,January 2015.

[23]. Tsukasa-Pierre Nakao and Yasuyuki Miura, The Study on Adaptive Routing Algorithm of 2-D Torus Network with Fault Tolerance, IEICE Technical Report (FIIS64),October 2017.

[24]. J. Duato A Necessary and Sufficient Condition for Deadlock-Free Adaptive Routing Wormhole Networks, Proc. of the International Conference on Parallel Processing, vol. 1, pp. 142-149, 1994.

[25]. E. Fleury and P. Fraigniaud, A General Theory for Deadlock Avoidance in Wormhole-Routing Networks, IEEE Trans. on Parallel and Distributed Systems, vol. 9, no. 7, pp. 626–638, 1998.

[26]. Naohisa Fukase, Yasuyuki Miura, Shigeyoshi Watanabe, and M. M. Hafizur Rahman, The Performance Evaluation of a 3D Torus Network Using Partial Link-Sharing Method in NoC Router Buffer, IEICE Trans. on Information and Systems, vol. E100–D, no. 10, October 2017.

[27]. Naohisa Fukase, Yasuyuki Miura, and Shigeyoshi Watanabe, Link-Sharing Method of Buffer in Router Circuit of Direct-Connection Network, IEEJ Trans EIS, vol. 132, no. 10, pp. 1675–1688, 2012.

[28]. N. Aosaka, Y. Fukushima, M. Fukushi, and M. Kameyama, Fault-Tolerant Cogestion-Avoidance Routing for 2D-Mesh Network-on-Chip,Technical report of IEICE. FIIS10, no. 271, March 2010.