

# A Fault-Tolerant Adaptive and Minimal Routing Approach in $n$ -D Meshes \*

Jie Wu

Department of Computer Science and Engineering  
Florida Atlantic University  
Boca Raton, FL 33431  
jie@cse.fau.edu

## Abstract

*In this paper a sufficient condition is given for minimal routing in  $n$ -dimensional ( $n$ -D) meshes with faulty nodes contained in a set of disjoint fault regions. It is based on an early work of the author on minimal routing in low dimension meshes (such as 2-D meshes with faulty blocks). Unlike many traditional models that assume all the nodes know global fault distribution, our approach is based on the concept of limited global fault information. First, a fault model called fault region is used in which all faulty nodes in the system are contained in a set of disjoint regions. Fault information is coded in a  $2n$ -tuple called extended safety level associated with each node of an  $n$ -D mesh to support minimal routing. Specifically, we study the existence of minimal paths at a given source node, limited distribution of fault information, minimal routing, and deadlock-free routing. Our results show that any minimal routing that is partially adaptive can still be applied as long as the destination node meets a certain safety condition. A dynamic planar-adaptive routing scheme is presented that offers better fault tolerance and adaptivity than the regular planar-adaptive routing scheme in  $n$ -D meshes.*

## 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. The *mesh-connected topology*

is one of the most thoroughly investigated network topologies for multicomputers. Many multicomputers that use 2-dimensional (2-D) meshes include the MIT J-machine, the Symult 2010, and the Intel Touchstone. The CRAY T3D and T3E systems use a 3-D torus.

As the number of nodes in a mesh-connected multicomputer increases, the chance of failure also increases. The complex nature of networks also makes them vulnerable to disturbances which can be either deliberate or accidental. Therefore, the ability to tolerate failure becomes increasingly important, especially in the communication subsystem. Several studies have been conducted which achieve fault tolerance by adding (or deleting) extra components of the system. However, adding and deleting nodes and/or links require modifications of network topologies which may be expensive and difficult. We focus here on achieving fault tolerance using the inherent redundancy present in the mesh-connected multicomputer, without adding spare nodes and/or links.

An important and challenging issue is to extend communication subsystems which include various routing algorithms to cope with faulty components. To this end, fault models and routing algorithms are the two keys to successfully extend the existing approaches. We use a convex type of fault region as our fault model and propose a novel *information model* in which each node in a mesh-connected multicomputer collects and distributes fault information concurrently but in a decentralized way. To ensure that this approach is scalable for a large and complex network, only specially coded fault information is distributed rather than detailed information. Unlike many existing information models that require each node to have knowledge of the entire network, the coded fault information associated with each node represents limited-global information by exploring locality of disturbances in the network. This approach also reduces the memory requirement [6] to store fault information at each node. When a disturbance occurs, only those affected nodes need to update local information to keep it consistent.

---

\*This work was supported in part by NSF grant CCR 9900646 and a grant from Motorola Inc.

The *safety-level-based* (or *safety-vector-based*) routing [9, 11], 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. This approach differs from many existing ones where information is brought by the *header* of the routing message [1], and the routing function is defined based on header information and local state of the current node [5]. In this 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  $m$  (an integer), there is at least one Hamming distance (or minimal) path from this node to any node within Hamming-distance- $m$  [11]. Using the safety level (or safety vector) associated with each node, a routing algorithm 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 binary hypercubes but less efficient when it is directly applied to mesh topologies such as 2-D and 3-D meshes. In [10], the author introduced the concept of *extended safety level* with its use in achieving minimal routing in 2-D meshes with faulty nodes contained in a set of faulty blocks.

In this paper, the extended safety level concept is further extended for general  $n$ -D meshes. The challenge is to find a minimal path in an  $n$ -D mesh with faulty nodes contained within a set of disjoint *fault regions* (a fault model extended from the commonly used faulty block model in 2-D meshes). The amount of limited-global-information should be kept minimum and be easy to obtain and maintain. 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 routing. The concept of partial adaptive routing is defined and a dynamic planar-adaptive routing approach is proposed that trades routing adaptivity for a simple deadlock-free routing with a better fault tolerance capability than Chien and Kim's regular planar-adaptive routing [2]. Our approach is the first attempt to address the minimal routing in  $n$ -D meshes with faulty nodes using limited fault information.

The collision-free routing in the presence of obstacles is also studied in other fields such as routing urban vehicles, motion planning in robotics, wire routing in VLSI and logistics in operations research. The focus in these fields are different. For example, given a set of obstacles and two points in the plane, most studies try to find a shortest path, not necessarily a minimal one, among all the avail-

able collision-free paths. In addition, most problems are optimization problems associated with a certain optimization function such as minimum number of bends as in VLSI routing. See [7] for a survey of research results in these fields. All proofs for the results in this paper are omitted due to space limitation.

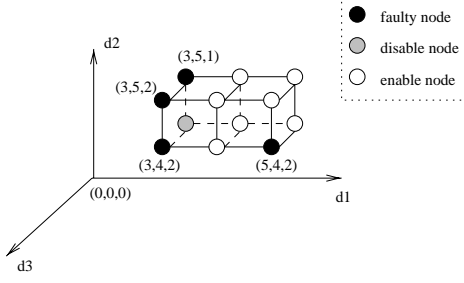
## 2 Preliminaries

A  $k$ -ary  $n$ -dimensional mesh with  $k^n$  nodes (also called  $n$ -D mesh) has an interior node degree of  $2n$  and a network diameter of  $n(k - 1)$ . Each node  $u$  has an address:  $(u_1, u_2, \dots, u_n)$ , where  $u_i \in \{0, 1, \dots, k - 1\}$  corresponds to the location of  $u$  in dimension  $d_i$ . Two nodes  $v: (v_1, v_2, \dots, v_n)$  and  $u: (u_1, u_2, \dots, u_n)$  are connected if their addresses differ in one and only one element (dimension), say dimension  $d_i$ ; moreover,  $|u_i - v_i| = 1$ . In the subsequent discussion, an  $n$ -D mesh is considered without specifying its size. A coordinate system in an  $n$ -D mesh is defined in such a way that it has a "floating" origin in node  $(0, 0, \dots, 0)$  adjustable by constant  $c$ . A  $d_i$ -dimensional axis (denoted as  $d_i$ -axis) consists of nodes with addresses  $(0, 0, \dots, 0, u_i, 0, \dots, 0)$ , where  $u_i$  is an integer. Because of the floating origin of the mesh, we can assume without loss of generality that the source in a routing is  $(0, 0, \dots, 0)$  and the destination is  $(u_1, u_2, \dots, u_n)$ . Along each dimension  $d_i$ , there are two directions: positive  $d_i$  (also  $d_i+$ ) and negative  $d_i$  ( $d_i-$ ). Along the positive  $d_i$  (negative  $d_i$ ) dimension, the coordinate in dimension  $d_i$  increases (decreases). Clearly, there are  $2n$  directions in an  $n$ -D mesh.

A routing process is *minimal* if the length of the routing path from source  $(0, 0, \dots, 0)$  to destination  $(u_1, u_2, \dots, u_n)$  is the distance between these two nodes, i.e.,  $\sum_{i=1}^n |u_i|$ . Throughout this paper, we focus on minimal routing in an  $n$ -D mesh with faulty nodes (link faults can be treated as node faults by disabling the corresponding adjacent nodes). To simplify the routing process, a labeling scheme is introduced to quickly identify those nonfaulty nodes that cause routing difficulty and disables them. As a result, a set of convex-type of *fault regions* is formed.

**Definition 1:** *In an  $n$ -D mesh, a nonfaulty node is either marked enabled or disabled. Initially, all nonfaulty nodes are marked enabled. A nonfaulty node is marked disabled if there are two or more disabled or faulty neighbors along different dimensions. A fault region contains all the connected disabled and faulty nodes.*

Based on Definition 1, there are three types of nodes: faulty nodes, enabled nodes, and disabled nodes. The node status can be easily determined through rounds of status exchanges among neighboring nodes. It is assumed that both source and destination nodes in a routing process are non-faulty and they are marked enabled. A fault region has the



**Figure 1. Fault regions in a 3-D mesh.**

following desirable features that facilitate simple and minimal routing.

**Theorem 1:** *In an  $n$ -D mesh, a fault region defined by Definition 1 has the following properties: (1) Every neighbor of a fault region has one and only one faulty or disabled neighbor (in the fault region). (2) The distance between any two fault regions is at least two.*

With the first property of a fault region (also called the convex feature), the address of a fault region can be simply described by a range along each dimension, e.g.,  $f_i : f'_i$ , with  $f_i \leq f'_i$ , specifying the range along dimension  $d_i$ . A general fault region can be represented by  $[f_1 : f'_1, f_2 : f'_2, \dots, f_n : f'_n]$  which covers  $\prod_{i=1}^n (f'_i - f_i + 1)$  nodes. When the range along a dimension is one, say  $f_i : f_i$ , the number of dimensions spanned by the corresponding region is reduced by one.

For  $n$ -D meshes with boundary, we can add “ghost” nodes around boundary nodes to change these nodes to regular interior nodes. Ghost nodes are assumed to be non-faulty and are marked enabled. Note that in some models such as the one proposed by Chien and Kim [2], a pessimistic fault model is used. Specifically, one of the ghost nodes has to be marked disabled, and hence, generating more disabled nodes in the given mesh.

In 3-D meshes, fault regions are called *faulty cubes*. Figure 1 shows a 3-D mesh with four faulty nodes (3,4,2), (3,5,1), (3,5,2), and (5,4,2). Based on Definition 1, these four faulty nodes generate two disjoint faulty cubes: [3:3, 4:5, 1:2] (also called a *faulty block* in 2-D meshes), and [5:5, 4:4, 2:2], a single node.

### 3 Extended Safety Level

In this section, an information model representing fault distribution is studied to represent fault distribution. We first extend the safety level concept to  $n$ -D meshes. The *safety level* [11] concept was originally proposed to capture limited global information in a binary hypercube. It was extended to 2-D meshes as *extended safety level* [10] which

includes four elements, each of which indicates the distance to the closest faulty block to East (E), South (S), West (W), and North (N) 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  $n$ -D meshes and it serves as a basis of our routing approach.

**Theorem 2:** *Assume that node  $(0, 0, \dots, 0)$  is the source and node  $(u_1, u_2, \dots, u_n)$  is the destination. If there is no fault region that intersects with any of the axes, there exists at least one minimal path between  $(0, 0, \dots, 0)$  and  $(u_1, u_2, \dots, u_n)$ . This result holds for any location of the destination and any number and distribution of fault regions.*

The above result can be strengthened by including the location of destination  $(u_1, u_2, \dots, u_n)$ .

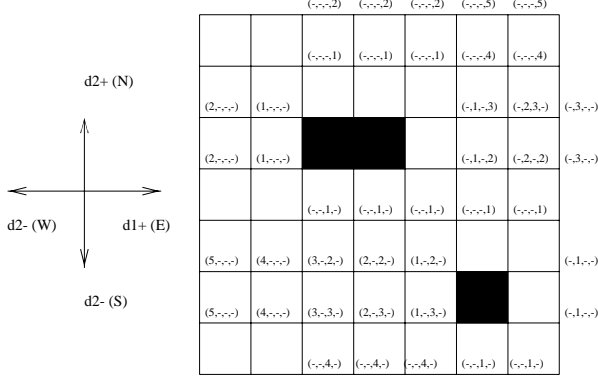
**Corollary 2:** *If there is no fault region that intersects with the sections of  $[0, u_i]$  along the  $d_i$ -axis for all  $i \in \{1, 2, \dots, n\}$ , there exists at least one minimal path between the source and destination nodes.*

Note that the role of source and destination can be interchanged if there exists a minimal path between them. That is, if there exists a minimal path from a source to a destination, then there exists a minimal path from the destination to the source. However, their roles cannot be interchanged in Theorem 2 (and Corollary 2). 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  $n$ -D meshes. Source node  $(0, 0, \dots, 0)$  is associated with a  $2n$ -tuple  $(p_1, n_1, p_2, n_2, \dots, p_n, n_n)$ , where  $p_i$  and  $n_i$  represent the distance to the closest fault region along the positive and negative  $d_i$  dimensions, respectively. Basically, extended safety level is coded information about fault distribution in the neighborhood. Such information can be used to determine the existence of a minimal path between a given pair of source and destination nodes. Symbol “-” is used to represent the fact that there is no fault region along the corresponding direction. A node is called *safe* if its extended safety level is  $(-, -, \dots, -)$ ; otherwise, it is *unsafe*.

**Definition 2:** *The extended safety level of node  $(0, 0, \dots, 0)$  in a given  $n$ -D mesh is a  $2n$ -tuple:  $(p_1, n_1, p_2, n_2, \dots, p_n, n_n)$ . This node is extended safe with respect to a destination  $(u_1, u_2, \dots, u_n)$ , if  $|u_i| < p_i$  (when  $u_i > 0$ ) and  $|u_i| < n_i$  (when  $u_i < 0$ ) for all  $i \in \{1, 2, \dots, n\}$ ; otherwise, it is extended unsafe.*

In a 3-D mesh, 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  $d_1$ ), West (negative  $d_1$ ),



**Figure 2. A faulty  $8 \times 8$  2-D mesh with extended safety levels associated with unsafe nodes.**

North (positive  $d_2$ ), South (negative  $d_2$ ), Front (positive  $d_3$ ), and Back (negative  $d_3$ ) directions. In a 2-D mesh, each node is associated with a vector  $(E, W, N, S)$ . Figure 2 shows a faulty  $8 \times 8$  2-D mesh with extended safety levels associated with unsafe nodes.

An intuitive explanation of the extended safe node is the following: A node is extended safe to a destination node, as long as there is no fault region that intersects with the sections between the source and the destination along each axis. Based on Corollary 2, there always exists a minimal path between two nodes, as long as one node is extended safe with respect to the other.

Like fault regions, the extended safety level of each node can be calculated through iterative rounds of message exchanges among neighboring nodes. Assume that each node knows the status of its neighbors (faulty, enabled, and disabled). When a node identifies a faulty or disabled neighbor, it passes information to the neighbor in the opposite direction. For example, if the neighbor to its positive  $d_i$  dimension is faulty or disabled, the current node passes information (distance: 2 and direction: positive  $d_i$ ) to its neighbor at the negative  $d_i$  dimension. 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  $2n$  distance values together with their directions from  $2n$  different directions. The default value for each direction is  $-$ ; that is, there will be no overhead when there is no fault in an  $n$ -D mesh. Since information is transmitted along one direction in a dimension. The number of (synchronous) rounds of message exchanges among neighboring nodes is bounded by  $k$  in a  $k$ -ary  $n$ -D mesh; that is,  $O(k)$  in a  $k^n$ -node  $n$ -D mesh.

## 4 Routing Protocols

The fault-tolerant adaptive and minimal routing is based on the following assumptions: (1) The fault region defined early is used as the fault model. (2) The source knows the extended safety level of the destination. (3) Each node knows the status of its adjacent nodes. (4) Only the static fault model is used, i.e., it is assumed that no new fault occurs during a routing process.

### Fully adaptive and minimal routing

For the convenience of feasibility checking of a routing process,  $(0, 0, \dots, 0)$  is now considered as a destination with a  $2n$ -tuple  $(p_1, p_2, \dots, p_n)$  and  $(u_1, u_2, \dots, u_n)$  as a source (still  $u_i \geq 0$  for all  $i \in \{1, 2, \dots, n\}$ ). The way the safety status of the destination is collected at the source is discussed in [10]. Although each node still holds a safety vector  $(p_1, n_1, p_2, n_2, \dots, p_n, n_n)$ , a subvector is used 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 applied 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.

#### FEASIBILITY\_CHECK\_ $n$ -D-MESHES

{At source  $(u_1, u_2, \dots, u_n)$ , destination  $(0, 0, \dots, 0)$   
with safety vector  $(p_1, p_2, \dots, p_n)$ }  
Minimal routing is feasible if  $(u_1, u_2, \dots, u_n) \leq$   
 $(p_1, p_2, \dots, p_n)$  and returns YES;  
otherwise, returns NO.

#### FT-ROUTING\_IN\_ $n$ -D-MESHES

{At source  $(u_1, u_2, \dots, u_n)$ }  
**if** Feasibility\_Check\_ $n$ -D-Meshes = YES  
**then** apply any fully adaptive and minimal routing in  
regular  $n$ -D meshes.  
**else** the proposed routing approach cannot be applied.

Instead of proving the correctness of the above approach, let us look at its application in 2-D meshes. Again, assume that  $(0, 0)$  is the destination and node  $(u_1, u_2)$  is the source, with  $u_1, u_2 \geq 0$ . If there is no faulty block that intersects with the  $d_1$ -axis and  $d_2$ -axis, then there exists at least one minimal path from  $(u_1, u_2)$  to  $(0, 0)$ , i.e., the length of this path is  $|u_1| + |u_2|$ . Results in [10] 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.

The correctness of the proposed algorithm can be described as follows through induction on dimension  $n$ . Clearly, this algorithm works for 2-D meshes. Assume that this algorithm works for meshes with up to  $n - 1$  dimensions. In  $n$ -D meshes, we assume that a source (or an intermediate node) and destination pair spans  $n$  dimensions; otherwise, the problem is reduced to minimal routing in  $l$ -D meshes (with  $l < n$ ) and its correctness follows directly. In other words, at each intermediate node the message can be forwarded along any one of the  $n$  dimensions. When an intermediate node is adjacent to a fault region, since each neighbor of a fault region is adjacent to exactly one disabled (or faulty) node in the fault region, the message can still be forwarded along either one of the other directions (a fully adaptive routing algorithm allows this). The disjoint property of fault regions ensures that the routing process can still enjoy  $(n - 1)$ -D freedom until it hits either a new fault region or the offset along one dimension  $d_i$  is reduced to zero, i.e.,  $u_i = 0$ . Based on the induction, a minimal path is guaranteed in the remaining routing process.

### Partially adaptive and minimal routing

*Deadlock* due to dependencies on consumption resources (such as channels) is a fundamental problem in routing. A deadlock involving several routing processes occurs when there is a cyclic dependency for consumption channels. To ensure freedom of deadlock and to support a “truly” fully adaptive routing without using the flow control mechanism, Linder and Harden [8] showed a *virtual network* approach that requires  $O(2^n)$  *virtual channels* [3] in an  $n$ -D mesh. In this approach, each physical channel may support several logical or virtual channels multiplexed across the physical channel. The reason for using multiple channels is to avoid cyclic dependencies among channels to prevent deadlock. Other simpler approaches [4] exist that support fully adaptive routing using constant number of virtual channels. However, routing decisions have to be made based on accurate buffer status, i.e., routing and flow control have to be coupled.

*Planar-adaptive routing* [2] is one of the popular partially adaptive routings that requires few virtual channels (three), and at the same time, allowing flow control and routing to be decoupled. It offers cost-effectiveness in preventing deadlock 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  $A_1, A_2, \dots, A_n$  in an  $n$ -D mesh. Each 2-D plane  $A_i$  is formed by two dimensions  $d_i$  and  $d_{i+1}$ . Plane  $A_n$  consists of dimensions  $d_n$  and  $d_1$ . 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 immediat-

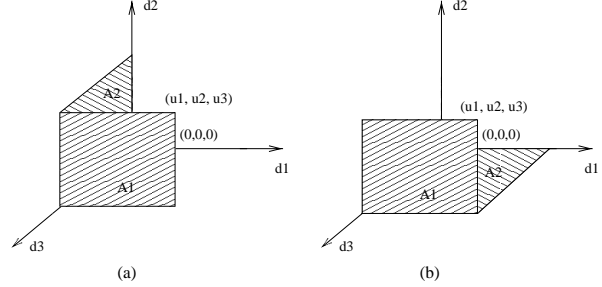


Figure 3. Planar-adaptive routing.

edly shifted to plane  $A_{i+1}$ . Apply this routing approach to 3-D meshes, we first construct two planes  $A_1$  and  $A_2$ . Assume that  $A_1$  contains dimensions  $d_1$  and  $d_2$  and plane  $A_2$  contains  $d_2$  and  $d_3$  (see Figure 3 (a)). Again, the source is  $(u_1, u_2, u_3)$  and the destination is  $(0, 0, 0)$ . The routing starts from  $(u_1, u_2, u_3)$  along plane  $A_1$  which is plane  $d_3 = u_3$ , once the offset in dimension  $d_1$  is reduced to zero it switches to plane  $A_2$  which is plane  $d_1 = 0$  (see Figure 3 (a)).

Unfortunately, planar-adaptive-routing cannot be directly applied to achieve fault-tolerant and minimal routing using our model. Consider a routing example in a 3-D mesh with source  $(3, 3, 3)$  and destination  $(0, 0, 0)$ . Assume that there is a faulty cube  $[1 : 2, -1 : 4, 2 : 4]$ . Clearly, all the minimal paths from  $(3, 3, 3)$  in plane  $A_1 : d_3 = 3$  to any node along adjacent line,  $d_3 = 3$  and  $d_1 = 0$ , of  $A_1$  and  $A_2$  are all blocked by the faulty cube.

**Theorem 3:** Consider an  $n$ -D mesh with fault regions. If there is no fault region that intersects with 2-D planes  $d'_1 = d'_2 = \dots = d'_{n-2} = 0$ , where  $\{d'_i | i \in \{1, 2, \dots, n-2\}\}$  is a subset of  $\{d_i | i \in \{1, 2, \dots, n\}\}$  then the planar-adaptive routing can be applied in FT-ROUTING\_IN- $n$ -D-MESHES to any source  $(u_1, u_2, \dots, u_n)$  to generate a minimal path at  $(0, 0, \dots, 0)$ .

The above result shows the planar-adaptive routing can still be applied in an  $n$ -D mesh with fault regions under a strengthened constraint (i.e., a weaker sufficient condition associated with the destination node). Note that there are  $\binom{n}{2}$  2-D planes defined in Theorem 3, compared with  $n$  axes in Theorem 2. 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 needs to collect information in  $\binom{n}{2}$  adjacent planes instead of nodes along  $n$  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 in FT-ROUTING\_IN- $n$ -

D-MESHES. Before considering other possible solutions based on partially adaptive approaches, let us formally define 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 neighbor is closer to the destination. In an  $n$ -D mesh, there are at most  $n$  preferred directions, out of  $2n$  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 in a 3-D mesh the source is  $(2, -2, -4)$  and the destination is  $(1, 2, -3)$ , then preferred dimensions at the source are West (negative  $d_1$ ), North (positive  $d_2$ ), and Front (positive  $d_3$ ). 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 3:** A minimal routing is fully adaptive if it can select any preferred direction at any step of the routing process. A minimal routing is partially adaptive if it can select from at least two preferred directions at any step whenever there are two or more preferred directions.

The traditional X-Y routing in 2-D meshes is not a partially adaptive routing, since at any step 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 exists.

Minimal and partially adaptive routing can also be ranked in terms of degree of adaptivity. 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  $d_1$  and  $d_2$  in  $A_1$ , then there are two possible choices in selecting dimensions in  $A_2$ . One possibility

is dimensions  $d_1$  and  $d_3$  and the other one is dimensions  $d_2$  and  $d_3$ . Again, the routing starts from plane  $A_1 : d_3 = u_3$ , and within this plane, randomly reduces offsets in dimension  $d_1$  and dimension  $d_2$ . If the offset in dimension  $d_1$  is reduced to zero before one in dimension  $d_2$ ,  $A_2$  that spans dimensions  $d_2$  and  $d_3$  is selected (see Figure 3 (a)); otherwise,  $A_2$  that spans dimensions  $d_1$  and  $d_3$  is used (see Figure 3 (b)). Assuming that destination  $(0, 0, \dots, 0)$  is extended safe with respect to the source), we have the following result.

**Theorem 4:** Consider an  $n$ -D mesh with fault regions. If there is no fault region that intersects with any axis  $d_i$ ,  $i \in \{1, 2, \dots, n\}$ , then the dynamic planar-adaptive routing can be applied in FT-ROUTING\_IN- $n$ -D-MESHES to any source  $(u_1, u_2, \dots, u_n)$  to generate a minimal path to  $(0, 0, \dots, 0)$

## 5 Extensions

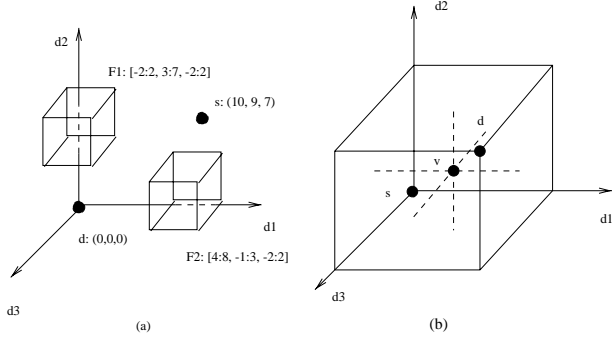
### An enhanced sufficient condition

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. Again, we start with 3-D meshes. 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  $[u_1 - 2 : u_1 - 1, -1 : u_2 + 1, -1 : u_3 + 1]$  that goes through the  $d_1$  axis, this faulty cube blocks all the possible minimal paths between  $(u_1, u_2, u_3)$  and  $(0, 0, 0)$  (assuming  $u_1 > 2, u_2, u_3 \geq 0$ ). On the other hand, there are cases when faulty cubes intersect one or more axis, a minimal path still exists as in the case of Figure 4 (a) where two faulty cubes  $F_1$  and  $F_2$  intersect the  $d_2$  and  $d_1$  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  $(u_1, u_2, u_3)$ . The condition is associated with node  $v$  in region  $[0 : u_1, 0 : u_2, 0 : u_3]$  (see Figure 4 (b)).

**Theorem 5:** Consider two nodes  $(0, 0, 0)$  and  $(u_1, u_2, u_3)$  in a 3-D mesh. If there exists a node  $v : (v_1, v_2, v_3)$  with  $0 \leq v_i \leq u_i$ , such that nodes along three line sections  $d_1 = v_1$  and  $d_2 = v_2$ ,  $d_1 = v_1$  and  $d_3 = v_3$ ,  $d_2 = v_2$  and  $d_3 = v_3$  within region  $[0 : u_1, 0 : u_2, 0 : u_3]$  are fault-free, a minimal path exists between  $(0, 0, 0)$  and  $(u_1, u_2, u_3)$ .

Theorem 5 can be easily proved by applying Corollary 2 twice: one at the source-destination pair  $(v, s)$  and another one at the  $(v, d)$ . In the following, we use  $(v_1, v_2, \dots, v_{i-1},$



**Figure 4. (a) A failure condition. (b) An enhanced sufficient condition.**

$*, v_{i+1}, \dots, v_n$ ) to represent a line section  $(d_1 = v_1) \wedge (d_2 = v_2) \wedge \dots \wedge (d_{i-1} = v_{i-1}) \wedge (d_{i+1} = v_{i+1}) \wedge \dots \wedge (d_n = v_n)$ , that is, this line section is along dimension  $d_i$ .

**Corollary 5:** Consider two nodes  $(0, 0, \dots, 0)$  and  $(u_1, u_2, \dots, u_n)$  in an  $n$ -D mesh. If there exists a node  $v: (v_1, v_2, \dots, v_n)$  with  $0 \leq v_i \leq u_i$ , such that nodes along  $n$  line sections  $(*, v_2, \dots, v_n), (v_1, *, \dots, v_n), \dots, (v_1, v_2, \dots, v_{n-1}, *)$  within region  $[0 : u_1, 0 : u_2, \dots, 0 : u_n]$  are fault-free, a minimal path exists between  $(0, 0, \dots, 0)$  and  $(u_1, u_2, \dots, u_n)$ .

Clearly, Corollary 2 is a special case of Corollary 5 with node  $s$  being chosen as node  $v$ .

### Extensions to $n$ -D tori

Our results here can be easily extended to an  $n$ -D torus. A torus is a mesh with wraparound connections. Because an  $n$ -D mesh is a subgraph of an  $n$ -D torus, any solutions for  $n$ -D meshes can be directly applied to  $n$ -D tori. However, since an  $n$ -D torus has extra connections, solutions can be simplified and cost can be reduced. Another difference is that a fault region in an  $n$ -D torus may affect the safety level of a node in both directions of a dimension because of the wraparound links.

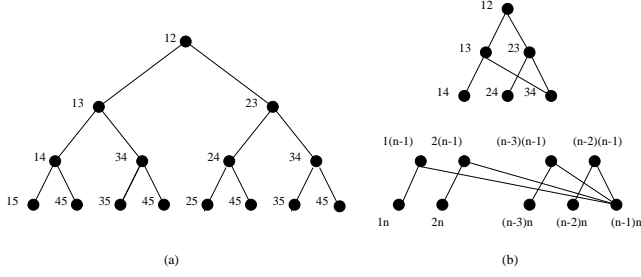
Once the extended safety level has been decided at each node of an  $n$ -D mesh, the same sufficient condition (Theorem 2 and Corollary 2) can be extended and applied in the  $n$ -D torus. Specifically, when the source and destination nodes are randomly distributed, say source  $(u_1, u_2, \dots, u_n)$  with safety vector  $(p_1, n_1, p_2, n_2, \dots, p_n, n_n)$  and destination  $(0, 0, \dots, 0)$ , the conditions in the Corollary 2 can be changed to the following:  $|u_i| < p_i$  (if  $u_i \bmod k \geq -u_i \bmod k$ ) or  $|u_i| < n_i$  (if  $u_i \bmod k \leq -u_i \bmod k$ ) for all  $i$ , where  $1 \leq i \leq n$ .

### Deadlock- and livelock-freedom

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* [8], 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*.

Again, we use a 3-D mesh as an example. Partition a 3-D mesh into eight subnetworks:  $d_1 + d_2 + d_3 +, d_1 + d_2 + d_3 -, d_1 + d_2 - d_3 +, d_1 + d_2 - d_3 -, d_1 - d_2 + d_3 +, d_1 - d_2 + d_3 -, d_1 - d_2 - d_3 +, d_1 - d_2 - d_3 -$ . 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, this approach needs four virtual channels. For example, if source and destination are  $(u_1, u_2, u_3)$  and  $(0, 0, 0)$ , respectively, if  $u_1 > 0, u_2 < 0$ , and  $u_3 > 0$ , subnetwork  $d_1 - d_2 + d_3 -$  is selected. To reduce the number of virtual channels, eight subnetworks can be paired to form four subnetworks:  $d_1 - d_2 - d_3 *, d_1 * d_2 + d_3 -, d_1 * d_2 + d_3 +, d_1 + d_2 - d_3 *$ , where  $*$  stands for  $+$  and  $-$ , i.e., a bidirectional channel supported by one virtual channel. Clearly, at most three virtual channels are needed along each dimension. It has been shown in [12] that three virtual channels are required for dynamic planar adaptive routing for minimal routing. Therefore, 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.

In general, the adjacent 2-D planes in the dynamic planar-adaptive routing form a tree as shown in Figure 5 (a). Each routing process start from the root and reaches the destination through a branch in the tree. Here,  $ij$  ( $i, j \in \{1, 2, \dots, n\}$  and  $i < j$ ) represents a 2-D plane that includes dimensions  $d_1$  and  $d_2$ . Clearly, 2-D planes that have the same dimensions can be combined to form a directed acyclic graph (DAG) as shown in Figure 5 (b). Based on Figure 5 (b), a series of virtual subnetworks  $VSN_{ij}$  is constructed, with one subnetwork for each node  $ij$  of the DAG in Figure 5 (b). Specifically,  $VSN_{ij} : d_i + d_j +, d_i + d_j -, d_i - d_j +, d_i - d_j -$ . Clearly, two virtual channels are used in  $VSN_{ij}$ . Because each  $i \in \{1, 2, \dots, n\}$  appears exactly  $n - 1$  times in the DAG of Figure 5 (b),  $2(n - 1)$  virtual channels are needed. A total order on  $VSN$  can be defined based on the partial order defined in Figure 5 (b). The use of these subnetworks strictly follows the total order to avoid cyclic dependency among virtual subnetworks  $VSN$ . To reduce the overall number of virtual channels,  $VSN_{ij}$  can be defined as  $VSN_{ij} : d_i + d_j *, d_i - d_j *$ . Three vir-



**Figure 5. (a) The tree structure of adjacent 2-D planes. (b) The directed acyclic graph (DAG).**

tual channels are used for  $VSN_{ij}$  with one for  $d_i$  and two for  $d_j$ . Because there are  $\frac{n(n-1)}{2}$   $VSN$ 's and  $n$  dimensions, based on the principle of pigeon hole, at least  $\frac{3(n-1)}{2}$  virtual channels are needed. The following theorem shows that using another way of virtual network formation, it is possible to further reduce the number of virtual channels to  $n(n+1)$  if  $n$  is even in an  $n$ -D mesh.

**Theorem 6:** *It is possible to construct a set of virtual networks with  $n(n+1)$  (when  $n$  is even) virtual channels for an  $n$ -D mesh to ensure freedom of deadlock using dynamic planar-adaptive routing.*

This theorem can be proved by constructing  $n$  virtual networks  $VSN_i$ :  $d_i + d_{i+1} * d_{i+2} * \dots * d_{i+k} *$ ,  $d_i - d_{i+1} * d_{i+2} * \dots * d_{i+k} *$  for  $i \in \{1, 2, \dots, n\}$ , where  $d_{i+j}$  (including  $j = 0$ ) should be interpreted as  $d_{(i+j-1) \bmod n+1}$ . We assume that dimension  $n$  is odd (if the given dimension is even, add one to it to make it odd) and  $k = \frac{n-1}{2}$ .  $VSN_i$  covers  $k$  nodes  $d_i d_{i+1}, d_i d_{i+2}, \dots, d_i d_{i+k}$  of Figure 5 (b). Sequence  $VSN_1, VSN_2, \dots, VSN_n$  also respects the partial order of Figure 5 (b) starting from the root of the DAG. In  $VSN$ 's each  $d_i *$  appears  $2k$  times,  $d_i +$  and  $d_i -$  each once; therefore,  $2k + 1 = n$  virtual channels are used.

## 6 Conclusions

We have proposed a general theory of minimal routing in  $n$ -D meshes with fault regions. Unlike many traditional models that assume all the nodes know global fault distribution or only adjacent fault information, our approach is based on the limited distribution of fault information. Specifically, we have proposed a fault-tolerant adaptive and minimal routing algorithm based on the proposed extended safety level information associated with each node in  $n$ -D meshes. Our approach is the first attempt to provide insight on the design of fault-tolerant and minimal routing in  $n$ -D meshes. Our future work will focus on extending the approach in a system with dynamic faults.

## References

- [1] 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.
- [2] 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.
- [3] W. J. Dally and C. L. Seitz. Deadlock-free message routing in multiprocessor interconnection networks. *IEEE Transactions on Computers*. 36, (5), May 1987, 547-553.
- [4] J. Duato. A necessary and sufficient condition for deadlock-free adaptive routing in wormhole networks. *IEEE Transactions on Parallel and Distributed Systems*. 6, (10), 1995, 1,055-1,067.
- [5] 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.
- [6] P. Fraigniaud and C. Gavoille. Memory requirement for universal routing schemes. *Proc. of the 14th Annual ACM Symp. on Principles of Distributed Computing*. 1995, 223-230.
- [7] D. T. Lee, C. D. Yang, and C. K. Wong. Rectilinear paths among rectilinear obstacles. *Discrete Applied Mathematics*. 70, 1996, 185-215.
- [8] 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.
- [9] 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.
- [10] 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.
- [11] J. Wu. Reliable unicasting in faulty hypercubes using safety levels. *IEEE Transactions on Computers*. 46, (2), Feb. 1997, 241-247.
- [12] J. Wu. A simple fault-tolerant adaptive and minimal routing approach in 3-D meshes. *Proc. of the 7th Int'l Conf. on Parallel and Distributed Systems (ICPADS)*. July 2000, to appear.