A Simple Fault-Tolerant Adaptive and Minimal Routing Approach in 3-D Meshes *†

Jie Wu

Department of Computer Science and Engineering

Florida Atlantic University

Boca Raton, FL 33431

Abstract

In this paper we propose a sufficient condition for minimal routing in 3-dimensional (3-D) meshes with faulty nodes. It is based on an early work of the author on minimal routing in 2-dimensional (2-D) meshes. Unlike many traditional models that assume all the nodes know global fault distribution or just adjacent fault information, our approach is based on the concept of *limited global fault information*. First, we propose a fault model called *faulty cube* in which all faulty nodes in the system are contained in a set of faulty cubes. Fault information is then distributed to limited number of nodes while it is still sufficient to support minimal routing. The limited fault information collected at each node is represented by a vector called *extended safety level*. The extended safety level associated with a node can be used to determine the existence of a minimal path from this node to a given destination. Specifically, we study the existence of minimal paths at a given source node, limited distribution of fault information, minimal routing, and deadlock-free and livelock-free routing. Our results show that any minimal routing that is partially adaptive can be applied in our model as long as the destination node meets a certain condition. We also propose a dynamic planar adpative routing scheme that offers better fault tolerance and adaptivity than the planar adaptive routing scheme in 3-D meshes. Our approach is the first attempt to address adaptive and minimal routing in 3-D meshes with faulty nodes using limited fault information.

Keywords: *3-D meshes, adaptive routing, deadlock, fault tolerance, livelock, minimal routing*

---

*This work was supported in part by NSF grant CCR 9900646 and ANI 0073736.

†A preliminary version of this paper was presented at the International Conference on Parallel and Distributed Systems (ICPADS), 2000.

1 Introduction

In a multicomputer system, a collection of processors (or nodes) work together to solve large application problems. These nodes communicate data and coordinate their efforts by sending and receiving messages through the underlying communication network. Thus, the performance of such a multicomputer system depends on the end-to-end cost of communication mechanisms. Routing time of messages is one of the key factors that are critical to the performance of multicomputers. Basically, routing is the process of transmitting data from one node called the *source* node to another node called the *destination* node in a given system.

The *mesh-connected topology* [6, 9] is one of the most thoroughly investigated network topologies for multicomputer systems. Mesh-connected topologies, also called k-ary n-dimensional meshes, have an n-dimensional grid structure with k nodes in each dimension such that every node is connected to two other nodes in each dimension by a direct communication. Mesh-connected topologies include n-dimensional meshes, tori, and hypercubes.

The safety-level-based (or safety-vector-based) routing [14, 16], a special form of *limited-global-information-based* routing, is a compromise between local-information- and global-information-based approaches. In this type of routing, a routing function is defined based on current node, destination node, and limited global fault information gathered at the current node. Our approach differs from many existing ones where information is brought by the *header* of the routing message [3] and the routing function is defined based on header information and local state of the current node [8]. In our approach, neighborhood fault information is captured by an integer (safety level) or a binary vector (safety vector) associated with each node. For example, in a binary hypercube, if a node's safety level is l (an integer), then there is at least one Hamming distance (or minimal) path from this node to any node within l-Hamming-distance [16]. Using the safety level (or safety vector) associated with each node, a routing algorithm normally can obtain an optimal or suboptimal solution and requires a relatively simple process to collect and maintain fault information in the neighborhood. Therefore, limited-global-information-based routing can be more cost effective than routing based on global or local information. The safety-level-based routing has been successfully applied to high-dimensional mesh-connected topologies, such as binary hypercubes, but less efficient when it is directly applied to low-dimensional mesh-connected topologies, such as 2-D and 3-D meshes.

In this paper, we extend the safety level concept for 3-D meshes. First, we propose a novel fault model called *faulty cube* in which all faulty nodes in the system are contained in a set of faulty cubes. Fault information is then distributed to limited number of nodes while it is still sufficient to support minimal routing. The amount of limited-global-information should be kept minimum and it should be easy to obtain and maintain. In this paper, the limited fault information collected at each node is represented by a vector called *extended safety level.* The extended safety level associated with a node can be used to determine the existence of a minimal path from this node to a given destination. Specifically, we address the issues of the existence of a minimal path at a given source node, limited distribution of fault information, minimal routing, and deadlock-free and livelock-free routing. We also show the effect of fault information on the result of routing.

Our approach is the first attempt to address the adaptive and minimal routing in 3-D meshes with faulty nodes using limited fault information. Our results are extended from the research on 2-D meshes [15]. In 2-D meshes, the fault type is captured by a faulty block. In [15], a sufficient condition for the existence of a minimum path in 2-D meshes was determined. We also defined an extended safety level for 2-D meshes. The region of minimal paths in 2-D meshes was also identified as well as a minimal routing algorithm in 2-D meshes. This paper shows that these results can be extended to 3-D meshes.

Our main results include the following:

- A novel fault model called *faulty cube* is introduced in 3-D meshes. A simple labeling scheme is introduced to classify nodes into faulty, enable, and disable. A faulty cube consists of adjacent faulty and disable nodes.

- A new limited global information model called *extended safety level* associated with each node is proposed. The safety level information can be used to determine the existence of a minimal path for a given pair of source and destination nodes in 3-D meshes.

- We formally define the concepts of fully and partially adaptive routing. We also show that the *planar-adaptive routing* [5] fails to meet the proposed partially adaptive requirement.

- We propose a dynamic planar adaptive routing scheme as an extension of Chien and Kim's planar adaptive routing scheme. We show that within the context of minimal routing in 3-D meshes, dynamic planar adaptive routing offers better fault tolerance and adaptivity without using extra virtual channels compared with planar adaptive routing.

- We prove that any minimal routing that is partially adaptive can be applied in our model as long as the destination node meets a certain safety requirement.

- We propose a simple deadlock-free and livelock-free routing based on the use of virtual network. This approach can be implemented using only three virtual channels.

This paper is organized as follows: Section 2 presents some preliminaries. Section 3 proposes a new fault model called faulty cube in 3-D meshes. Section 4 introduces the concept of extended safety level as a special form of limited global information. Section 5 offers a simple adaptive and minimal

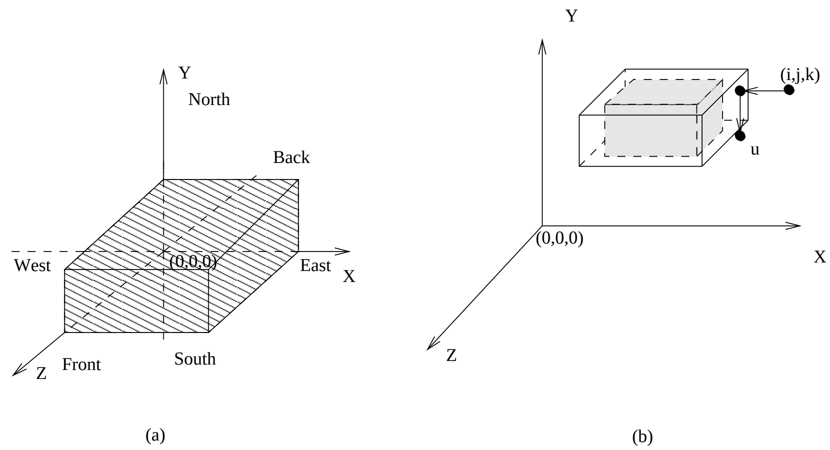

Figure 1: (a) Directions of a 3-d space. (b) Six surfaces of a faulty cube.

routing algorithm based on the limited global information provided in 3-D meshes. We also show that any partially adaptive and minimal routing algorithm can be applied in our model as long as the destination meets a certain safety requirement. Section 6 discusses possible extensions, including a strengthened sufficient condition, the application of the proposed approach in a 3-D torus, and deadlock-free and livelock-free routing. Section 7 concludes the paper.

2 Notation and Preliminaries

A k-ary n-dimensional mesh with $N{=}k^n$ nodes has an interior node degree of $2n$ and a network diameter of $n(k-1)$. Each node u has an address $(u_1, u_2, ..., u_n)$, where $u_i = 0, 1, ..., k-1$. Two nodes $(v_1, v_2, \cdots, v_n)$ and $(u_1, u_2, ..., u_n)$ are connected if their addresses differ in one and only one element (dimension), say dimension i; moreover, $|u_i - v_i| = 1$. Basically, nodes along each dimension are connected as a linear array. Normally, we do not specify the size of a mesh, i.e., the value k. A k-ary n-dimensional is simply denoted as an n-mesh. Each u_i in u: $(u_1, u_2, ..., u_n)$ is an integer.

A 3-D mesh can be drawn in a three-dimensional (3-D) space. Each node in a 3-D mesh is labeled as (i, j, k). We define the six directions in a 3-D mesh as follows: The direction along positive X is *East*, along negative X is *West*. The direction along positive Y is *North*, along negative Y is *South*. The direction along positive Z is *Front*, along negative Z is *Back* (see Figure 1 (a)). A 3-D space can be partitioned into eight regions based on positive and negative directions along three dimensions. To simplify our discussion, we assume that source is $(0, 0, 0)$ and destination is (i, j, k), with $i \geq 0, j \geq 0$ and $k \geq 0$ (see the shadow region in Figure 1 (a)). The other seven regions can be defined in a similar way.

Routing is a process of sending a message from a source to a destination. A routing is *minimal* if the length of the routing path from source $(0,0,0)$ to destination (i,j,k) is the distance between these two nodes, i.e., $|i|+|j|+|k|$. Throughout this paper, we focus only on minimal routing in a 3-D mesh with faulty components. The challenge is to find a minimal path (if there exists one) by avoiding faults in the system.

The simplest routing algorithms are *deterministic* which define a single path between the source and destination nodes. The X-Y-Z routing is an example of deterministic routing in which the message is first forwarded along the X dimension, then along the Y dimension, and finally along the Z dimension. *Adaptive* routing algorithms, on the other hand, support multiple paths between the source and destination nodes. *Fully adaptive and minimal* routing algorithms allow all messages to use any minimal paths.

3 Faulty cubes

Before proposing a minimal routing algorithm in a 3-D mesh with faulty components, we first discuss the fault model under consideration. We consider node faults only (link faults can be treated as node faults by disabling the corresponding adjacent nodes). Moreover, we have the following classification of nodes in a 3-D mesh and propose a fault model based on this classification.

Definition 1: *In a 3-D mesh, a healthy node is marked disable if there are two or more disable or faulty neighbors along different dimensions; otherwise, it is marked enable. A faulty cube contains all the connected disable and faulty nodes.*

Based on Definition 1, there are three types of nodes: faulty nodes, enable nodes, and disable nodes. To determine the status of a node in a 3-D mesh using Definition 1, all the healthy nodes are initially marked enable. In this way, there is no need to perform any calculation when a given 3-D mesh is fault-free; that is, all healthy nodes are marked enable by default. We assume that both source and destination nodes in a routing process are healthy and enable.

The concept of faulty cube in 3-D meshes stems from the faulty block model ([1], [2], [5], [10], [13]) in 2-D meshes. In 3-D meshes, connected faulty components may span three dimensions, rather than two dimensions in 2-D meshes. To study the property of a faulty cube, we first give the following definition: A *cube* is a solid that has six surfaces and any two cross sections perpendicular to the same surface generate two rectangles of the same size and shape.

Theorem 1: *In a 3-D mesh, a faulty cube defined by Definition 1 has the following properties: (1) Each faulty cube is a cube. (2) Each of the six surfaces of the faulty cube is perpendicular to an axis in 3-D meshes. (3) The distance between any two faulty cubes is at least two.*

Proof: We construct two 2-D planes that are perpendicular to an axis and both intersect with a given faulty cube. Without loss of generality, we assume that these two planes are adjacent, $x = x_1$ and $x = x_1 + 1$, and are perpendicular to the X axis. If we can show that the two regions (called faulty regions) generated from the intersections of the faulty cube and these two planes are two identical rectangles, we prove that the faulty cube is a cube (since dimension x can be replaced by y and z to show cases for other cross sections). The shapes of the regions in both planes are rectangles based on the definition of faulty block in 2-D meshes [15]. Suppose the two rectangles are not identical. Without loss of generality, we assume that there exists a node (x_1, y, z) that is just outside the faulty region in plane $x = x_1$ (i.e., it has one neighbor on plane $x = x_1$ that is inside the faulty region); however, node $(x_1 + 1, y, z)$ is inside the faulty region in plane $x = x_1 + 1$. Since node (x_1, y, z) is adjacent to two disable nodes, it should be marked disable based on Definition 1. This contradicts to the condition that it is outside the faulty region. Therefore, any cross section perpendicular to the X axis generates two rectangles of the same size and shape. The same procedure can be applied to the other two cross sections. Therefore, the faulty cube is a cube. In addition, each of the six surfaces of the faulty cube is perpendicular to an axis in 3-D meshes.

In order to show that the distance between any two faulty cubes is at least two, we show that the distance is at least two and two-distance cases exist. If two faulty cubes are less than two-distance, they are adjacent and must be grouped into one faulty cube based on the faulty cube definition. When a 3-D mesh contains two faulty nodes that are two-distance apart along one dimension and they have the same coordinates along the other two dimensions, these two nodes form two faulty cubes. For example, faulty nodes (x, y, z) and $(x + 2, y, z)$ are such examples. The common adjacent node $(x + 1, y, z)$ is marked enable. □

With the second property in Theorem 1, the address of a faulty cube can be simply described by the range along each dimension, e.g., $x_1 : x_2$ specifies the range along the x axis. A general faulty cube can be represented by $[x_1 : x_2, y_1 : y_2, z_1 : z_2]$ which covers $(x_2 - x_1 + 1) \times (y_2 - y_1 + 1) \times (z_2 - z_1 + 1)$ nodes. When the range along an axis is one, say $x_1 : x_1$, the corresponding cube is reduced to a block. When there are unit ranges along two dimensions, the corresponding cube becomes a line, and for three unit ranges, a point (a single faulty node).

Note that there are several different definitions of a faulty region in meshes. In [4], a healthy node in a 3-D mesh is marked disable if there are two or more disable or faulty neighbors; otherwise it is marked enable. The difference is that in [4], a healthy node is marked disable if there are two disable or faulty neighbors even they are on the same dimension. For example, suppose there are three faulty nodes in a 3-D mesh: (3,4,2), (3,5,1), (3,5,2) and (5,4,2) (Figure 2). Based on the faulty cube definition from [4], it will generate a faulty cube [3:5, 4:5, 1:2] and this cube contains $3 \times 2 \times 2 = 12$ nodes (see Figure 2 (b)). Based on Definition 1, these four faulty nodes generate two separate cubes: [3:3, 4:5,

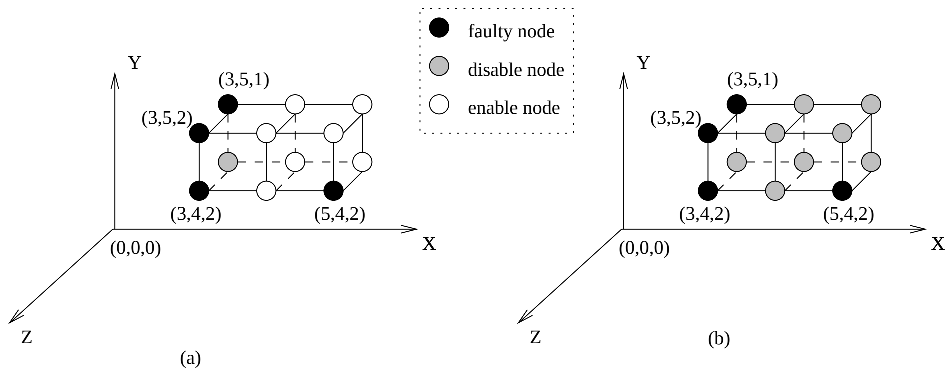

Figure 2: Two different formations of a faulty cube: (a) Based on Definition 1. (b) Based on the one in [4].

1:2], a block, and [5:5, 4:4, 2:2], a single node (see Figure 2 (a)). Clearly, the faulty cube model here includes fewer disable nodes.

Figure 3 (a) shows the average number of (synchronous) rounds needed to form a set of disjoint fault regions in 2-D meshes (which is 100×100) and 3-D meshes (which is $21 \times 21 \times 21$). Figure 3 (b) shows the average number of nonfaulty nodes marked as disable in 2-D meshes and 3-D meshes. Results show that the average number of rounds needed to form fault regions is between 1 and 4 if the number of faulty nodes stays within 100. The number of nonfaulty nodes marked as disable is small, compared to the number of faulty nodes in the system (this is especially true in 2-D meshes).

4 Extended Safety Level

In this section, we extend the safety level concept to 3-D meshes. The *safety level* [16] concept was originally proposed to capture limited global fault information in a binary hypercube. It was extended to 2-D meshes as *extended safety level* [15] which includes four elements, each of which indicates the distance to the closest faulty block to East, South, West, and North of the current node. The limited global information (captured by extended safety level) at each node can be used to decide the feasibility of a minimal routing. The following shows an important theorem that leads to our extended safety level definition in 3-D meshes and it serves as a basis of our approach.

Theorem 2: *Assume that node* $(0,0,0)$ *is the source and node* (i,j,k) *is the destination. If there is no faulty cube that goes across the X, Y and Z axes, then there exists at least one minimal path between* $(0,0,0)$ *and* (i,j,k)*. This result holds for any location of the destination and any number and distribution of faulty cubes.*

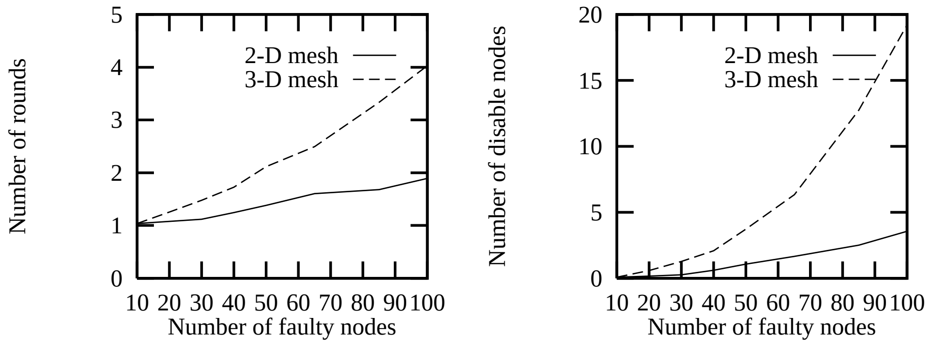

Figure 3: (a) Average number of rounds needed to construct fault regions. (b) Average number of nonfaulty nodes marked disable.

Proof: Without loss of generality, we assume that i, j, and k are non-negative integers. We prove this theorem by induction on m, the number of faulty cubes in region $R : [0 : i, 0 : j, 0 : k]$. Clearly, if region R does not contain any faulty cube, routing would be a regular one without faults. If $m = 1$, starting from node (i, j, k) and going straight along negative X until reaching plane $x = 0$. If it is able to reach plane $x = 0$ without hitting the faulty cube, the problem is reduced to routing in a 2-D mesh. The result in [15] shows that a minimal routing path exists in $x = 0$ as long as both Y and Z axes are clear of faulty blocks (cross sections of faulty cubes in plane $x = 0$). On the other hand, if it hits the faulty cube before reaching plane $x = 0$, then it must reach one of the six adjacent surfaces (of the faulty cube) that is perpendicular to the X axis (see Figure 1 (b)). In this case, the routing message turns South and goes along the negative Y direction. There are two cases: (1) If the faulty cube intersects with plane $y = 0$, then the message will eventually reach plane $y = 0$ and the problem is then reduced to routing in a 2-D mesh. (2) If the faulty cube does not intersect with plane $z = 0$, the message will eventually reach a node u that is on the edge of two adjacent surfaces of the faulty cube (see Figure 1 (b)). At node u, the faulty cube becomes irrelevant, since it will not block any minimal path between u and $(0, 0, 0)$. The remaining routing from u to $(0, 0, 0)$ is like a regular one without faulty cubes.

Assume that the theorem holds for $m = k - 1$. When $m = k$, we use the same approach for the $m = 1$ case. When the message reaches plane $x = 0$, the problem is reduced to routing in a 2-D mesh. Therefore, we only need to consider the case when the routing message hits a faulty cube. The routing message turns South when it hits a faulty cube. If the cube intersects with plane $y = 0$, the message will eventually hit plane $y = 0$ and the problem is then reduced to routing in a 2-D mesh; otherwise, the routing message will eventually reach a node u that is on the edge of two adjacent surfaces of the faulty cube. At node u, the faulty cube becomes irrelevant and remaining routing becomes a routing in a 3-D mesh with $m = k-1$ faulty cubes. Based on the induction assumption, the remaining routing should be able to find a minimal path from u to $(0,0,0)$. Combining two minimal paths, the resultant path is a minimal one between (i,j,k) and $(0,0,0)$. $\square$

The above result can be strengthened by including the location of destination (i,j,k).

Corollary: *Assume that node $(0,0,0)$ is the source and node (i,j,k) is the destination. If there is no faulty cube that goes across the sections of $0 \le x \le i$, $0 \le y \le j$, and $0 \le z \le k$ along the X, Y and Z axes, respectively, then there exists at least one minimal path between $(0,0,0)$ and (i,j,k).*

Note that the result of Theorem 2 can be weakened by strengthen the constraint in the theorem: If there is no faulty cube that goes across planes $x = 0$, $y = 0$, and $z = 0$. Theorem 2 is then renamed as Theorem 2a as follows:

Theorem 2a: *Assume that node $(0,0,0)$ is the source and node (i,j,k) is the destination. If there is no faulty cube that goes across planes $x = 0$, $y = 0$, and $z = 0$, then there exists at least one minimal path between $(0,0,0)$ and (i,j,k). This result holds for any location of the destination and any number and distribution of faulty cubes.*

Note that the role of source and destination can be exchanged if there exists a minimal path between them. However, their roles cannot be exchanged in Theorem 2 (its Corollary and Theorem 2a). Because if a source node is extended safe with respect to a destination, it does not imply that the destination is extended safe with respect to the source.

The following definition gives an extended safety level definition for 3-D meshes. Node (0,0,0) is associated with a vector (E,N,F) to represent the distance to the closest faulty cube along East (positive X), North (positive Y), and Front (positive Z) directions.

Definition 3: *The extended safety level of node (0,0,0) in a given 3-D mesh is a 3-tuple: (E,N,F). This node is extended safe with respect to a destination (i,j,k), with $i,j,k \ge 0$, if $i \le E$, $j \le N$, and $k \le F$, that is, $(i,j,k) \le (E,N,F)$.*

An intuitive explanation of the extended safe node is the following: A node (0,0,0) is extended safe to a destination node as long as there is no faulty cube that goes across the sections between the source and destination nodes along each axis. Based on the Corollary of Theorem 2, there always exists a minimal path between two nodes as long as one node is extended safe with respect to the other. Therefore, vector (E,N,F) at source (0,0,0) can be used to check the existence of a minimal path between the source and a selected destination.

For a general routing where the source node can be in any location, each node is associated with a vector (E,W,N,S,F,B) to represent the distance to the closest faulty cube along East (positive X), West (negative X), North (positive Y), South (negative Y), Front (positive Z), and Back (negative Z) directions. The extended safe node can be defined in a similar way. Symbol ∞ is used if there is no faulty cube in the corresponding direction. A node with (∞,∞,∞,∞,∞,∞) as its extended safety level is called *safe*. Based on this definition, a safe node is the one without faulty cubes along all six directions.

The extended safety level of each node can be calculated through iterative rounds of message exchange among neighboring nodes. Assume that each node knows the status of its neighbors (faulty, enable, and disable). When a node identifies a faulty or disable neighbor, it passes information to the neighbor in the opposite direction. For example, if the neighbor to its East is faulty or disable, the current node passes information (distance: 2 and direction: positive X) to the neighbor to its West. Once a node receives fault information it keeps a copy and increments its distance value by one before forwarding it to the neighbor in the opposite direction. Clearly, each node will receive up to six distance values together with their directions from six different directions. The default value for each direction is ∞; that is, there will be no overhead when there is no fault in a 3-D mesh. Since information is transmitted along one direction in a dimension. The number of (synchronous) rounds needed is bounded by n in an $n \times n \times n$ mesh; that is, $O(\sqrt[3]{N})$ in an N-node 3-D mesh.

5 Fault-Tolerant Adaptive and Minimal Routing

In this section, we first show that any fully adaptive and minimal routing can be applied in 3-D meshes with faulty cubes as long as the destination is extended safe with respect to the source. Then, we extend the result to any partially adaptive and minimal routing. We also show that the planar-adaptive routing [5] fails to meet the partially adaptive routing requirement.

Our fault-tolerant adaptive and minimal routing is based on the following assumptions: (1) Faulty cubes are used as our fault model. (2) The source knows the extended safety level of the destination. (3) Each node knows the status of its adjacent nodes.

Fully adaptive and minimal routing

For the convenience of discussion, we now use $(0, 0, 0)$ as a destination with a safety vector (E, N, F) and (i, j, k) as a source with $i, j, k \geq 0$. Although each node still holds a safety vector (E, W, N, S, F, B), we use here only a subvector (E, N, F) because of the specific locations of source and destination in the assumption.

The routing algorithm consists of two parts: *feasibility check* and *routing*. Feasibility check at the source is used to check if it is possible to perform a minimal routing. This can be easily done by comparing the relative coordinates between the source and destination nodes with the safety vector of the destination.

Before presenting a simple fault-tolerant and minimal routing in 3-D meshes, we first review a fault-tolerant routing algorithm in 2-D meshes [15]. Again, assume that $(0,0)$ is the destination and node (i,j) is the source, with $i,j \geq 0$. If there is no faulty block that goes across the X and Y axes, then there exists at least one minimal path from (i,j) to $(0,0)$, i.e., the length of this path is $|i|+|j|$. Results in [15] also show that any fully adaptive and minimal routing in a 2-D mesh can still be applied if the above condition holds and there is no need of additional fault information during the routing process. Whenever a message reaches a faulty block, it just goes around the block towards the destination and it will never be forced to a detour path or a trap where backtracking is required.

Actually, given a source s and a destination d in a 2-D mesh, all the intermediate nodes of a minimal path between s and d is enclosed in a *region of minimal paths* (RMP) as shown in Figure 4 (a). RMP is constructed by determining two special paths (Path A and Path B) from source to destination. Starting from source, Path A is constructed by going West (negative X) until reaching the Y axis and then along the Y axis going towards the destination. If the path hits a faulty block, it goes around the faulty block by going South (negative Y). Make a south-west turn whenever possible and continue going West (negative X). Path B is constructed in a similar way. Starting from source and going South (negative Y) until reaching the X axis and then going along the X axis toward the destination. If the path hits a faulty block, it goes around the faulty block by going West (negative X). Make a west-south turn whenever possible and continue going South.

It is clear that any fully adaptive and minimal routing for regular 2-D meshes can still be applied to find a minimal path as long as the destination meets the above condition. An intuitive explanation is that because of the convex nature of a faulty block, each faulty block can block at most one dimension. Therefore, at least one dimension remains free for any source and destination pair that spans two dimensions. When the source (or an intermediate node) and destination pair spans only one dimension, the condition associated with the destination ensures that there is no faulty block along that dimension.

FEASIBILITY_CHECK_IN_2D-MESHES
{At source (i,j), a destination $(0,0)$ with a safety vector (E,N)}
Minimal routing is feasible if $(i,j) \leq (E,N)$ and returns YES; otherwise returns NO.

Again, each node in a 2-D mesh holds a safety vector of (E,W,N,S). A subvector (E,N) is used here because of the specific locations of source and destination under the assumption. A more general feasibility check (and routing) can be derived in a straightforward way.

FT-ROUTING_IN_2D-MESHES
{At source (i,j)}
if Feasibility_Check_2D-Meshes = YES

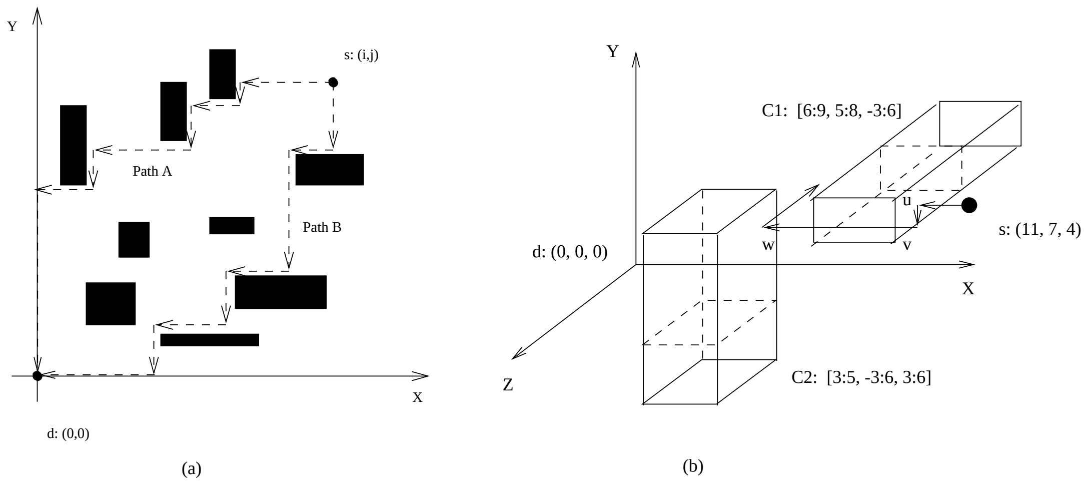

Figure 4: (a) A sample RMP. (b) An example.

then use any fully adaptive and minimal routing in regular 2-D meshes
else this approach cannot be applied.

The above routing algorithm in 2-D meshes can be directly extended to 3-D meshes.

FEASIBILITY_CHECK_3D-MESHES
{At source (i, j, k), a destination $(0, 0, 0)$ with a safety vector (E, N, F)}
Minimal routing is feasible if $(i, j, k) \leq (E, N, F)$ and returns YES; otherwise returns NO.

FT-ROUTING_IN_3D-MESHES
{At source (i, j, k)}
if Feasibility_Check_3D-Meshes = YES
then use any fully adaptive and minimal routing in regular 3-D meshes.

The correctness of the above algorithm can be described as follows: In 3-D meshes, we assume that a source (or an intermediate node) and destination pair spans three dimensions; otherwise, the problem is reduced to minimal routing in 2-D meshes and its correctness follows directly. In other word, at each intermediate node the message can be forwarded along any one of the three directions. When an intermediate node is adjacent to a faulty cube, since the surface of the cube is perpendicular to an axis, the message can still be forwarded along either one of the other directions (a fully adaptive routing algorithm allows this). The disjoint property of faulty cubes ensures that the routing process can still enjoy 2-D freedom until either the faulty cube becomes irrelevant (it does not block any minimal path) and the routing process still enjoys 3-D freedom or it hits one of the three planes, i.e., $x = 0$, $y = 0$, or $z = 0$. Once at one of these three planes, FT-Routing_In_2D-Meshes is applied and the remaining routing process resembles the one in a 2-D mesh.

Note that a faulty cube may intersect with one of the three planes ($x = 0, y = 0$, or $z = 0$). However, it does not cause any problem and the cross section becomes a faulty block in the plane. The routing process itself does not use any fault information, except local fault information of adjacent nodes. In addition, there is no need of faulty cube information, i.e., size and orientation of a faulty cube.

Figure 4 (b) shows a routing example with two faulty cubes. The routing starts from node $(11, 7, 4)$ and goes West (negative X). Once the routing message hits faulty cube C_1: [6:9, 5:8, -3:6] at node u, it turns South (negative Y) and makes a south-west turn at node v (which is the intersection of two adjacent surfaces of faulty block C_1). The routing message then goes West until it hits another faulty cube C_2: [3:5, -3:6, 3:6] at node w. It then turns Back (negative Z). Once the routing message passes the intersection of two adjacent surface of faulty block C_2, the remaining routing resembles the one in a regular 3-D mesh without faulty cubes.

Minimal routing based on planar-adaptive routing

Planar-adaptive routing [5] is one of the popular partially adaptive routings. It offers cost-effectiveness while still keeps a certain degree of adaptivity. Planar-adaptive routing restricts the way the routing message is routed. Specifically, the routing message is routed following a series of 2-D planes A_0, A_1, ... A_{n-1} in an n-D mesh. Each 2-D plane A_i is formed by two dimensions d_i and d_{i+1}. Planes A_i and A_{i+1} share dimension d_{i+1}. However, the order of dimensions is arbitrary. If the offset in dimension d_i is reduced to zero, then routing can be immediatedly shifted to plane A_{i+1}. Apply this routing approach to 3-D meshes, we first construct two planes A_0 and A_1. Assume A_0 contains dimensions Y and Z and plane A_1 contains Z and X (see Figure 5 (a)). Again, assume that the source is (i, j, k) and the destination is $(0, 0, 0)$. The routing starts from (i, j, k) along plane A_0 which is plane $x = i$, once the offset in dimension Y is reduced to zero it switches to plane A_1 which is plane $y = 0$ (see Figure 5 (a)).

Unfortunately, planar-adaptive-routing cannot be directly applied to achieve fault-tolerant and minimal routing using our model. Consider a routing example with source $(3, 3, 3)$ and destination $(0, 0, 0)$. Assume there is a faulty cube $[2 : 4, 1 : 2, -1 : 4]$. Node $(0, 0, 0)$ is extended safe with respect to $(3, 3, 3)$. However, all the minimal paths from $(3, 3, 3)$ in plane $x = 3$ to any node along adjacent line, $x = 3$ and $y = 0$, of A_0 and A_1 are all blocked by the faulty cube. That is, the offset in dimension Y cannot be reduced to zero in A_0 in order to switch to A_1.

However, if we strengthen the constraint at destination $(0, 0, 0)$ (as in Theorem 2a) to: There is

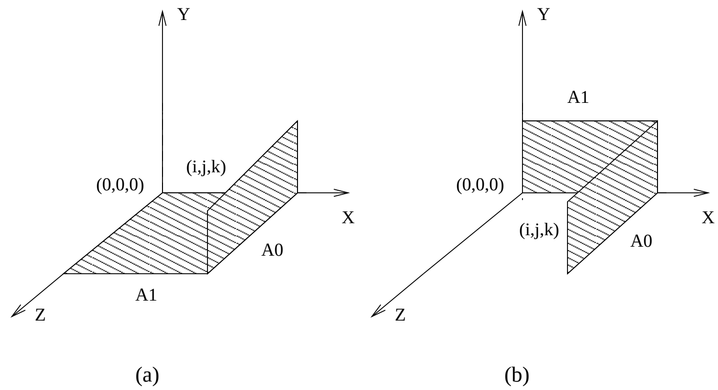

Figure 5: Planer-adaptive routing

no faulty cube that goes across planes $x = 0$, $y = 0$, and $z = 0$, then the planar-adaptive routing can still be applied.

Theorem 3: *Consider a 3-D mesh with faulty cubes. If there is no faulty cube that goes across planes $x = 0$, $y = 0$, and $z = 0$, then the planar-adaptive routing can be applied to any source (i, j, k) to generate a minimal path to $(0, 0, 0)$.*

Proof: Without loss of generality, assume that plane A_0 contains dimensions Y and Z and plane A_1 contains dimensions Z and X. The routing starts from (i, j, k) along plane A_0 (which is plane $x = i$). Plane A_0 can be represented by the one in Figure 4 (a) by treating Y and X as Y and Z, respectively. The condition at destination $(0, 0, 0)$ ensures that there is no faulty block (a cross section of a faulty cube) along the Y or Z axis. The minimal routing in this plane tries to reach any node along the Z axis (the X axis in the figure). The existence of Path B ensures its feasibility. Once the routing message reaches a node along the Z axis (meaning the offset in dimension Y has been reduced to zero), the routing process is then switched to plane A_1. The problem is then reduced to a minimal routing in a 2-D mesh. $\square$

The above result shows that the planar-adaptive routing can still be applied in a 3-D mesh with faulty cubes under a strengthened constraint (i.e., a weaker sufficient condition associated with the destination node). That is, it is less likely for a destination to meet the strengthened constraint than the one based on the extended safety level. Moreover, it is more difficult and expensive for each node to calculate its safety status under the strengthened constraint: Each node has to collect information in three adjacent planes instead of nodes along three dimensions.

Clearly, the above problem stems from the planar-adaptive routing itself which is too restrictive. The question is the existence of other partially adaptive and minimal routing that can still be used under the original extended safety level model.

Partially adaptive and minimal routing

The proposed fault-tolerant and minimal routing applies to any fully adaptive routing (in a regular 3-D mesh) but fails to apply to the planar-adaptive routing unless the constraint on faulty cube distribution is strengthened. Before considering other possible solutions based on partially adaptive approaches, we formally define the concepts of fully and partially adaptive routing.

Minimal routing only consider minimal paths between a given source and destination pair. A *preferred direction* is one along which the corresponding neighbor is closer to the destination. In a 3-D mesh, there are at most three preferred directions, out of six possible directions, for a routing process. Actually, the number of preferred directions is equal to the number of dimensions spanned by the source and destination pair. For example, suppose in a routing the source is $(2, -2, -4)$ and the destination is $(1, 2, -3)$, then preferred dimensions at the source are West, North, and Front. During a minimal routing, the number of preferred directions from an intermediate node to the destination reduces and it eventually becomes zero upon reaching the destination.

Definition 4: *A minimal routing is fully adaptive if it can select any preferred direction at any stage of the routing process. A minimal routing is partially adaptive if it can select from at least two preferred directions at any stage whenever there are two or more preferred directions.*

The X-Y-Z routing is not a partially adaptive routing, since at any stage the routing process can have only one choice. The planar-adaptive routing also fails to meet the partially adaptive routing requirement. In 2-D plane A_i, when it happens that the offset in dimension d_{i+1} is first reduced to zero. It is forced to reduce offset of d_i before switching to 2-D plane A_{i+1}. That is, only one preferred direction can be selected even though more than one may exist.

Among minimal routing approaches that are partially adaptive, we can also rank them in terms of degree of adaptivity. For a given partially adaptive routing, a set of preferred directions that can be selected at an intermediate node (including the source) is called a set of *legitimate preferred directions* at this node. A partially adaptive routing R_1 is *more restrictive* than another one R_2 if at any intermediate node (including the source) the set of legitimate preferred directions of R_1 is a subset of the one of R_2. In addition, the set of legitimate preferred directions of R_1 is a proper subset of the one of R_2 at at least one intermediate node (including the source). Note that the relation "more restrictive" is a partial order; that is, not every two partially adaptive routing algorithms can be compared under this relation.

We introduce here a most restrictive partially adaptive routing called *dynamic planar-adaptive routing*. Like regular planar-adaptive routing, the routing message is routed through a series of 2-D planes. Two adjacent planes still share a common dimension. The difference is that the planes in the series are dynamically generated. Again we use 3-D meshes to illustrate this approach. Suppose we select dimensions Y and Z in A_0, then there are two possible choices in selecting dimensions in A_1. One possibility is dimensions Z and X and the other one is dimensions Y and X. Again, the routing starts from plane $x = i$ and within this plane randomly reduces offsets in dimensions Y and Z. If the offset in dimension Y is reduced to zero before the one in dimension Z, A_1 that spans dimensions Z and X is selected (see Figure 5 (a)); otherwise, A_1 that spans dimensions Y and X is used (see Figure 5 (b)).

Theorem 4: *Consider a 3-D mesh with faulty cubes. If there is no faulty cube that goes across the axes X, Y, and Z, then the dynamic planar-adaptive routing can be applied to any source* (i, j, k) *to generate a minimal path to* $(0, 0, 0)$ *(assuming that destination* $(0, 0, 0)$ *is extended safe with respect to the source).*

Proof: Without loss of generality, assume that plane A_0 contains dimensions Y and Z and plane A_1 contains dimensions Z and X or dimensions Y and X. The routing starts from (i, j, k) along plane A_0 (which is plane $x = i$). Plane A_0 can be represented by the one in Figure 6 (a). Some faulty blocks (cross sections of faulty cubes) may go across both the Y and Z axes. The minimal routing in this plane tries to reach any node along either the Y or Z axis. The existence of such a minimal path can be proved by induction on m, the number of faulty blocks in this plane. When $m = 1$, the routing process starts from (i, j, k) and goes straight along negative Z until reaching the Y axis. If it hits the only faulty block, it turns South and goes along negative Y until reaching the Z axis. Assume that the theorem holds for $m = k - 1$. When $m = k$, we use the same approach for the $m = 1$ case. Again, the routing process starts from (i, j, k) and goes straight along negative Z until reaching the Y axis. If it hits a faulty block, say B (see Figure 6 (a)), there are two cases: (1) If faulty block B intersects with the Z axis, then the routing message turns South and goes along negative Y until reaching the Z axis. (2) If faulty block B does not intersect with the Z axis, it still turns South to the lower-right corner u of the faulty block (see Figure 5). Clearly, faulty block B becomes irrelevant to the remaining minimal routing from u to $(0, 0, 0)$. Since the number of faulty blocks is reduced to no more than $k - 1$, a minimal path exists from u to $(0, 0, 0)$ based on the induction assumption. □

Based on the result of Theorem 4, we conclude that any partially adaptive and minimal routing (which is less restrictive than the dynamic planar-adaptive routing) can be applied in our model.

6 Extensions

In this section, we discuss possible extensions, including an enhanced sufficient condition, the application of the proposed approach in a 3-D torus, and deadlock-free and livelock-free routing.

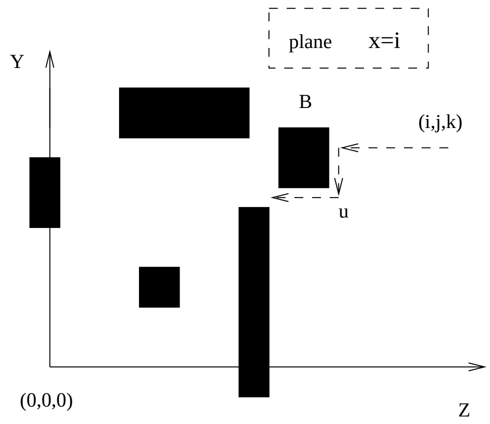

Figure 6: Routing in plane A_0 (i.e. plane $x = i$).

An enhanced sufficient condition

Before considering possible extensions of the proposed model, we first re-examine the sufficient condition in Theorem 2. We first show that the sufficient condition associated with the source cannot be further relaxed. Suppose two axes (out of possible three axes) are clear of faulty cubes, we show that a minimal path may not exist for a given destination. For example, in a system with only one faulty cube $[i-2 : i-1, -1 : j+1, -1 : k+1]$ that goes through the X axis, this faulty cube blocks all the possible minimal paths between (i, j, k) and $(0, 0, 0)$. On the other hand, there are cases when faulty cubes go across one or more axis, a minimal path still exists as in the case of Figure 7 (a) where two faulty cubes C_1 and C_2 go across the Y and X axes, respectively. However, a minimal path exists between nodes $(10, 9, 7)$ and $(0, 0, 0)$. Clearly, a stronger condition cannot be directly associated with node $(0, 0, 0)$.

The following result provides an enhanced sufficient condition for the existence of a minimal path between nodes $(0, 0, 0)$ and (i, j, k). The condition is associated with node u in region $[0 : i, 0 : j, 0 : k]$ (see Figure 7 (b)).

Theorem 5: *Consider two nodes $(0, 0, 0)$ and (i, j, k), with $i, j, k \geq 0$, in a 3-D mesh. If there exists a node $u : (p, q, r)$ with $0 \leq p \leq i$, $0 \leq q \leq j$, and $0 \leq r \leq k$ such that nodes along three line sections $x = p$ and $y = q$, $x = p$ and $z = r$, $y = q$ and $z = r$ within region $[0 : i, 0 : j, 0 : k]$ are fault-free, a minimal path exists between $(0, 0, 0)$ and (i, j, k).*

Proof: Applying Theorem 2 to nodes u and s with node u being the source and s the destination, we know that there exists a minimal path between u and s. Then apply Theorem 2 to nodes u and d with node u being the source and d the destination. Again we find a minimal path between d and u.

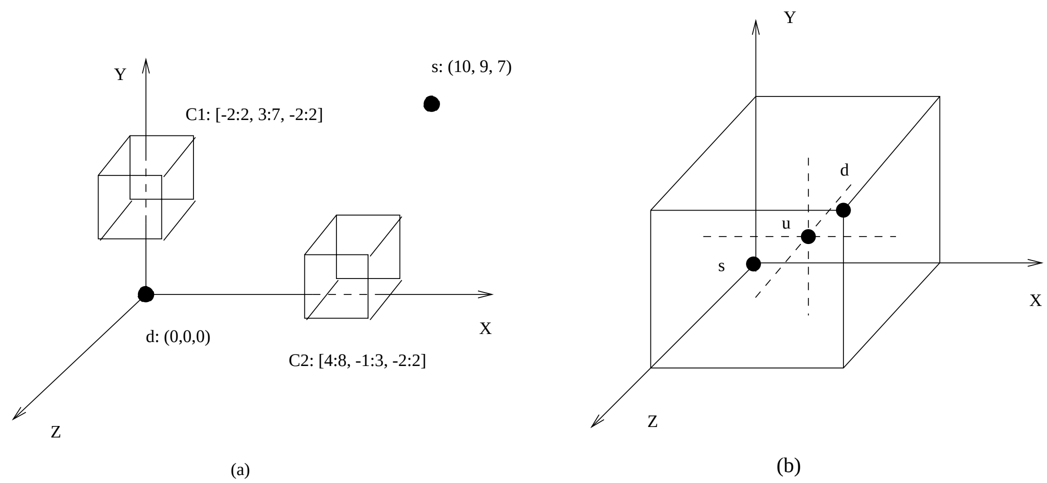

Figure 7: (a) A failure condition. (b) An enhanced sufficient condition.

Combining these two minimal paths, we construct a minimal path between s and d. □

Clearly, Theorem 2 is a special case of Theorem 5 with node s chosen as node u.

Extensions to 3-D meshes with boundary

Our approach can also be applied to k-ary 3-dimensional meshes. That is, the number of nodes along a dimension is bounded by k. We can add healthy "ghost" nodes along the boundary of each dimension. In this way, a given mesh with boundary is converted to the one without boundary. Faulty cubes can still be defined in the same way. For example, the southwestern and front corner (a node with three adjacent "ghost" nodes along West, South, and Front) has an extended safety level (-, ∞, -, ∞, ∞, -), where - represents a component that depends on fault distribution in the given 3-D mesh.

Notice the deference between our approach and the one proposed by Chien and Kim [5]. In [5], "ghost" nodes along the boundary of each dimension are considered faculty. Actually, the rule for enable/disable nodes is more complicated. Corner nodes (including ones with two or more adjacent "ghost" nodes) are considered to have only one adjacent "ghost" node. Based on the Chien and Kim's faulty region definition which is the same as the faulty cube definition, a faulty boundary node (a node with at least one adjacent "ghost" node) will disable the whole 2-D boundary plane that contains this node (one of the six 2-D boundary planes of a 3-D mesh)! We can easily prove that if there is one fault in each cross section (along an axis) of a given 3-D mesh, all the nodes in the 3-D mesh will be marked disable based on Chien and Kim's model.

Consider an example of a $k \times k \times k$ mesh as shown in Figure 8. Suppose there is one column

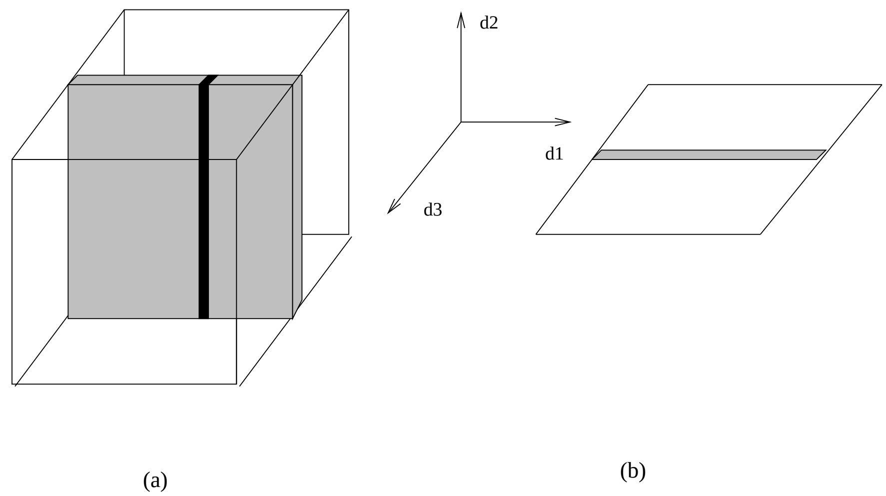

Figure 8: (a) Plane $z = c$, and (b) plane $y = i$, with $0 \leq i \leq k-1$ in a $k \times k \times k$ mesh.

of faulty nodes (the black column in Figure 8 (a)) in $z = c$ cross section. Based on Chien and Kim's faulty region definition, the complete $z = c$ cross section will be disabled. Then for each cross section $y = i$, $0 \leq i \leq k-1$, as shown in Figure 8, since one strip (marked gray) is marked disable, the corresponding cross section is also disabled. Therefore, all the nodes in the given 3-D mesh are marked disable, although there are only as few as k faulty nodes among k^3 nodes! However, using our faulty cube model, all nonfaulty nodes are marked enable in the example of Figure 8. In other word, any partially adaptive and minimal routing can still be applied as long as the destination meets the safety requirement.

Note that Chien and Kim's planar adaptive routing cannot be applied using the fault model proposed in this paper, not even for non-minimal routing. Consider again the example of Figure 8, with two planes A_0 (X and Y) and A_1 (Y and Z). Suppose the source is on cross section $z = c$ and is at the east side of the faulty column, and the destination is at the southwestern corner of another cross section $z = c'$ ($c \neq c'$). Clearly, regular planar adaptive routing fails, since the offset in dimension X cannot be reduced to zero in plane $z = c$, although there exists a minimal path. Applying adaptive planar adaptive routing, the offset in dimension Y will be first reduced to zero and then the routing process continues on A_1 (X and Z) to the destination.

Extensions to 3-D tori

A torus is a mesh with wraparound connections. Since a 3-D mesh is a subgraph of a 3-D torus, any solutions for 3-D meshes can be directly applied to 3-D tori. However, since a 3-D torus has extra connections, solutions can be simplified and cost can be reduced. Another difference is that a faulty cube in a 3-D torus may affect the safety level of a node in both directions of a dimension because of the wraparound links.

However, once the extended safety level has been decided at each node, the same sufficient condition (Theorem 2 and its Corollary) can be applied in a 3-torus. Specifically, when source and destination nodes are randomly distributed, say source $(i^{'}, j^{'}, k^{'})$ with safety vector (E, W, N, S, F, B) and destination (i, j, k), the conditions in the Corollary of Theorem 2 can be changed to the following: $|i - i^{'}| \leq W$ (if $i < i^{'}$), $|i - i^{'}| \leq E$ (if $i > i^{'}$), $|j - j^{'}| \leq S$ (if $j < j^{'}$), $|j - j^{'}| \leq N$ (if $j > j^{'}$), $|k - k^{'}| \leq F$ (if $k > k^{'}$), and $|k - k^{'}| \leq B$ (if $k < k^{'}$). Similarly, Theorem 5 can be applied to a 3-D torus.

Deadlock and livelock freedom

Deadlock due to dependencies on consumption resources (such as channels) is a fundamental problem in routing [7]. A deadlock involving several routing processes occurs when there is a cyclic dependency for consumption channels. *Livelock* occurs when a routing message travels around its destination node, never reaching it because the channels required to do so are occupied by other messages. Livelock is relatively easy to avoid, actually, any minimal routing is livelock-free [7].

Unlike many non-minimal fault-tolerant routing algorithms, the deadlock issue in the proposed model can be easily solved through the use of *virtual network* [11] where a given physical network consists of several virtual networks. Each virtual network is partitioned into several virtual channels arranged in such a way that no cycle exists among channels, i.e., there is no *intra-virtual-network cycle*.

A partition of a 3-D mesh into eight subnetworks: X+Y+Z+, X+Y+Z−, X+Y−Z+, X+Y−Z−, X−Y+Z+, X−Y+Z−, X−Y−Z+, X−Y−Z−. Figure 9 shows the subnetwork X+Y+Z+. Depending on the relative location of the source and destination nodes, one of the eight virtual subnetworks is selected and the corresponding routing can be completed within the selected subnetwork without using any other subnetwork. In this way, any *inter-virtual-network cycle* is avoided. Converting to virtual channel usage, our approach needs four virtual channels. To reduce the number of virtual channels, eight subnetworks can be pairwised to form four subnetworks: X−Y−Z∗, X∗Y+Z−, X∗Y+Z+, X+Y−Z∗, where ∗ stands for + and −, i.e., a bidirectional channel. Figure 9 (b) shows the X∗Y+Z+ subnetwork. Clearly, at most three virtual channels are required along each dimension. We can easily show that three virtual channels are required for dynamic planar adaptive routing for minimal routing. For minimal routing, since fully adaptive and minimal routing needs only three virtual channels, the dynamic planar adaptive routing, a special partially adaptive routing, needs no more than three virtual channels. Note that the planar adaptive routing also requires two virtual channels. Therefore, within the context of minimal routing in 3-D meshes, dynamic planar adaptive routing offers better

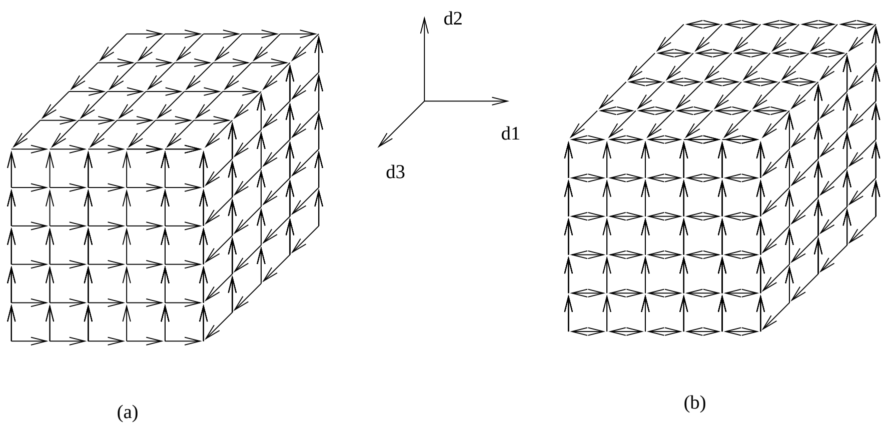

Figure 9: (a) A network partition X+Y+Z+. (b) A network partition X∗Y+Z+.

fault tolerance and adaptivity without using extra virtual channels compared with planar adaptive routing.

Applying our approach to n-dimensional meshes, dynamic planar adaptive routing requires n virtual channels [17]. Note that Linder and Harden's virtual network approach requires 2^n virtual channels and Chien and Kim's planar adaptive approach requires three virtual channels. For low-dimensional meshes, such as 3-D meshes, the difference between regular planar adaptive routing and dynamic planar adaptive routing, in terms of the number of required virtual channels, is insignificant.

7 Conclusion

In this paper we have proposed a sufficient condition for minimal routing in 3-D meshes with faulty cubes. Unlike many traditional models that assume all the nodes know global fault distribution or only adjacent fault information, our approach is based on the concept of limited global fault information. Specifically, we have proposed a simple fault-tolerant adaptive and minimal routing approach based on the proposed extended safety level information associated with each node in 3-D meshes. We also have shown that any partially adaptive and minimal routing can be applied in our approach as long as the destination node meets a certain condition.

Our approach is the first attempt to provide insight on the design of fault-tolerant and minimal routing in 3-D meshes. The proposed extended safety level model is a practical model that captures fault information in a concise format and supports various applications such as minimal routing. This study has shown that the safety level concept for binary hypercubes can still by effectively used in low-dimensional meshes with a proper extension. Our future research will focus on extending the proposed approach to high-dimensional meshes and to collective communications [12] which include multicast, broadcast, and barrier synchronization.

References

[1] R. V. Boppana and S. Chalasani. Fault tolerant wormhole routing algorithms for mesh networks. *IEEE Transactions on Computers.* 44, (7), July 1995, 848-864.

[2] Y. M. Boura and C. R. Das. Fault-tolerant routing in mesh networks. *Proc. of 1995 International Conference on Parallel Processing.* 1995, I, 106 - 109.

[3] M. S. Chen and K. G. Shin. Depth-first search approach for fault-tolerant routing in hypercube multicomputers. *IEEE Transactions on Parallel and Distributed Systems.* 1, (2), April 1990, 152-159.

[4] X. Chen and J. Wu. Minimal routing in 3-D mehses using extended safety levels. *Proceedings of ISATED International Conference on Parallel and Distributed Systems.* Oct. 1998, to appear.

[5] A. A. Chien and J. H. Kim. Planar-adaptive routing: Low-cost adaptive networks for multiprocessors. *Journal of ACM.* 42, (1), January 1995, 91-123.

[6] W. J. Dally. The J-machine: System support for Actors. *Actors: Knowledge-Based Concurrent Computing.* Hewitt and Agha (eds.), MIT Press, 1989.

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

[8] E. Fleury and P. Fraigniaud. A general theory for deadlock avoidance in wormhole routed networks. *IEEE Transactions on Parallel and Distributed Systems.* 9, (7), July 1998, 626-638.

[9] R. K. Koeninger, M. Furtney, and M. Walker. A shared memory MPP from Cray research. *Digital Technical Journal.* 6, (2), Spring 1994, 8-21.

[10] R. Libeskind-Hadas and E. Brandt. Origin-based fault-tolerant routing in the mesh. *Proc. of the 1st International Symposium on High Performance Computer Architecture.* 1995, 102-111.

[11] D. H. Linder and J. C. Harden. An adaptive and fault tolerant wormhole routing strategy for k-ary n-cubes. *IEEE Transactions on Computers.* 40, (1), Jan. 1991, 2-12.

[12] D. K. Panda. Issues in designing efficient and practical algorithms for collective communication on wormhole-routed systems. *Proc. of the 1995 ICPP Workshop on Challenges for Parallel Processing.* Aug. 1995, 8-15.

[13] C. C. Su and K. G. Shin. Adpative fault-tolerant deadlock-free routing in meshes and hypercubes. *IEEE Transactions on Computers.* 45, (6), June 1996, 672-683.

[14] J. Wu. Adaptive fault-tolerant routing in cube-based multicomputers using safety vectors. *IEEE Transactions on Parallel and Distributed Systems.* 9, (4), April 1998, 321-334.

[15] J. Wu. Fault-tolerant adaptive and minimal routing in mesh-connected multicomputers using extended safety levels. *IEEE Transactions on Parallel and Distributed Systems.* 11, (2), Feb. 2000, 149-159.

[16] J. Wu. Reliable unicasting in faulty hypercubes using safety levels. *IEEE Transactions on Computers.* 46, (2), Feb. 1997, 241-247.

[17] J. Wu. A theory of fault-tolerant adaptive and minimal routing in n-dimensional meshes. in progress.