Deadlock-Free Adaptive Routing in Meshes Based on Cost-Effective Deadlock Avoidance Schemes *

Dong Xiang
School of Software
Tsinghua University
Beijing 100084, China

Yueli Zhang
School of Software
Tsinghua University
Beijing 100084, China

Yi Pan
Dept. of Computer Sci.
Georgia State University
Atlanta, GA 30302-3994, USA

Jie Wu
Dept. of Computer Sci. and Eng.
Florida Atalantic University
Boca Raton, FL 33431, USA

Abstract — A new deadlock-free adaptive routing algorithm is proposed for $n-$dimensional meshes with only two virtual channels, where a virtual channel can be shared by two consecutive planes without any cyclic channel dependency. A message is routed along a series of planes. The proposed planar adaptive routing algorithm is enhanced to a fully adaptive routing version for 3-dimensional meshes using the idle virtual channels along the last dimension. Another deadlock avoidance technique is proposed for 3-dimensional meshes using a new virtual network partitioning scheme with only two virtual channels. Two virtual networks can share some common virtual channel based on the virtual network partitioning scheme. The deadlock-free adaptive routing scheme is then modified to a deadlock-free adaptive fault-tolerant routing scheme based on a planarly constructed MCC fault model. Sufficient simulation results are presented to demonstrate the effectiveness of the proposed algorithm.

Keywords —Deadlock-free adaptive routing, fault-tolerant routing, mesh, minimum-connected components, planar adaptive routing.

I. Introduction

Mesh-connected networks have been widely used in recent experimental or commercial multicomputers [1, 13]. A mesh network has an $n-$dimensional grid structure with k nodes in each dimension (called a mesh for short). An effective fault-tolerant routing algorithm in a mesh network is essential for a high-performance multicomputer system. Dally [4] presented the sufficient and necessary conditions for deadlock-free routing in meshes/tori.

Chien and Kim [3] proposed an important partially adaptive routing algorithm called the planar-adaptive routing algorithm. This algorithm constrains routing inside a sequence of planes. It prevents deadlocks with only three virtual channels. Judicious extension of the algorithm can efficiently handle fault-tolerant routing inside faulty n-dimensional meshes. Liu, *et al.* [11] proposed an improved planar adaptive routing algorithm for $n-$dimensional fault-free meshes with two virtual channels. However, Liu, *et al.* [11] cannot handle fault-tolerant adaptive routing in $n-$dimensional meshes.

It is not difficult to present a fully adaptive routing algorithm for fault-free $n-$dimensional meshes based on the dimension-order routing scheme and Duato's protocol [5], but it is not easy to handle fault-tolerant adaptive deadlock-free routing in $n-$dimensional meshes with only two virtual channels. Boppana and Chalasani [2] developed fault-tolerant routing algorithms for wormhole-routed meshes based on the $e-$cube routing algorithm and the block fault model. Four virtual channels are sufficient to present fully adaptive fault-tolerant routing in meshes. However, the methods [2, 3, 16] must disable some fault-free nodes to construct the fault blocks, which can result in a great loss of computational power for 3-dimensional or higher dimensional networks.

Pipelined-circuit-switching (PCS) [7] establishes a path by reserving a virtual channel path before sending a message. This provides very good reliability and simplifies the deadlock-free design. Wu [16] and Xiang [17] proposed adaptive and deadlock-free fault-tolerant routing algorithms with linear virtual channels corresponding to the number of dimensions in meshes based on limited-global-safety measures. The extended local safety in [17] proposed a new fault-tolerant routing scheme by setting up a path without reserving any system resource before sending a message. Their scheme used a planarly constructed fault block to improve the fault-tolerance capability of the method.

In [8], Gomez *et al.* presented a two phase routing scheme by selecting an intermediate node to avoid faulty nodes, which needs three virtual channels in order to implement fully-adaptive, deadlock-free, fault-tolerant routing in meshes. Recently, Puente, Gregorio, Vallejo, and Beivide [14] proposed a fault-tolerant routing mechanism for the virtual cut-through switched $k-$ary n-cubes based on the bubble flow control scheme, which can handle any number of faults if the network is connected. Recently, Wang [15] proposed a minimum-connected component (MCC) fault block model to do fault-tolerant routing in 2D meshes by disabling fewer fault-free nodes. In this model, each fault-free node is required to store several copies of safety information. The MCC fault model was extended to 3-dimensional meshes in [9].

The main contribution of this paper includes: (1) a new deadlock-free routing planar adaptive routing scheme with only two virtual channels, proposed for meshes by sharing a common virtual channel for two consecutive planes, and extended to faulty meshes; (2) a new planarly constructed MCC fault model presented to support fault-tolerant routing in wormhole-routed meshes; (3) a fully adaptive, deadlock-free, fault-tolerant routing scheme for 3-dimensional meshes that uses two virtual channels based on a channel overlap scheme.

The remainder of this paper is set up as follows. The planarly constructed MCC fault model is introduced in Section 2. The new deadlock-free routing schemes in 3-dimensional meshes are proposed in Section 3 by using only two virtual channels, which is extended to n-dimensional meshes in

*This work was partially supported by the National Science Foundation of China under grants 60425203 and 60573055.

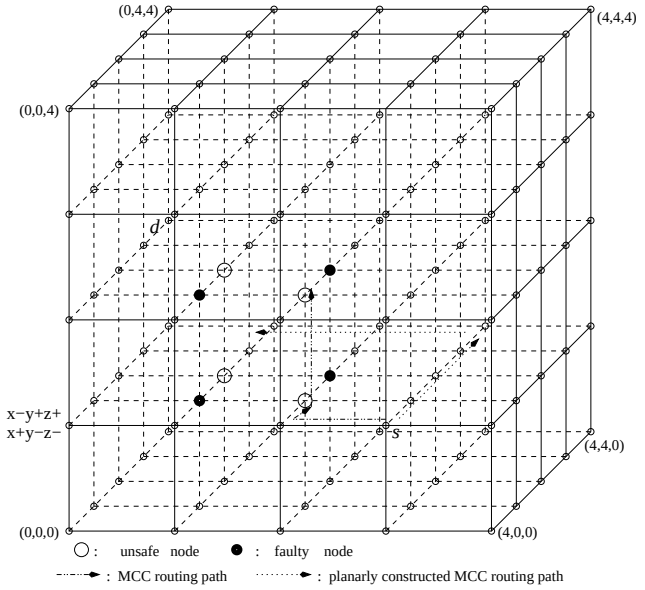

Figure 1: Planarly constructed MCC fault blocks.

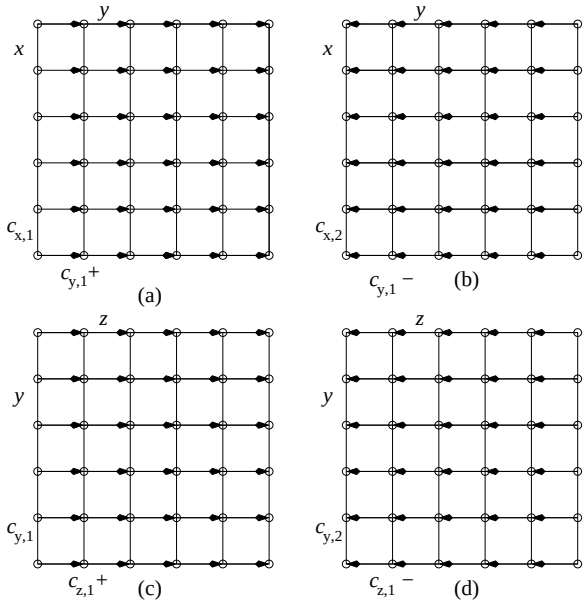

Figure 2: The proposed deadlock avoidance in 3-dimensional meshes: (a), (b) The increasing and decreasing networks in an $x-y$ plane, (c), (d) the increasing and decreasing networks in a $y-z$ plane.

Section 4. A new deadlock-free, fault-tolerant routing algorithm based on the planarly constructed MCC fault model is proposed in Section 5, which supports minimal and non-minimal routing in $n-$dimensional meshes. Extensive simulation results are presented in Section 6. The paper is concluded in Section 7.

II. The Planarly Constructed MCC Fault Model

The block fault model in [2, 3, 16] can label too many fault-free nodes as faulty or unsafe, which can greatly reduce the computational power of the system. The MCC fault model was proposed in [15] for 2D meshes, and it does not use global fault information. Each fault-free node in a 2D mesh keeps two copies of fault information, one for the $x+y+$(or $x-y-$) directions, and the other for $x-y+$(or $x+y-$) directions.

The MCC fault block model was extended to 3D meshes in [9]. The method in [9] labels only a few number of fault-free nodes as unsafe. Let us consider the labeling process in directions $x-y+z+$ (or $x+y-z-$). The technique can be stated as follows. Initially, all fault-free nodes are set as safe. A safe node is set to unsafe if it has three faulty or unsafe neighbors along $x-$, $y+$, and $z+$ (or $x+$, $y-$, and $z-$). Continue the above process until all nodes get stable states. In a 3-D mesh, each fault-free node keeps a 4-element tuple (a, b, c, d) to store the states of the node, where a, b, c, and d represent states of the node along directions $x+y+z+$ ($x-y-z-$), $x-y+z+$ (or $x+y-z-$), $x-y-z+$ (or $x+y+z-$), $x+y+z-$ (or $x-y-z+$). Here, a, b, c, and d can be faulty, unsafe, or safe.

As shown in Fig. 1, the 5x5x5 mesh contains four faulty nodes. Fault-free nodes (2,1,1), (2,1,2), (1,2,1), and (1,2,2) are set to safe for message routing along directions $x-y+z+$ (or $x+y-z-$) by the 3-dimensional MCC fault model [9]. Those nodes can lead the message to dead-ends because the MCC-fault-model-based routing scheme only supports minimum routing. A dead-end makes a message undeliverable in a minimum routing scheme although a minimum path is available. Therefore, the 3-dimensional MCC fault model for 3-dimensional meshes is not suitable for the proposed planar adaptive routing scheme in 3-dimensional meshes. As shown in Fig. 1, fault-free nodes (2,1,1) and (1,2,1) should be labeled unsafe in the plane (*,*,1) in order to avoid a dead-end. The fault-free nodes (2,1,2) and (1,2,2) should also be labeled unsafe in the plane (*,*,2).

Let the source s and destination d of a message be (3,0,1) and (0,3,2) as shown in Fig. 1, respectively. Let us consider the 3-dimensional MCC fault model [9] extended from the 2-dimensional MCC fault model and the routing algorithm in [15]. As mentioned earlier in this section, nodes (2,1,1), (2,1,2), and (2,2,1) are set to safe by the 3-dimensional MCC in [9]. The message can be routed to (2,0,1) first, and node (2,1,1) can be selected as the next hop, where (2,1,2) can be selected as the next hop after (2,1,1). Therefore, the message is led to a dead-end, which is undeliverable based on the minimum routing routing algorithm in [15, 9].

A new scheme to collect fault information for the planarly adaptive routing schemes is proposed, where fault information is obtained corresponding to each plane that contains the node. Each node calculates its safety information in three planes (xy,yz,zx) that contain the node, and keeps two copies of safety information in each plane. The nodes (2,1,1) and (1,2,1) are set to unsafe in the plane (*,*,1) and the nodes (2,1,2) and (1,2,2) are set to unsafe in the plane (*,*,2). Let us consider routing a message from (3,0,1) to (0,3,2) again based on the planarly constructed MCC fault model. The safety information along directions $x-y+z+$ (or $x+y-z-$) are used to guide fault-tolerant routing. The proposed fault-tolerant routing algorithm is still planarly adaptive like [3], where the xy plane is selected to route the message first. The two unsafe nodes (2,1,1) and (2,1,2) can be avoided when routing a message from (3,0,1) to (0,3,2) as shown in Fig. 1.

Deadlock-free-adaptive-routing-xy+()
Input: Coordinates of the current node (x_c, y_c, z_c) and the destination (x_d, y_d, z_d).
Output: Selected output channel.

1. $x_{off} = x_d - x_c$, $y_{off} = y_d - y_c$;
2. if $x_{off} > 0$ and $y_{off} > 0$,
   channel:= select($c_{x,1}+$, $c_{y,1}+$);
3. if $x_{off} < 0$ and $y_{off} > 0$,
   channel:= select($c_{x,1}-$, $c_{y,1}+$);
4. if $x_{off} = 0$ and $y_{off} > 0$, channel:= $c_{y,1}+$ if $z_{off} = 0$;
   else *Deadlock-free-adaptive-routing-yz*();
5. if $y_{off} = 0$ and $x_{off} > 0$, then channel: = $c_{x,1}+$;
6. if $y_{off} = 0$ and $x_{off} < 0$, channel: = $c_{x,1}-$;
7. if x_{off}, y_{off}, and z_{off} are equal to 0,
   channel: = internal.

Figure 3: Deadlock-free adaptive routing in the increasing network of an $x-y$ plane in 3-dimensional meshes.

III. Deadlock-Free Fully Adaptive Routing in 3-Dimensional Meshes

We would like to propose two different techniques to avoid deadlocks using two virtual channels for 3-dimensional meshes. Our method still partitions the mesh network into two planes $x-y$ and $y-z$. Virtual channel assignment is almost the same as that of the planar adaptive routing scheme [3]. All channels along dimension x in plane $x-y$ use virtual channel $c_{x,1}$ in the increasing network, and $c_{x,2}$ virtual channels in the decreasing network as shown in Fig. 2. A message is routed across a hop along dimension y via virtual channel $c_{y,1}+$ in the increasing network, and $c_{y,1}-$ in the decreasing network.

Fig. 3 presents the proposed algorithm to route a message in the increasing network of plane $x-y$. The algorithm takes priority in selecting a hop along the x dimension if the offset along dimension x is greater than 0, and the corresponding $c_{x,1}$ virtual channel is available. The algorithm selects a $c_{y,1}+$ virtual channel along dimension y if the offset of dimension y is greater than 0 and the $c_{y,1}$ channel in a minimum path to the destination along dimension x is not available. The algorithm is simply extended to deadlock-free routing in the decreasing network of an $x-y$ plane in 3D meshes by using virtual channel $c_{x,2}$ for hops along dimension x, and $c_{y,1}-$ for hops along dimension y.

Let the offset between the source and destination along dimension x be traversed first. The message turns to the $y-z$ plane. Fig. 4 presents the deadlock avoidance technique in a $y-z$ plane. All channels along dimension z use virtual channels $c_{z,1}+$ in the increasing network, and $c_{z,1}-$ virtual channels in the decreasing network, while the virtual channels $c_{y,1}$ and $c_{y,2}$ are used for hops along dimension y in two networks, respectively. A message is routed across a hop along dimension y via virtual channel $c_{y,1}$ in the increasing network, and $c_{y,2}$ in the decreasing network.

Lemma 1 *No cyclic channel dependency exists in an $x-y$ plane based on the procedures presented in Fig. 3.*

Proof: A message in an $x-y$ plane traverses only in either the increasing network or the decreasing network. Therefore, no cyclic interdependency exists between the increasing and decreasing networks. Just like the planar adaptive routing algorithm, only $c_{y,1}+$ virtual channels for any hops along dimension y are used in the increasing network. Therefore, no cyclic channel dependency exists in the increasing network. Only the $c_{y,1}-$ virtual channels are used for hops along dimension y. Thus, no cyclic channel dependency exists in the decreasing network of an $x-y$ plane. ∎

Deadlock-free-routing-yz()

1. $y_{off} = y_d - y_c$, $z_{off} = z_d - z_c$;
2. if $y_{off} \neq 0$ and $z_{off} > 0$, channel: = select($c_{y,1}$, $c_{z,1}+$);
3. if $y_{off} \neq 0$ and $z_{off} < 0$, channel:= select($c_{y,2}$, $c_{z,1}-$);
4. if $y_{off} = 0$ and $z_{off} \neq 0$, then channel: = $c_{z,1}$;
5. if $z_{off} = 0$ and $y_{off} \neq 0$, then channel: = $c_{y,1}$ in the increasing network;
   else channel: = $c_{y,2}$ in the decreasing network;
6. if $x_{off} = 0$, $y_{off} = 0$ and $z_{off} = 0$, then channel: = internal.

Figure 4: Deadlock-free routing in a $y-z$ plane in 3D meshes.

Fig. 4 presents the routing algorithm in a $y-z$ plane. In a $y-z$ plane, the role of the dimension z is similar to the dimension y in an $x-y$ plane while dimension y becomes the low dimension in the plane. The algorithm does not take precedence to select a channel between dimensions y and z. A profitable $c_{y,1}$ channel along dimension y or a $c_{z,1}+$ channel along dimension z in the increasing network can be selected if the offset along dimension y is not equal to 0, and the offset along dimension z is greater than 0. A profitable $c_{y,2}$ channel along dimension y or a $c_{z,1}-$ channel along dimension z in the decreasing network can be selected if the offset along dimension y is not equal to 0, and the offset along dimension z is less than 0. Let the offset along dimension z become 0, a profitable $c_{y,1}$ or $c_{y,2}$ channel along dimension y is selected in the increasing network or the decreasing network, respectively. The routing scheme in a $y-z$ plane is presented in Fig. 4.

Lemma 2 *No cyclic channel dependency exists in any $y-z$ plane based on the proposed routing scheme in Fig. 4.*

Proof: As shown in Fig. 4, any message routed in a $y-z$ plane traverses in either the increasing network or the decreasing network. A message is routed in the increasing network if the destination has a greater label along dimension y than the source, and in the decreasing network if the destination has a smaller label along dimension y than the destination. No cyclic channel dependency between the increasing network and the decreasing network occurs because no message enters the other network if it is classified into one network. Still, no cyclic dependency exists in the increasing network or the decreasing network because there only unidirectional virtual channels exist for hops along dimension z. ∎

Lemma 3 *No cyclic channel dependency exists between planes $x-y$ and $y-z$.*

Proof: Some channel dependencies may exist from channels in an $x-y$ plane to channels in a $y-z$ plane because

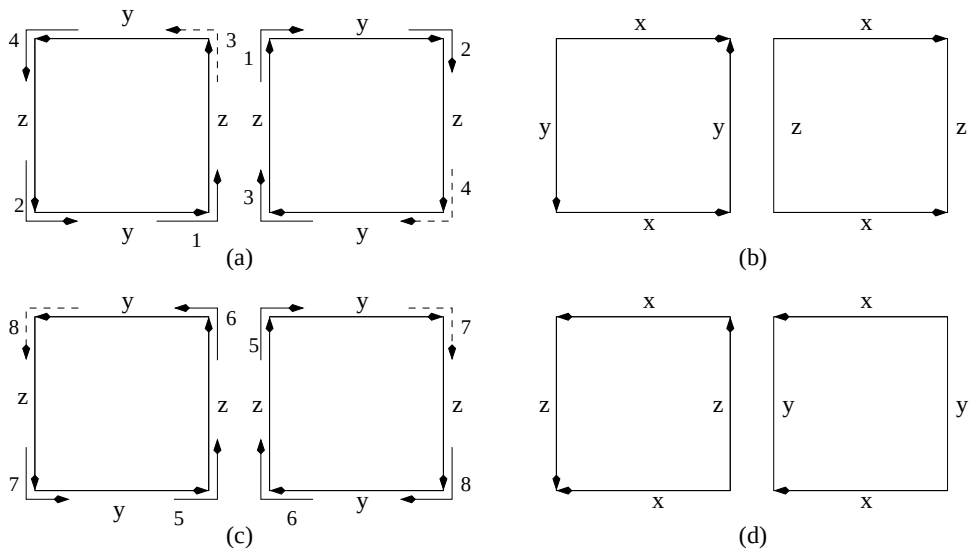

Figure 5: Deadlock-free fault-tolerant routing using channel overlap: (a) $x{+}y{+}z^*$ and $x{+}y{-}z^*$, (b) acyclic dependencies in $x{+}y{+}z^*$ and $x{+}y{-}z^*$, (c) $x{-}y^*z{+}$ and $x{-}y^*z{-}$, and (d) acyclic dependencies in $x{-}y^*z{+}$ and $x{-}y^*z{-}$.

virtual channel $c_{y,1}$ for channels along dimension y is shared by planes $x{-}y$ and $y{-}z$. There should be some channel dependencies from channels in a $y{-}z$ plane to channels in an $x{-}y$ plane in order to form cyclic channel dependencies. Regardless, no channel dependency exists from a channel in plane $y{-}z$ to a channel in an $x{-}y$ plane because a message in a $y{-}z$ plane never turns to a channel in an $x{-}y$ plane. ■

An idle virtual channel, $c_{z,2}$, exists for any channel along dimension z, which is not used by any message. A fully adaptive routing protocol can be proposed by using the idle virtual channels $c_{z,2}$ along dimension z, where the idle channel $c_{z,2}$ along dimension z is used as an adaptive channel. Note that the proposed fully adaptive, deadlock-free routing scheme is different from Duato's protocol, which does not provide extra adaptive channels for all physical channels along dimensions x and y.

As for n-dimensional mesh and torus networks, Linder and Harden [10] need $O(2^{n-1})$ and $(n+1)\cdot 2^{n-1}$ virtual channels, respectively. Wu [16] used 3 virtual channels to support minimal deadlock-free fault-tolerant routing in 3-dimensional meshes. We show that 2 virtual channels are sufficient in order to support minimal and non-minimal routing in faulty 3-dimensional meshes by using a channel overlapping scheme.

Let the source and destination be enabled in all planes that contain them in 3-dimensional meshes. The virtual network partitioning scheme combined with the fault-tolerant routing scheme presents deadlock-free routing. Eight virtual networks: $x{+}y{+}z{+}(1)$, $x{+}y{+}z{-}(2)$, $x{+}y{-}z{+}(3)$, $x{+}y{-}z{-}(4)$, $x{-}y{+}z{+}(5)$, $x{-}y{-}z{+}(6)$, $x{-}y{+}z{-}(7)$, and $x{-}y{-}z{-}(8)$, can be merged into 4 different virtual networks: $x{+}y{+}z^*$ ($c_{x,1}{+}$, $c_{y,1}{+}$, $c_{z,1}$), $x{+}y{-}z^*$ ($c_{x,2}{+}$,$c_{y,1}{-}$, $c_{z,1}$), $x{-}y^*z{+}$($c_{x,1}{-}$, $c_{y,2}$, $c_{z,2}{+}$), and $x{-}y^*z{-}$($c_{x,2}{-}$, $c_{y,2}$, $c_{z,2}{-}$), "*" indicates that the virtual network along the given dimension includes both "+" and "−" direction physical channels. In the first two merged virtual networks $x{+}y{+}z^*$ and $x{+}y-z^*$, all z channels share the same virtual channel $c_{z,1}$. In the last two virtual networks $x{-}y^*z{+}$ and $x{-}y^*z{-}$, all y channels share the same virtual channels $c_{y,2}$. Therefore, only two virtual channels are used.

As shown in Fig. 5, let all messages be classified corresponding to all eight original virtual networks. The labels on the arrowed lines represent the classes of messages. It is found that message classes 1, 2, 3, and 4 can form cyclic dependencies as shown in Fig. 5(a), and messages 5, 6, 7, and 8 establish cyclic dependencies as shown in Fig. 5(c). It is clear that only dimensions y and z can form cyclic dependencies, where no cyclic dependencies form in planes xy and xz. All cyclic channel dependencies can be removed by preventing the dashed, arrowed lines as presented in Figs. 5(a) and 5(c). The method used to prevent cyclic dependencies, as shown in Fig. 5(a) for message class 3, is to use $c_{y,2}{-}$ instead of $c_{y,1}{-}$ after the turn from a channel along dimension $z{+}$ to a channel along dimension $y{-}$. The turn from a channel along dimension $z{-}$ to a channel along dimension $y{-}$ for message class 4 also uses channel $c_{y,2}{-}$ instead of $c_{y,1}{-}$. As for message classes 1 and 2, no constraints are necessary.

Similarly, the cyclic channel dependencies for message classes 5, 6, 7, and 8 can also be prevented. As shown in Fig. 5(c), a turn from a $y{-}$ channel to a $z{-}$ channel for message class 8 uses $c_{z,1}{-}$ instead of $c_{z,2}{-}$. A turn from a $y{+}$ channel to a $z{-}$ channel for message class 7 uses virtual channel $c_{z,1}{-}$ instead of $c_{z,2}{-}$. As for message classes 5 and 6, no constraints are necessary. This scheme avoids channel dependencies among virtual networks. A message can be derouted along the dimension with an "*" label in a merged virtual network without any constraint, where the derouted message does not incur any additional channel dependency.

IV. Deadlock-Free Adaptive Routing in n−Dimensional Meshes

The planar adaptive deadlock-free routing protocol in Section 3 is extended to n-dimensional meshes. Let the mesh network contain dimensions $x_1, x_2, \ldots, x_{i-1}, x_i, x_{i+1}, \ldots, x_n$. The deadlock-free adaptive routing protocol is proposed in planes $x_{i-1}x_i$ and x_ix_{i+1} $(2 \le i \le n-1)$. The deadlock

Deadlock-free-adaptive-routing$(i, i+1)$

1. $x_i(off) = x_{d,i} - x_{c,i}$, $x_{i+1}(off) = x_{d,i+1} - x_{c,i+1}$;
2. let $x_i(off) \neq 0$, if $x_{i+1}(off) > 0$, channel:= select($c_{x_i,1}$, $c_{x_{i+1},1}+$);
   else if $x_{i+1}(off) < 0$, channel:= select($c_{x_i,2}$, $c_{x_{i+1},1}-$);
3. let $x_i(off) = 0$ and $x_{i+1}(off) \neq 0$, if $i+2 \leq n$, then *Deadlock-free-adaptive-routing*$(i+1, i+2)$; else if $i+1 = n$, channel := select($c_{x_{i+1},1}, c_{x_{i+1},2}$);
4. if $x_{i+1}(off) = 0$ and $x_i(off) \neq 0$, channel: $= c_{x_i,1}$ if the message is in the increasing network;
   else channel: $= c_{x_i,2}$ if the message is in the decreasing network;
5. if $x_1(off) = 0$, $x_2(off) = 0$, ..., $x_n(off) = 0$, then channel: = internal.

Figure 6: Deadlock-free adaptive routing in a plane (x_i, x_{i+1}).

avoidance technique is completely the same as that in the 3-dimensional mesh presented in Section 3. Let ($x_{c,1}$, $x_{c,2}$, ..., $x_{c,n}$) and ($x_{d,1}$, $x_{d,2}$, ..., $x_{d,n}$) be the coordinates of the current node and the destination, respectively.

The procedure as shown in Fig. 6 presents the deadlock-free adaptive routing procedure in the plane (x_i, x_{i+1}). Dimension x_i is used as the low dimension in a plane (x_i, x_{i+1}), where a common virtual channel $c_{x_i,1}$ along dimension x_i is shared by the planes (x_{i-1}, x_i) and (x_i, x_{i+1}). The virtual channels $c_{x_i,1}$ and $c_{x_i,2}$ are assigned to the channels of the increasing and decreasing networks along dimension x_i, and the virtual channels $c_{x_{i+1},1}+$ and $c_{x_{i+1},1}-$ are assigned to the channels along dimension x_{i+1} in the increasing network and decreasing network, respectively. The idle virtual channel $c_{x_n,2}$ for any channels along dimension x_n can still be used to improve the adaptivity of the deadlock-free routing protocol for n-dimensional meshes, which is also quite useful for the partially adaptive fault-tolerant routing protocol presented in Section 6.

A plane (x_{i-1}, x_i) is still partitioned into the increasing network and the decreasing network. Any message is routed inside one of the networks. Virtual channels $c_{x_{i-1},1}$ and $c_{x_{i-1},2}$ for channels along dimension x_{i-1} are used in the increasing network and the decreasing network, respectively. The virtual channels $c_{x_i,1}+$ and $c_{x_i,1}-$ for channels along dimension x_i are used for the increasing network and decreasing network, respectively.

Theorem 1 *Two virtual channels are sufficient to present deadlock-free adaptive routing in n-dimensional meshes.*

Proof: Just like the situations in the 3-dimensional meshes, a plane (x_{i-1}, x_i) is partitioned into an increasing network and a decreasing network, where any message is routed inside one of the networks. Therefore, no cyclic channel dependency can be formed in the plane. Similarly, no cyclic channel dependency can be established in the plane (x_i, x_{i+1}).

Any message routed in the increasing network of a plane (x_i, x_{i+1}) uses virtual channel $c_{x_i,1}$ for any channel along dimension x_i, which does not form any turn from a channel along dimension x_i to a channel along dimension x_{i-1}. Therefore, it does not establish any cyclic channel dependency with messages in plane (x_{i-1}, x_i) although a common virtual channel $c_{x_i,1}$ is used by both planes for channels along dimension x_i. No cyclic channel dependency can be formed between planes (x_{i-1}, x_i) and (x_i, x_{i+1}). Similarly, any two consecutive planes of (x_1, x_2), (x_2, x_3), ..., (x_{i-1}, x_i), (x_i, x_{i+1}), ..., (x_{n-1}, x_n) cannot form any cyclic channel dependency. ∎

route(c, d, x_i, x_{i+1})

1. channel:= select($c_{x_i,1}$, $c_{x_{i+1},1}$)) if $x_i(off) \neq 0$ and $x_{i+1}(off) \neq 0$, and both neighbors $c^{(i)}$ and $c^{(i+1)}$ be safe; if $c^{(i)}$ is unsafe or faulty, channel:= $c_{x_{i+1},1}$; if $c^{(i+1)}$ is unsafe or faulty, channel:= $c_{x_i,1}$ in the increasing network, channel:= $c_{x_i,2}$ in the decreasing network; if $c^{(i)}$ and $c^{(i+1)}$ are unsafe or faulty, channel:=$c_{x_n,2}$.
2. Let $x_{i+1}(off) = 0$ and $x_i(off) \neq 0$, if $c^{(i)}$ is safe, channel:= $c_{x_i,1}$ in the increasing network, channel:= $c_{x_i,2}$ in the decreasing network;
   else if $c^{(i)}$ is unsafe or faulty, channel:= $c_{x_n,2}$;
3. Let $x_i(off) = 0$, $x_{i+1}(off) \neq 0$ and at least one of $c^{(i+1)}$ and $c^{(i+2)}$ is safe in the plane (x_{i+1}, x_{i+2}), *route*(c, d, x_{i+1}, x_{i+2});
   else if $c^{(i+1)}$ and $c^{(i+2)}$ are unsafe in the plane (x_{i+1}, x_{i+2}) or faulty, deroute the message along dimension x_n;
4. If $x_1(off) = 0$, $x_2(off) = 0$, ..., $x_n(off) = 0$, channel:= internal.

Figure 7: Deadlock-free fault-tolerant routing in the plane (x_i, x_{i+1}) in n-dimensional meshes.

V. Fault-Tolerant Deadlock-free Adaptive Routing

A new fault-tolerant routing scheme in a wormhole-switched mesh network is proposed based on a planarly constructed MCC fault model. This fault model is quite similar to the planarly constructed fault block model in [17], however, more unsafe nodes inside planarly constructed fault models are activated. Let the mesh network contain dimensions $x_1, x_2, \ldots, x_{i-1}, x_i, x_{i+1}, \ldots, x_n$. We present the fault-tolerant routing procedures in the plane $x_i x_{i+1}$ $(1 \leq i \leq n-1)$. The deadlock avoidance technique is almost the same as the fault-free network presented in Section 3.

It is not necessary for each fault-free node to keep the safety information corresponding to all planes that contain it. Each node should keep the safety information based on the MCC fault model in the following planes, (x_1, x_2), (x_2, x_3), ..., (x_{n-1}, x_n). Fig. 7 presents the detailed deadlock-free planar adaptive fault-tolerant routing protocol inside a plane (x_i, x_{i+1}) in an n-dimensional mesh. The parameter $c^{(i)}$ represents the neighbor of node c along dimension x_i in a minimum path from c to the destination d. A node called safe or unsafe in Fig. 7 represents that it is locally safe inside the plane under consideration based on the planarly constructed MCC fault model. The procedure takes precedence when selecting a hop along the lower dimension in order to improve the adaptivity of the routing protocol when profitable neighbors along two dimensions are available.

Without loss of generality, let the source and destination differ in dimensions $x_1, x_2, \ldots, x_n$. Two consecutive dimen-

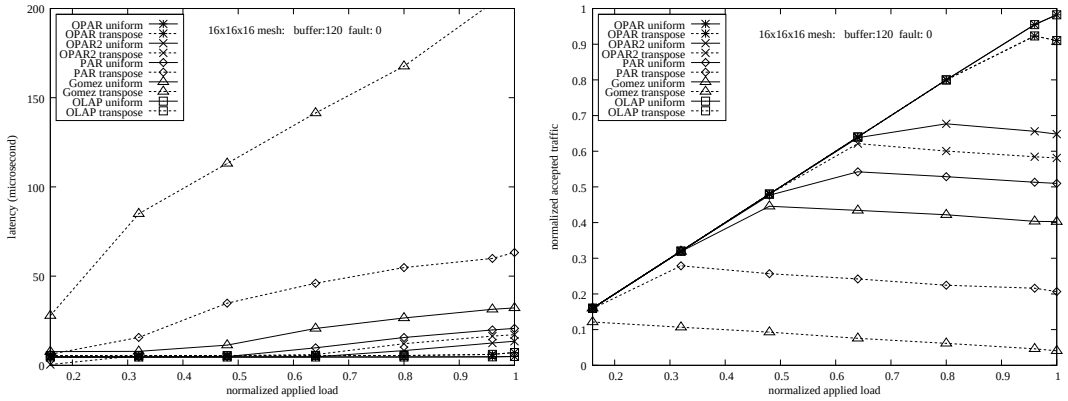

Figure 8: Performance comparison with previous methods in a fault-free network for different loads.

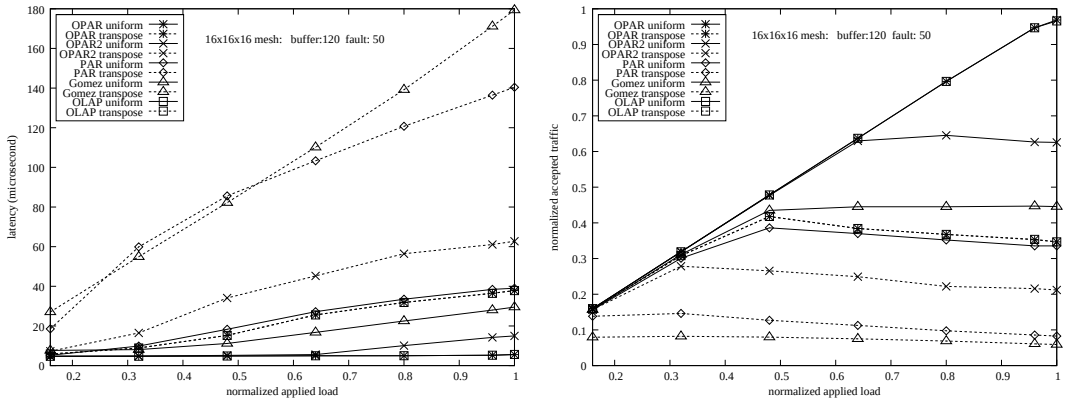

Figure 9: Fault-tolerant performance comparison with previous methods in 16×16×16 meshes.

sions establish a plane. A message is routed inside a series of planes like PAR [3]. In the plane (x_i, x_{i+1}), dimensions x_i and x_{i+1} are thought of as the low dimension and the high dimension, respectively. The plane is still partitioned into two virtual networks: (1) the increasing network and (2) the decreasing network. Any message is routed inside one of the networks considering the offset between c and d. A message is routed along dimension x_{i+1} via $c_{x_{i+1},1}+$ and $c_{x_{i+1},1}-$ in the increasing network and decreasing network, respectively. Virtual channels $c_{x_{i+1},1}$ and $c_{x_{i+1},2}$ are used in the increasing network and decreasing network when routing a message along dimension x_{i+1} in the plane (x_{i+1}, x_{i+2}). The message must be routed across a hop along the other dimension when the profitable neighbor along one dimension is faulty or unsafe in the plane based on the MCC fault model. The message turns to the new plane (x_{i+1}, x_{i+2}) by calling the procedure *route*(c, d, x_{i+1}, x_{i+2}) when the offset of dimension x_i has been reduced to 0.

The message can be derouted in plane $(x_i, x_{i+1})(i < n-1)$ in only one case; when the offset along dimension x_{i+1} becomes 0 and the offset along dimension x_i is still not 0. Derouting is necessary when no feasible path leading to the destination along dimension x_i exists. The planar adaptive routing scheme [3] is unable to deliver the message to the destination d in this case. Our method routes (or deroutes) the message via the idle virtual channel $c_{x_n,2}$ by a hop along the last dimension x_n. The virtual channels $c_{x_n,2}$ along dimension x_n according to the deadlock avoidance technique presented in Fig. 2 are always idle. The proposed routing protocol can take precedence to select a hop across the low dimension in a plane in order to enhance adaptivity when profitable hops along both dimensions are available. However, two dimensions have the same precedence in the last plane (x_{n-1}, x_n).

Consider a message with offsets along dimensions $x_1, x_3, x_5, \ldots, x_{2i-1}, \ldots$ without loss of generality. The message has to be routed along a single path based on the planarly-adaptive routing algorithm [3], where the message cannot be delivered to the destination when one of the nodes in the path is faulty. The following planes can be established, (x_1, x_2), (x_3, x_4), …, (x_{2i-1}, x_{2i}), …, where planes (x_2, x_3), (x_4, x_5), …, (x_{2i}, x_{2i+1}), … can also be included if necessary. The message can be routed inside any of the planes sequentially, and some deroutes may be necessary inside any of the planes mentioned above when a faulty node exists in the single path mentioned. This technique can improve the adaptivity and reachability of the proposed routing protocol.

Just like the deadlock-free, fully adaptive routing scheme presented, the idle virtual channel $c_{x_n,2}$ along dimension x_n can still be used to improve the adaptivity of the deadlock-free adaptive routing algorithm in an n-dimensional mesh. The proposed deadlock-free, fault-tolerant routing algorithm is enhanced to a fully adaptive protocol for 3-dimensional meshes by making full use of the idle virtual channel $c_{x_3,2}$ along dimension x_3.

VI. Simulation Results

Flit-level simulators have been implemented to evaluate the proposed wormhole-routing-based deadlock-free routing

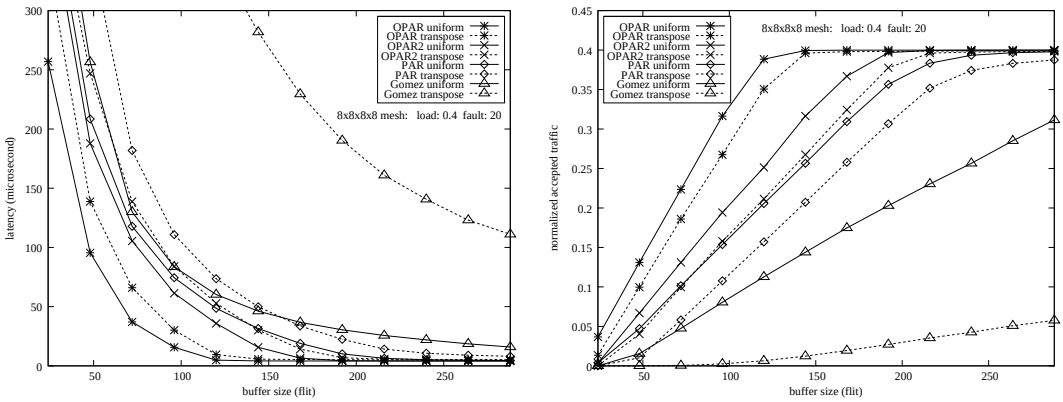

Figure 10: Impact of buffer size on performance in 8×8×8×8 meshes.

algorithms called OPAR (the new planar adaptive routing algorithm) and OLAP (channel overlap). A flit-level simulator on the original planar adaptive routing algorithm called PAR, proposed by Chien and Kim [3], is also implemented. Two different traffic patterns, uniform and transpose, are considered in all simulation results. As for the transpose traffic pattern in a $k \times k \times \ldots \times k$ mesh, the destination should be (k-i_1-1, k-i_2-1, …, k-i_n-1) if the source is (i_1, i_2, …, i_n). In all simulation results, we set both the start-up latency and receipt latency to 0.75 microseconds. The data transmission speed is 320 Mbytes/second between any two adjacent routers. Faults are randomly inserted into the network for fault-tolerant network evaluation. The message length is set to 16 flits in all cases.

Two important metrics, latency (time required to deliver a message) and the normalized accepted traffic (throughput divided by the saturation load, for example, the saturation loads for 8×8×8 and 16×16×16 meshes are 0.25 and 0.125 flit/node/cycle), are evaluated. Our methods need only two virtual channels in order to avoid deadlocks for each physical channel in $n-$dimensional meshes(OPAR2), and the planar adaptive routing scheme PAR [3] requires three virtual channels to avoid deadlocks. The proposed routing scheme is a fully adaptive routing scheme that uses the idle virtual channel of each channel along the last dimension. An extra virtual channel is used as an adaptive channel based on Duato's protocol [5] (OPAR and OLAP) in order to present fair comparison when comparing with PAR [3] and Gomez [8]. The OLAP method gets quite close results to OPAR in almost all cases.

Figure 8 presents a performance comparison among PAR [3], OPAR, and Gomez [8] when the buffer size of each node is set to 120 flits in the fault-free 16×16×16 mesh. It is found that the latency to deliver a message for the transpose communication pattern is greater than that of the uniform communication pattern for all three methods. OPAR needs less latency to deliver a message in all cases for both communication patterns. OPAR obtains the best *normalized accepted traffic* for both communication patterns for all load rate situations. The Gomez method gets the worst throughput for both communication patterns in the fault-free network. It is shown that the normalized accepted traffic for PAR and Gomez is insensitive to the normalized applied load for the transpose communication pattern. OPAR consistently gets better normalized accepted traffic until the normalized applied load increases to anything above 0.6.

Performance comparisons for the three methods in the faulty 16×16×16 mesh are presented in Figure 9 when the the buffer size for each node and the number of faults in the network are set to 120 flits and 50, and the *normalized applied load* changes from 0.16 to 1.0. OPAR requires the least latency in order to deliver a message for both communication patterns. OPAR needs even less latency than Gomez and PAR for the transpose communication pattern when the load rate increases. The Gomez method gets the worst normalized accepted traffic for the transpose communication pattern, while the PAR algorithm performs the worst for the uniform communication pattern. The throughput of the PAR algorithm decreases earlier than the other two methods.

Figure 10 presents the impact of buffer size in the faulty 8×8×8×8 mesh when the normailized applied load is set to 0.4., which contains 20 faulty nodes. It is shown that the Gomez method requires the most latency to deliver a message in all cases for both communication patterns. Especially, the Gomez method under the transpose communication pattern needs much more latency in all cases. The OPAR algorithm needs the least latency to deliver a message in all cases. As for the normailized accepted traffic, Gomez's method gets worse results in all cases for both communication patterns when compared to the other two methods. The OPAR algorithm gets better throughput for all buffer sizes.

Figure 11 presents fault-tolerant performance of the three algorithms when the number of faults increases from 0 to 600 in 8×8×8×8 meshes. Simulation results show that the PAR algorithm is unable to deliver any message when the number of faults is close to 60 for the transpose communication pattern and 80 for the uniform communication pattern. The latency for the PAR algorithm increases sharply at those points because of the fault model. The Gomez algorithm needs much more time to deliver a message for the transpose communication pattern. The PAR algorithm gets much better normalized accepted traffic for the transpose communication pattern when the number of faulty nodes is less than 75. The OPAR algorithm gets better normalized accepted traffic for both communication patterns in all cases.

VII. Conclusions

The number of virtual channels per physical channel required for deadlock-free routing is important when designing a cost-effective and high-performance system. In this

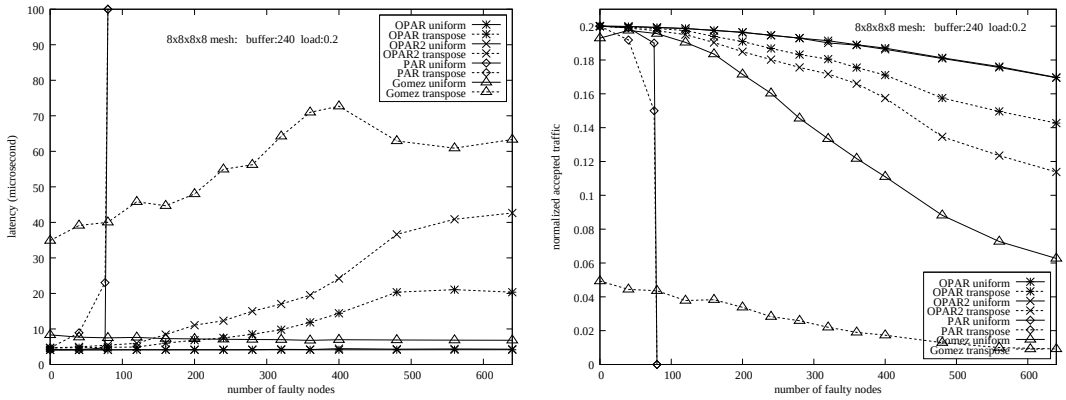

Figure 11: Fault-tolerant performance evaluation in 8×8×8×8 meshes.

paper, a new deadlock-free routing scheme is proposed for n-dimensional meshes, where any physical channel needs two virtual channels to avoid deadlocks. Two consecutive planes share the same virtual channel for channels along the same dimension. The proposed deadlock-free adaptive routing scheme is enhanced to a fully adaptive one for 3-dimensional meshes by using the idle virtual channel along the last dimension. The deadlock-free routing scheme is extended to fault-tolerant deadlock-free adaptive routing in n-dimensional meshes. Extensive simulation results show a very apparent advantage of the proposed routing scheme in fault-free and faulty meshes over the original planar adaptive routing scheme and other methods.

References

[1] F. Allen, *et al.*, "Blue gene: A vision for protein science using a petaflop supercomputer," *IBM Systems Journal*, vol. 40, pp. 310-327, 2001.

[2] R. V. Boppana and S. Chalasani, "Fault-tolerant wormhole routing algorithms for mesh networks," *IEEE Trans. Computers*, vol. 44, no. 7, pp. 848-864, 1995.

[3] A. A. Chien and J. H. Kim, "Planar adaptive routing: Low-cost adaptive networks for multiprocessors," *Journal of ACM*, vol. 42, no. 1, pp. 91-123, 1995.

[4] W. J. Dally and G. L. Seitz, "Deadlock-free message routing in multiprocessor interconnection networks," *IEEE Trans. on Computers*, vol. 36, no. 5, pp. 547-553, 1987.

[5] J. Duato, S. Yalamanchili, and L. Ni, *Interconnection Networks: An Engineering Approach*, IEEE Press, 1997.

[6] J. Duato, "A new theory of deadlock-free adaptive routing in wormhole networks," *IEEE Trans. Parallel and Distributed Systems*, vol. 4, no. 12, pp. 1320-1331, 1993.

[7] P. T. Gaughan, B. V. Dao, S. Yalamanchili, and D. E. Schimmel, "Distributed, deadlock-free routing in faulty, pipelined, direct interconnection networks," *IEEE Trans. Computers*, vol. 45, no. 6, pp. 651-665, 1996.

[8] M. E. Gomez, N. A. Nordbotten, J. Flich, P. Lopez, A. Robles, J. Duato, T. Skeie, and O. Lysne, "A routing methodology for achieving fault tolerance in direct networks," *IEEE Trans. on Computers*, vol. 55, no. 4, pp. 400-415, 2006.

[9] Z. Jiang, J. Wu, and D. Wang, "A new fault information model for fault-tolerant adaptive and minimal routing in 3-D meshes," *Proc. of 34th Int. Conference on Parallel Processing*, pp. 500-507, 2005.

[10] D. H. Linder and J. C. Harden, "An adaptive and fault-tolerant wormhole routing strategy for $k-$ary $n-$cube," *IEEE Trans. Computers*, Vol. 40, no. 1, pp. 2-12, 1991.

[11] X. Liu, S. Zhang, and T. J. Li, "A cost-effective load balanced adaptive routing scheme for mesh-connected networks," *Proc. of 8th Int. Symp. on Modeling, Analysis and Simulation of Computer and Telecommunication Systems*, pp. 532-538, 2000.

[12] A. Mejia, J. Flich, J. Duato, S. Reinemo, and T. Skeie, "Segment-based routing: An efficient routing algorithm for meshes and tori," *Proc. of IEEE Int. Symp. on Parallel and Distributed Processing*, 2006.

[13] S. S. Mukerhjee, R. Bannon, S. Lang, and A. Spink, "The Alpha 21364 network architecture," *IEEE Micro*, vol. 22, pp. 26-35, 2002.

[14] V. Puente, J. A. Gregorio, F. Vallejo, and R. Beivide, "Immunet: A cheap and robust fault-tolerant packet routing mechanism," *Proc. of ACM/IEEE Int. Symp. on Computer Architecture*, pp. 198-209, 2004.

[15] D. Wang, "A rectilinear-monotone polygonal fault block model for fault-tolerant minimal routing in mesh," *IEEE Trans. Computers*, vol. 52, no. 3, pp. 310-320, 2003.

[16] J. Wu, "A fault-tolerant adaptive and minimal routing approach in n-dimensional meshes," *Proc. of IEEE Int. Conf. Parallel Processing*, pp. 431-438, Aug. 2000.

[17] D. Xiang, J. G. Sun, J. Wu, and K. Thulasiraman, "Fault-tolerant routing in meshes/tori using planarly constructed fault blocks," *Proc. of 34th Int. Conference on Parallel Processing*, pp. 577-584, 2005.