Extended Minimal Routing in 2-D Meshes with Faulty Blocks *

Jie Wu
Dept. of Computer Science and Engineering
Florida Atlantic University
Boca Raton, FL 33431

Zhen Jiang
Dept. of Computer Science
West Chester University
West Chester, PA 19383

Abstract

In this paper several enhanced sufficient conditions are given for minimal routing in 2-dimensional (2-D) meshes with faulty nodes contained in a set of disjoint faulty blocks. It is based on an early work of Wu's minimal routing in 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 notion of *limited global fault information*. First, a fault model called faulty block is proposed in which all faulty nodes in the system are contained in a set of disjoint faulty blocks. Fault information is coded in a 4-tuple called *extended safety level* associated with each node of a 2-D mesh to determine the feasibility of minimal routing. Specifically, we study the existence of minimal route at a given source node based on the associated extended safety level, limited distribution of faulty block information, and minimal routing. An analytical model for the number of rows and columns that receive faulty block information is also given. Extensions to Wang's minimal-connected-components (MCCs) are also considered. MCCs are rectilinear-monotone polygonal shaped fault blocks and are refinement of faulty blocks. Our simulation results show substantial improvement in terms of higher percentage of minimal routing in 2-D meshes under both fault models.

Keywords: *Fault models, fault tolerance, high assurance, minimal routing, 2-D meshes.*

*This work was supported in part by NSF grants CCR 9900646 and ANI 0073736. Contact E-mail: jie@cse.fau.edu.

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 packets 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 packets 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. A *minimal routing* always routes the packet to the destination through a shortest path. The *mesh-connected topology* is one of the most thoroughly investigated network topologies for multicomputer systems. The 2-dimensional (2-D) mesh is the most popular mesh-connected topology. Some multicomputers were built based on the 2-D mesh [4, 10, 11].

As the number of nodes in a 2-D mesh increases, the chance of failure also increases. The complex nature of networks also makes them vulnerable to disturbances. Therefore, the ability to tolerate failure is becoming increasingly important for obtaining high assurance systems, especially in the communication subsystems. Several studies have been conducted which achieve fault tolerance by adding (or deleting) extra components of the system [5, 14]. 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 2-D meshes, 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 model* and *information model* are the two keys to successfully extending the existing approaches. We use a convex shape of fault region called faulty block as our fault model and propose a novel information model in which each node in a 2-D mesh collects and distributes coded fault information (rather than detailed fault information) to support scalability. Unlike many existing information models that require each node to have knowledge of the entire network, coded fault information associated with each node represents limited global information by exploring locality of disturbances in the network. It also reduces the memory requirement [7] to store fault information at each node. When a disturbance occurs, only those affected nodes update their information to keep it consistent.

The *safety-level-based* routing [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. This approach differs from many existing ones where information is brought by the header of the routing packet [3], and the routing function is defined based on header information and local state of the current node [6]. In this approach, neighborhood fault information is captured by an integer (safety level) associated with each node. For example, in a binary hypercube, if a node's safety level is L (an integer), there is at least one Hamming distance (or minimal) path from this node to any node within Hamming-distance-L in the hypercube [18]. Using the safety level associated with each node, a routing algorithm can obtain an optimal or suboptimal solution. In addition, it 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 meshes. In [17], Wu 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. Basically, extended safety level associated with each node contains distances to closest faulty blocks in four directions (East, South, West, and North). When the extended safety level of the source meets a safety requirement (also called sufficient safe condition) with respect to its distance to the destination, minimal routing is guaranteed. Extended safety level only guarantees the existence of a minimal route. To facilitate the minimal routing, faulty block information also needs to be distributed to nodes along four boundary lines of each faulty block.

However, the sufficient safe condition in [17] is over conservative. In addition, only minimal routing is considered. In this paper, we provide several extended sufficient safe conditions without adding routing complexity. The notion of sub-minimal routing is also introduced together with a sufficient condition. Specifically, we address the existence of a minimal path at a given source node based on the associated extended safety level, limited distribution of faulty block information, and minimal and sub-minimal routing in these extensions. An analytical model for the number of rows and columns that receive faulty block information is also given. Extensions to Wang's minimal-connected-components (MCCs) [15] are also considered. MCCs are rectilinear-monotone polygonal shaped fault blocks and are refinement of faulty blocks. The size of faulty blocks is reduced by considering the relative locations of source and destination nodes during the block formation process. The purpose of MCC is to facilitate minimal routing in the presence of faults; that is, a node is included in an MCC block only if using that node in a routing will definitely make the route non-minimal.

Many studies have been done on routing in 2-D meshes based on the faulty block model [1, 2, 8, 12, 13, 19]. Most approaches try to reduce either the number of nonfaulty nodes in a faulty block by considering different types of fault regions or the number of virtual channels to support deadlock-free routing. Relatively few work has been done in minimal routing in 2-D meshes with faulty blocks. In other fields, faulty-block-based routing in 2-D meshes is called collision-free routing in the presence of obstacles. These fields include routing urban vehicles, motion planning in robotics, wire routing in VLSI and logistics in operations research. The focuses in these fields are different. Most problems are optimization problems associated with a certain optimization function such as the minimum number of bends as in VLSI routing. See [9] for a survey of research in these fields.

This paper is organized as follows: Section 2 presents some preliminaries, including extended safety levels in 2-D meshes under both the faulty block model and the MCC model, and Wu's safety-level-based minimal routing in 2-D meshes. Section 3 proposes extended sufficient safe conditions under both fault models. Section 4 discusses several implementation issues and an analytical model is also given. Simulation results are presented and we compare them with optimal results in Section 5. Section 6 concludes the paper.

2 Preliminaries

2-D meshes. An $n \times m$ 2-D mesh with $n \times m$ nodes has an interior node degree of 4. Each node u has an address (x_u, y_u), where $0 \leq x_u < n$ and $0 \leq y_u < m$. Two nodes $u : (x_u, y_u)$ and $v : (x_v, y_v)$ are connected if their addresses differ by one in one and only one dimension. Basically, nodes along each dimension are connected as a linear array. The distance between two nodes s and d, $D(s, d)$, is equal to $\mid x_d - x_s \mid + \mid y_d - y_s \mid$. Assume that node u is the current node, d is the destination node, and v is a neighbor of node u. v is called a *preferred neighbor* with respect to destination d if $D(v, d) < D(u, d)$; otherwise, it is called a *spare neighbor*. Respectively, the corresponding connecting directions are called *preferred direction* and *spare direction*. A minimal routing always selects a preferred neighbor at each hop towards the destination.

Block fault model. Most existing literatures on fault-tolerant routing use disjoint rectangular blocks to model node faults and to avoid routing difficulties in meshes. A node-labeling scheme that identifies nodes is defined as follows:

Definition 1: *In a 2-D mesh, a non-faulty node is initially labeled enabled; however, its status is changed to disabled if there are two or more disabled or faulty neighbors in different dimensions. Connected disabled and faulty nodes form a faulty block.*

A faulty block consists of faulty and disabled (but non-faulty) nodes. In a 2-D mesh, an enabled node is an *adjacent node* of a faulty block if it has one faulty or disabled neighbor in that faulty block. A *corner* of a faulty block is defined as an enabled node with two adjacent nodes of that faulty block in different dimensions. In Figure 1 (a), eight faults (3,3), (3,4), (4,4), (5,4), (6,4), (2,5), (5,5), and (3,6) form a rectangle [2:6, 3:6]. In general, $[x_{min} : x_{max}, y_{min} : y_{max}]$ represents a faulty block.

Minimal-connected-components (MCCs). The minimal-connected-component (MCC) [15] was proposed to reduce the size of a faulty block by "removing" four "corner sections" of the block based on

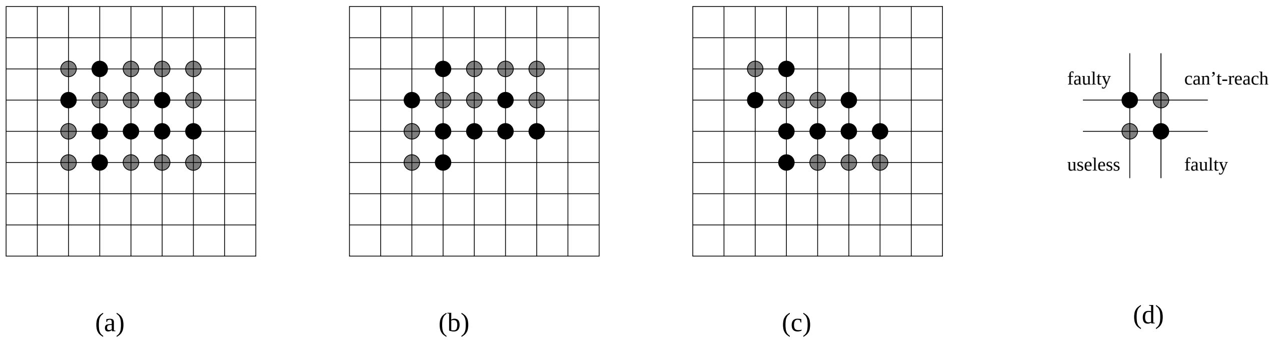

Figure 1: (a) A faulty block consisting of faulty and disabled nodes. (b) The corresponding type-one MCC. (c) The corresponding type-two MCC. (d) Useless and can't-reach nodes for quadrant I destinations.

the relative location of the source and destination. If the destination is at the first or third quadrant of the source, the NW and SE corner sections are removed and the corresponding MCC is called *type-one MCC*. If the destination is at the second or fourth quadrant of the source, the SW and NE corner sections are removed and the corresponding MCC is called *type-two MCC*.

We assume that the destination is at the first quadrant of the source to illustrate MCCs. Two types of disabled nodes are used: *useless* and *can't-reach* nodes. A node labeled useless is such a node that once it is entered in a routing, the next move must take either west or south direction, making a minimal routing (in the first quadrant) impossible. A node labeled can't-reach is such a node that to enter it in a routing, a west or south move must be taken, making a minimal routing impossible. The node status (faulty, fault-free, useless, and can't-reach) can be determined through a labeling procedure. Connected faulty, useless, and can't-reach nodes form an MCC. The following definition [15] defines a type-one MCC:

Definition 2: *Initially, all faulty nodes are labeled as* faulty *and all non-faulty nodes as* fault-free. *If node u is fault-free, but its north neighbor and east neighbor are faulty or useless, u is labeled* useless. *If node u is fault-free, but its south neighbor and west neighbor are faulty or can't-reach, u is labeled* can't-reach. *Connected faulty, useless, and can't-reach nodes form an MCC.*

The labeling procedure for quadrant II (the destination is in northwest) can be derived from that for quadrant I by exchanging the roles played by east and west neighbors. Again, the labeling procedure for quadrant IV (III) is from one for quadrant II (I) by exchanging the role of useless and can't-reach. MCCs generated from quadrants II and IV (I and III) are the same. Figure 1 (d) shows the definition of useless and can't-reach nodes (for quadrant I destinations). Figure 1 (a) shows a sample faulty block based on Definition 1. Figure 1 (b) shows the corresponding type-one MCCs (for quadrants I & III destinations). The corresponding type-two MCC (for quadrants II & IV destinations) is shown in Figure 1 (c).

Since a node may have different status (i.e., whether it is an MCC or non-MCC node) for quadrant I/III or quadrant II/IV routing, each node carries two status: $(status_1, status_2)$ where $status_1$ is for quadrant I/III routing and $status_2$ is for quadrant II/IV routing. Referring to Figures 1 (b) and (c) again, the status of node (4,3) is (fault-free, fault-free). The one for (2,6) is (fault-free, disabled), for (4,5) is (disabled, disabled), and for (2,3) is (disabled, fault-free). It is assumed that source and destination nodes both have (fault-free, fault-free) status.

Extended safety level. The extended safety level [17] of a node in a given 2-D mesh is a 4-tuple: (E, S, W, N), where E stands for the distance from this node to the closest faulty block to its East. S, W, and N are defined in a similar way. To ensure a minimal path from source node, Wu [17] provided the following safe node definition:

Definition 3: *Assume that source node s:*$(0,0)$ *has an extended safety level* (E, S, W, N) *and destination node is d:*(x_d, y_d)*, with* $x_d, y_d \geq 0$*. The source node is safe with respect to d if* $x_d \leq E$ *and* $y_d \leq N$*; otherwise, it is unsafe.*

Theorem 1 [17]: *If the source node is safe with respect to the destination node, a minimal path is guaranteed from source to destination.*

The proof of Theorem 1 can be done constructively starting from the destination. A west-bound message (see Figure 2 (a)) from the destination is sent from the destination until reaching the y axis. Then, the message follows the y axis to reach the origin (source). If the message hits a faulty block before reaching the y axis, it routes around the block by going south to reach the SE corner of the block, and then it continues west-bound. Similarly, another minimal path can be constructed by a south-bound message initiated from the destination as shown in Figure 2 (a). In fact, Wu [17] has shown that the region enclosed by the west-bound and south-bound paths includes intermediate nodes and only intermediate nodes of all minimal paths between the source and destination.

A node at source (0,0) is safe if section $[0, x_d]$ of the x axis and section $[0, y_d]$ of the y axis are both clear of any faulty block. Clearly, the above safety level model and the corresponding result are still applicable to 2-D meshes with MCCs. Figure 2 (b) illustrates the sufficient safe condition for source $(0,0)$ and destination (x_d, y_d).

Faulty-block-information used in minimal routing. The extended safety level associated with the source is used only to ensure the existence of a minimal route from the source to a given destination. However, extended safety level information only is not sufficient to support minimal routing. In the

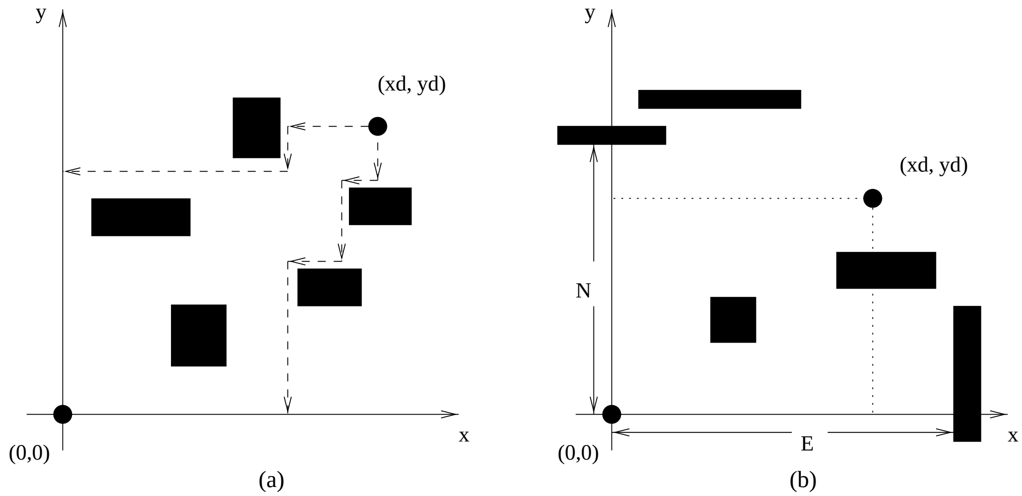

Figure 2: (a) Constructing minimal paths from the destination. (b) Sufficient safe condition.

example of Figure 3 (a), source (0,0) is safe with respect to a destination in region R_4. Without any other information, a minimal route may not be guaranteed even if a preferred neighbor is selected at each hop towards the destination. For example, consider any minimal routing that forwards the packet to a preferred neighbor. Once the packet enters region R_8, no more minimal route is possible (all such routes are blocked by the faulty block). Clearly, more information is needed to guarantee a minimal route.

In [17], Wu presented a *faulty-block-information* model to facilitate a minimal route. Faulty block information of each block (i.e., x_{min}, y_{min}, x_{max}, y_{max} in block $[x_{min} : x_{max}, y_{min} : y_{max}]$) is distributed to all nodes in the adjacent lines of the block. These lines (L_1, L_2, L_3, and L_4 in Figure 3 (a)) are called *boundaries* of that faulty block. Figure 3 (b) shows an example of boundaries of multiple faulty blocks for quadrant I destinations (i.e., $x_d, y_d \geq 1$). The boundaries start from two corners (NE-corner $(x_{max}+1, y_{max}+1)$ and SW-corner $(x_{min}-1, y_{min}-1)$) of each faulty block and go forward along with each direction of X and Y dimensions. Without any other faulty block, the propagation of boundary information is forwarded node by node in each direction until it reaches an edge of the mesh. If a boundary L_i intersects with another faulty block, a turn is made towards L_i of the encountered faulty block. Another turn is made at the corner of the second faulty block to join L_i (see Figure 3 (b) where L_3 of block i joins L_3 of block j).

In [17], Wu proposed a routing algorithm based on the faulty block information for any routing from safe source s:$(0,0)$ to destination d:(x_d, y_d), with $x_d, y_d \geq 0$. The safety status of the source is determined from the extended safety level associated with the source. The routing starts from a safe source and

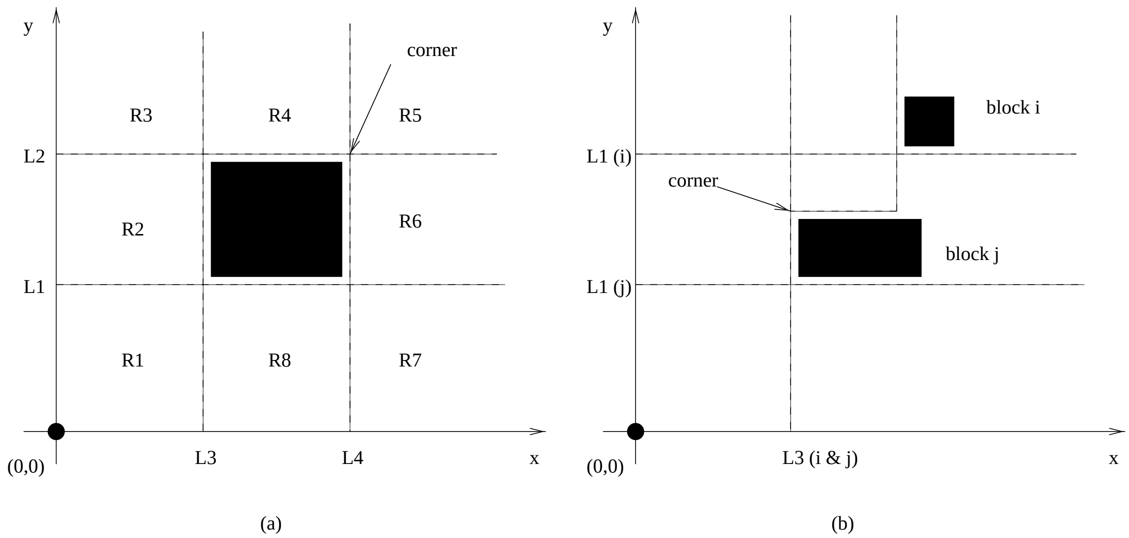

Figure 3: (a) Boundaries of a faulty block. (b) Boundaries of multiple faulty blocks for quadrant I destinations.

uses any adaptive minimal routing until the boundary of any faulty block is met. If the selection of any preferred neighbor does not affect the minimal routing, the path is *non-critical*; otherwise, it is *critical*. In case of a critical path, one of preferred directions cannot be selected in a minimal routing due to the effect of faulty blocks. Such a direction is called *preferred but detour direction*. The selection should be done at current node u:(x_u, y_u) based on the relative location of the destination in Wu's protocol [17]:

WU'S PROTOCOL
case current node u
  On the left section of L_1 of any faulty block:
    If the destination is in the area of R_6 divided by the boundaries of that faulty block, the routing packet should stay on L_1 until reaching the intersection of L_1 and L_4; otherwise, the next hop can be any preferred direction.
  On the lower section of L_3 of any faulty block:
    If the destination is in the area of R_4 divided by the boundaries of that faulty block, the routing packet should stay on L_3 until reaching the intersection of L_3 and L_2; otherwise, the next hop can be any preferred direction.
  Otherwise:
    Select any preferred direction.
end case

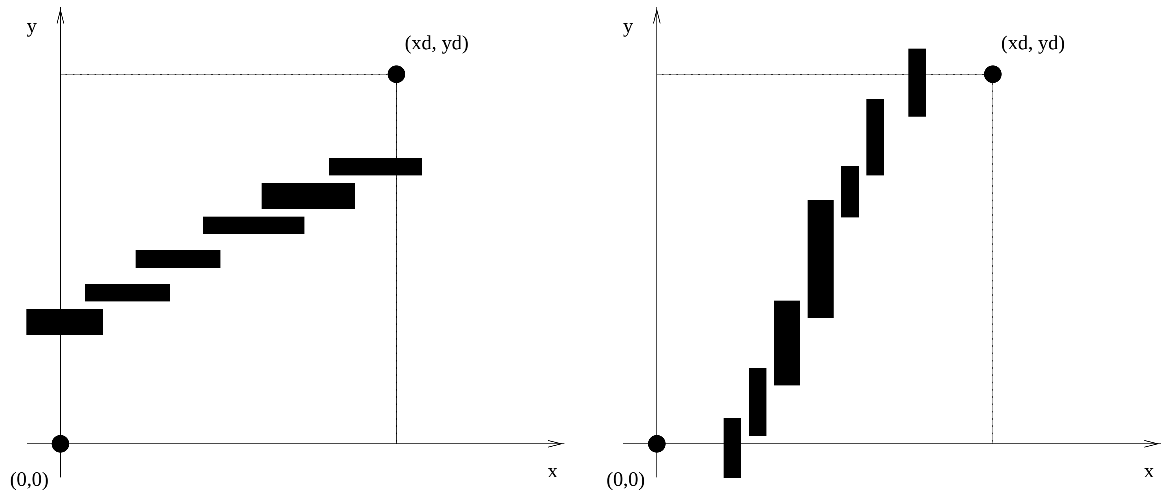

Figure 4: (a) A sequence of blocks covers s and d on y and (b) a sequence of blocks covers s and d on x.

For example, as shown in Figure 3 (b), when the routing packet from $(0, 0)$ meets the lower section of L_3 of faulty block j, it also meets L_3 of faulty block i. If the destination is not in R_4 of block i or R_4 of block j, the routing process is still non-critical and any of two preferred directions (positive X and positive Y) can be selected; otherwise, the routing is critical and the packet cannot be forwarded to positive X. In this case, positive X is the preferred but detour direction and positive Y is the only preferred direction that can be selected to construct a minimal path. Again, faulty-block-information used in minimal routing is the same under the MCC model.

Necessary and sufficient conditions. Wang [15] gave a necessary and sufficient condition for the existence of a minimal path in 2-D meshes with either faulty blocks or MCCs. We use faulty blocks to illustrate, assuming that the destination is in the first quadrant. Block i is represented as $[x(i)_{min} : x(i)_{max}, y(i)_{min} : y(i)_{max}]$. *Block i covers block j on y* if $y(i)_{min} > y(j)_{max}$ and $x(i)_{min} \leq x(j)_{max} \leq x(i)_{max}$. *A sequence of blocks $1, 2, ..., k$ cover source $s : (0, 0)$ and destination $d : (x_d, y_d)$ on y* if (a) block $i+1$ covers block i on y, for $i = 1, 2, ..., k-1$, (b) $x(1)_{min} \leq 0 \leq x(1)_{max}$ and $y(1)_{min} \geq 0$, and (c) $x(k)_{min} \leq x_d \leq x(k)_{max}$ and $y(k)_{max} \leq y_d$. The notions of *block i covers block j on y* and *a sequence of blocks $1, 2, ..., k$ cover source $s : (0, 0)$ and destination $d : (x_d, y_d)$ on x* are defined in a similar way by exchanging the role of x and y. Wang's necessary and sufficient condition can be stated as follows: A minimal route from s to d exists if and only if no sequence of blocks exists that covers s and d on x and no sequence of blocks exists that covers s and d on y. The notion of coverage is illustrated in Figure 4. However, global information of fault distribution is needed to apply this necessary and sufficient condition.

3 Extended Sufficient Conditions

In this section, we consider three extensions of the sufficient safe condition in Definition 2. Basically, we try to strengthen the sufficient safe condition without resorting to global fault information as used in Wang's necessary and sufficient condition. Note that in a global-fault-information model each node stores $\theta(n^2)$ of global information in an $n \times n$ 2-D mesh. Assume that source node s:$(0,0)$ has an extended safety level (E, S, W, N) and destination node is d:(x_d, y_d), with $x_d, y_d \geq 0$.

Extension 1. In the first extension, the sufficient safe condition is enhanced by taking into consideration of neighbors' (preferred or spare) safety status. First, the notion of *sub-minimal routing* is defined: sub-minimal routing is a minimal routing with one detour. A *detour* occurs when a spare neighbor is selected once during a routing process. Clearly, the length of a sub-minimal path is the length of the corresponding minimal path plus two. In this extension, the additional information stored at each node is of constant amount (information collected from four neighbors).

Theorem 1a: *Minimal routing exists if the source node is safe or one of the preferred neighbors is safe (with respect to the destination); otherwise, sub-minimal routing exists if one of the spare neighbors is safe with respect to the destination.*

When the source is unsafe, but one of its neighbors (preferred or spare) is safe, then the routing process consists of two phases: send the routing packet from the source to the selected neighbor, and then, apply Wu's protocol with the selected neighbor being the new source.

Extension 2. In this extension, the sufficient condition is extended to cover cases where either section $[0, x_d]$ of the x axis is clear of faulty blocks or section $[0, y_d]$ of the y axis is clear of faulty blocks, but not both. Suppose the extended safety level of the source is (E, S, W, N). Then it collects the extended safety level of each node that is within E hops in the East direction. Similarly, S, W, and N hops in the South, West, and North directions, respectively. Specifically, suppose the source is (0,0), the extended safety level of node $(+k, 0)$ (k-hop neighbor in the East direction) is $(*, *, *, N_{+k})$, where $*$ is a don't care (similarly, the extended safety level of node $(0, +k)$ (k-hop neighbor in the North direction) is $(E_{+k}, *, *, *)$), we have the following extension.

Theorem 1b: $(*, *, *, N_{+k})$ *is the extended safety level of node* $(+k, 0)$ *and* $(E_{+k}, *, *, *)$ *is the extended safety level of node* $(0, +k)$. *Minimal routing exists if*

- *the source node is safe; that is,* $x_d \leq E$ *and* $y_d \leq N$; *or*
- $x_d \leq E$ *and node* $(+k, 0)$ *with* $k \leq E$ *is safe with respect to* (x_d, y_d); *that is,* $y_d \leq N_{+k}$; *or*
- $y_d \leq N$ *and node* $(0, +k)$ *with* $k \leq N$ *is safe with respect to* (x_d, y_d); *that is,* $x_d \leq E_{+k}$.

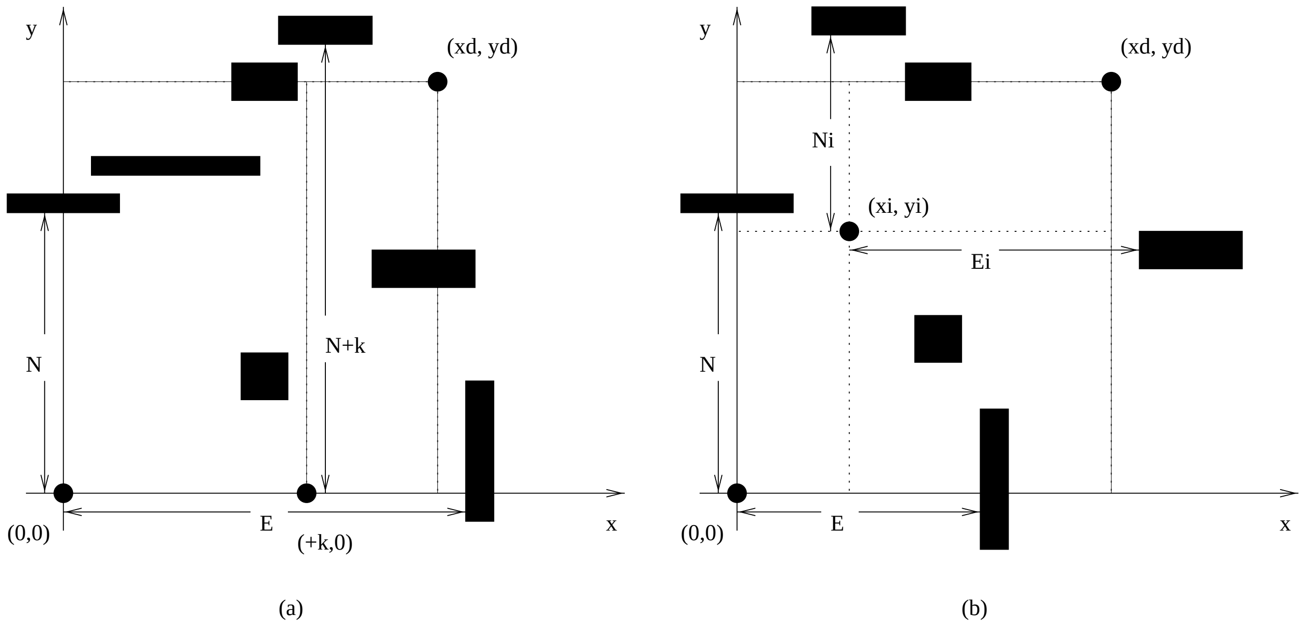

Figure 5: A two-phase routing process in 2-D meshes: (a) from $(0,0)$ to $(+k,0)$ and from $(+k,0)$ to (x_d,y_d). (b) from $(0,0)$ to (x_i,y_i) and from (x_i,y_i) to (x_d,y_d).

Again a two-phase routing process is used in extension 2: Wu's protocol is applied from $(0,0)$ to $(+k,0)$ (or $(0,+k)$) and, then, from $(+k,0)$ (or $(0,+k)$) to (x_d,y_d). Figure 5 (a) shows an example of extension 2. In this extension, additional information stored at each node is $x_d+y_d=O(n)$ in an $n \times n$ 2-D mesh.

Extension 3. Extension 2 works when the relevant section of one axis is clear of any faulty block. Extension 3 tries to cover cases when both sections intersect with faulty blocks.

Theorem 1c: *(E_i,S_i,W_i,N_i) is the extended safety level of node (x_i,y_i), where $0 \leq x_i \leq x_d$ and $0 \leq y_i \leq y_d$. Minimal routing exists if*

- *the source node is safe; that is, $x_d \leq E$ and $y_d \leq N$; or*
- *$(0,0)$ is safe with respect to (x_i,y_i) (i.e., $x_i \leq E$ and $y_i \leq N$) and (x_i,y_i) is safe with respect to (x_d,y_d) (i.e., $x_d-x_i \leq E_i$ and $y_d-y_i \leq N_i$).*

A two-phase routing process is again used in extension 3: Wu's protocol is applied from $(0,0)$ to (x_i,y_i) and, then, from (x_i,y_i) to (x_d,y_d) as shown in Figure 5 (b). In this extension, additional

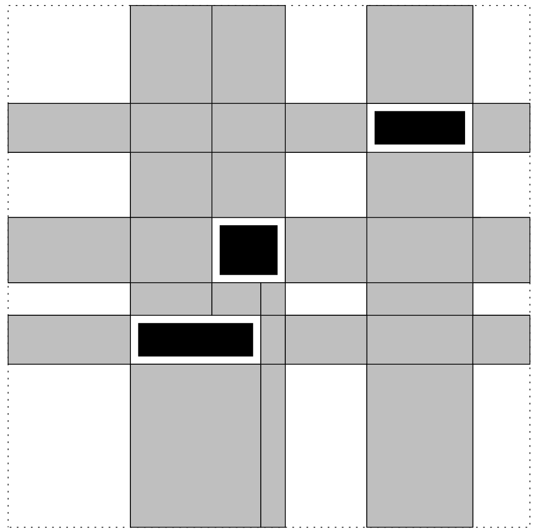

Figure 6: Distribution of faulty-block-information (lines) and extended-safety-level-information (shadowed regions).

information stored at each node is $x_d \cdot y_d = O(n^2)$. In the next section, we discuss some variations of this extension by selectively choosing some pivot nodes in region $[0 : x_d,\ 0 : y_d]$.

Again, all results apply to the MCC model. In order to distinguish two different models, extensions 1, 2, and 3 are labeled as extensions 1a, 2a, and 3a respectively under the MCC model.

4 Implementation Issues

Once faulty blocks are constructed, faulty-block-information and extended-safety-level-information need to be distributed. Faulty-block-information (two opposite corners of a faulty block) needs to be distributed to boundary lines of faulty blocks (see lines in Figure 6) and extended-safety-level-information (distance to the faulty block along each direction) needs to be distributed to only nodes between two parallel boundary lines of each faulty block (see shadowed regions in Figure 6). The same result also applies to the MCC model. Comparing with other fault information models such as a routing table associated with each node, the distribution and update process of such information is simple and converges quickly. Since the distribution is directional, this process is also free from oscillations occurring in a routing table.

The distribution of faulty-block-information has been discussed at Section 2. Here we focus on the distribution of extended-safety-level-information.

FORMATION-EXTENDED-SAFETY-LEVEL-INFORMATION

{At node u with (E, N, W, S) upon receiving an extended safety level (E', N', W', S')}

case sender

East neighbor: u's extended safety level is $(E' + 1, N, W, S)$
u forwards its extended safety level to its West neighbor (if any)

North neighbor: u's extended safety level is $(E, N' + 1, W, S)$
u forwards its extended safety level to its South neighbor (if any)

West neighbor: u's extended safety level is $(E, N, W' + 1, S)$
u forwards its extended safety level to its East neighbor (if any)

South neighbor: u's extended safety level is $(E, N, W, S' + 1)$
u forwards its extended safety level to its North neighbor (if any)

end case

Note that the default extended safety level is $(\infty, \infty, \infty, \infty)$. In the absence of faulty blocks, no information distribution is needed. Next, we consider information distribution in three extensions.

In extension 1, each node exchanges its extended safety level with its four neighbors. In extension 2, a row (and column) is called an *affected row* (and *affected column*) if the row (and column) intersects at least one faulty block. Note that nodes and only nodes on affected rows (and columns) need to collect extended safety level information. Boundary lines are not affected rows or columns. The number of boundary lines is always four times the number of faulty blocks. Nodes along each affected row (and affected column) exchange their extended safety levels. Note that each affected row (and affected column) is partitioned into several disjoint *regions* by faulty blocks and two edges of the 2-D mesh. Therefore, the exchange is within each region. A simple implementation of such an exchange starts from two ends of each region and pushes the partially accumulated information to the other end. Two partially accumulated information packets initiated from two ends form a complete packet (of extended safety level information within a region). In extension 3, one or more nodes, called *pivot nodes*, are selected to distribute its extended safety level information to all nodes in the 2-D mesh through broadcasting.

Several variations of extension 2 and extension 3 exist. In extension 2, to reduce the amount of information exchange, each region is further partitioned into several *segments*. One extended safety level from each segment is selected (typically the one with the highest safety level) to be passed around. The size of segments is adjustable. Another variation is to select up to four extended safety levels within each region (each one corresponds to the highest safety level along a particular direction within the region). In extension 3, the selection of pivot nodes can be done in a recursive way. For example, the center node of the 2-D mesh is first selected and, then, the mesh is partitioned into four submeshes by the pivot node. Each pivot node is selected from each of the four submeshes. This process continues until each submesh is sufficiently small. Clearly, the number of pivot nodes is $\sum_{i=1}^{k} 4^{i-1}$, where k is the level of partition. Another variation of extension 3 is to select pivot nodes in such a way that not only they are evenly distributed but also no two pivot nodes are on the same row or column.

We have the following analytical model on the expected number of affected rows (and columns) in an $n \times n$ 2-D mesh with k faults.

Theorem 2: *In an $n \times n$ 2-D mesh with k randomly generated faults. When k is relatively small with respect to n, the expected number x of affected rows (and affected columns) meets the following:*

$$\min_{x}\{|k - \sum_{i=1}^{x} \frac{n}{n-i+1}|\}.$$

In addition, this expected number remains the same under the faulty block and MCC models.

Proof: Let us call a case in which a fault falls into a row (column) that is *clean* (i.e., with no fault in the row (column) before the fault) a *hit* and the row (column) becomes *dirty*. The hits can be used to partition k faults into stages. The ith stage consists of the faults after the $(i-1)$th hit until the ith hit. The first stage consists of the first hit, since we are guaranteed to have a hit when all rows (columns) are clean. For each fault during the ith stage, there are $i-1$ rows that contain faults and $n-i+1$ rows that are clean. Thus, for each fault in the ith stage, the probability of a hit is $(n-i+1)/n$. Note that fault selection is not random with respect to a row (column). Because a dirty row (column) has fewer choices (among unselected columns (rows) within a row (column)) than a clean row (column), the probability of a hit (on a clean row or column) is slightly higher than $(n-i+1)/n$. Since k is relatively small with respect to n, the difference is negligible. Let n_i denote the number of faults in the ith stage. Thus, the number of stages x is

$$\min_{x}\{|k - \sum_{i=1}^{x} n_i|\}.$$

Each random variable n_i has a geometric distribution, and then,

$$E[n_i] = \frac{n}{n-i+1}.$$

By linearity of expectations

$$E[\sum_{i=1}^{x} n_i] = \sum_{i=1}^{x} E[n_i] = \sum_{i=1}^{x} \frac{n}{n-i+1}.$$

Since $E[k] = k$, we have the expected number of stages $E[x]$ meeting the following condition,

$$\min_{x}\{|k - \sum_{i=1}^{x} \frac{n}{n-i+1}|\}.$$

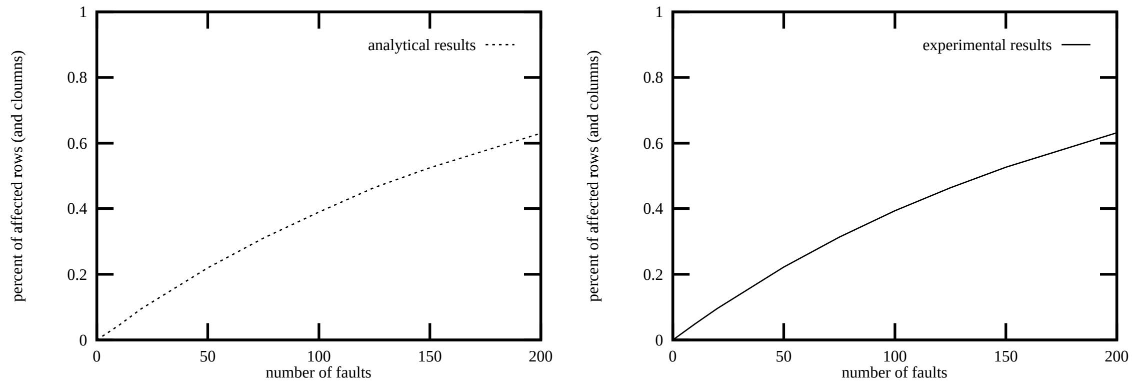

Figure 7: The expected percentage of affected rows (and columns) under the analytical model and the simulated model for $n = 200$.

Based on the definitions of disabled node under the faulty block and MCC models. A node is labeled disabled if there are two or more disabled or faulty neighbors in different dimensions. That is, a disabled node will not generate a new hit on row or column. ■

Figure 7 shows the expected number of affected rows (and columns) under the analytical model and the simulated model for $n = 200$. The results for both analysis and experiment are very close, even when the number of faults reaches 200. Again, this is not surprising, when the number of dirty rows (that cause hits) is 60% of 200 (i.e., 120), each dirty row on the average has less than 2 hits. That is, when a new fault is selected, a dirty row has over 198 free column slots while a clean row has 200 free column slots. The difference is less than 1%. From Figure 7, we can see that about 20% percent of rows (columns) are affected when the number of faults reaches 50, 40% percent when the number of faults reaches 100, and 60% percent when the number of faults reaches 200. This confirms the effectiveness of limited global distribution of fault information.

5 Simulation Results

A routing process using either the sufficient safe condition or any of its three extensions can decide if there is a minimal/sub-minimal path at the source. A simulation has been conducted to test the effectiveness of such a safe condition and its three extensions. We use a 200×200 mesh with randomly generated faults. The source is at the center of the mesh and is the origin of the coordinator. The coordinator divides the mesh into four 100×100 quadrants. Without the loss of generality, we assume that the destination is always located at the first quadrant.

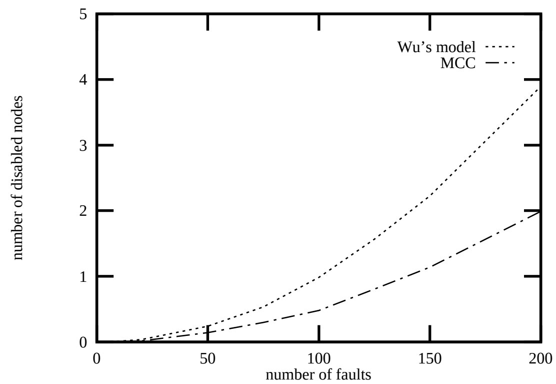

Figure 8: Average number of disabled nodes in a faulty block.

After the occurrence of faults (which are randomly generated), faulty blocks and MCCs are constructed, faulty-block-information and extended-safety-level-information are distributed. And then, we randomly pick a destination in the first quadrant (in the 100×100 submesh) and show the percentage of the existence of a minimal/sub-minimal path ensured at the source. To simplify the simulation, we assume that the source and destination are outside of any faulty block and the number of faults in the whole network is no more than 200. Figure 8 shows the average number of disabled nodes in faulty blocks under the regular faulty block model and the MCC model. Although the MCC model generates fewer disabled nodes than the faulty block does in terms of percentage, the actual number of disabled nodes generated are both very small. The reason is that in our simulation, faults are randomly generated and are scattered in the 200×200 2-D space, the probability of generating large faulty blocks is small. Therefore, in the following simulations the difference between the MCC model and the faulty block model is insignificant in terms of the percentage of the existence of a minimal/sub-minimal path.

Figures 9 (a) and (b) show the percentages of the existence of a minimal path ensured by extension 1 and extension 1a and compare them with the ones ensured by the sufficient safe condition under the regular faulty block model and under the MCC model (safe source in the figure). If the number of faults in the whole network is no more than 30, most of routing processes (90% by the sufficient safe condition and 99% by extension 1) can ensure a minimal path at their sources. Existence of a minimal path is based on Wang's necessary and sufficient condition. The percentage of the existence of a minimal path stays very high (close to 1) under both models even when the number of faults reaches 200. Figure 9 also shows the probabilities of a safe neighbor for the given source; that is, the corresponding routing process ensures at least a sub-minimal path (extension 1 (sub-min) and extension 1a (sub-min) in Figure 9) at the source. These percentages also indicate the feasibility of applying the minimal/sub-minimal routing.

Figures 10 (a) and (b) show the percentages of the existence of a minimal path ensured by variations

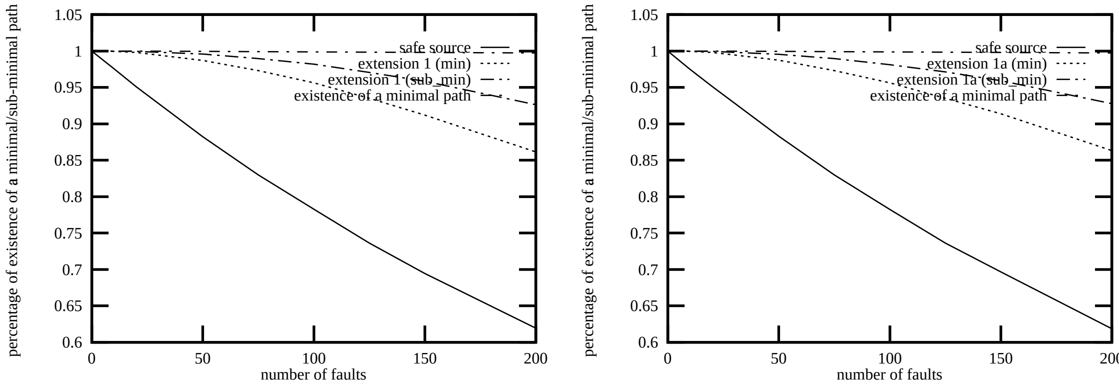

Figure 9: Percentage of a minimal/sub-minimal path ensured at the source by the sufficient safe condition and extension 1 (a) and extension 1a (b).

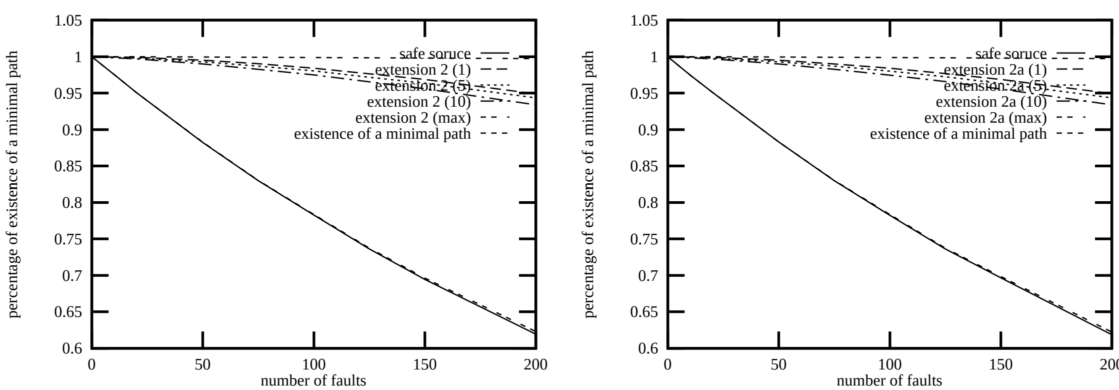

Figure 10: Percentage of a minimal path ensured at the source by the sufficient safe condition and variations of extension 2 (a) and extension 2a (b).

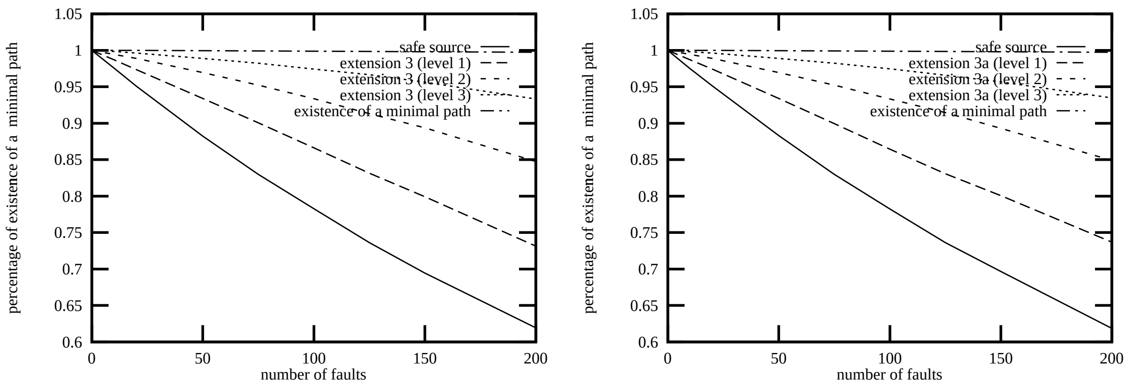

Figure 11: Percentage of a minimal path ensured at the source by the sufficient safe condition and variations of extension 3 (a) and extension 3a (b).

of extension 2 and extension 2a with different sizes of disjoint segments and compare them with the one ensured by the sufficient safe condition under both fault models. When the source has extended-safety-level-information of all the nodes along X and Y dimensions in the same region, most of routing processes ($\geq 94\%$) can ensure a minimal path at their sources. As the size of segments increases, the number of segments in a region is reduced (i.e., fewer samples are collected). Since only one extended safety level from each segment is selected, less safety level information can be used and the percentage goes down. Note that the percentage (of the existence of a minimal path) is relatively insensitive to the size of segments (especially when the size of segments is relatively small between 1 and 10). However, when only one segment is used in each region, segment information is not useful for most cases, either the selected node fails the sufficient safe condition or the selected node is outside the region $[0 : x_d, 0 : y_d]$. The percentage ensured in such a case is close to the one ensured by the sufficient safe condition only. (Both extension 2 and extension 2a contain at least the extended-safety-level-information of the source.)

Figures 11 (a) and (b) show the percentages of the existence of a minimal path ensured by variations of extension 3 and extension 3a with different levels of partition and compare them with the one ensured by the sufficient safe condition under both fault models. When the center node of the submesh (in the first quadrant) is selected in the first level, the source of a routing has the extended-safety-level-information from that pivot node. Thus, the routing process can find more minimal paths than the one based on the sufficient safe condition only. As the number of levels increases, the more pivot nodes are selected, and more safety-level-information is collected at source. Therefore, the percentage of the existence of a minimal path (if it exists) increases. Note that such a percentage changes significantly whenever an extra level is added.

Figures 12 (a) and (b) show the percentages of the existence of a minimal path ensured by strategies

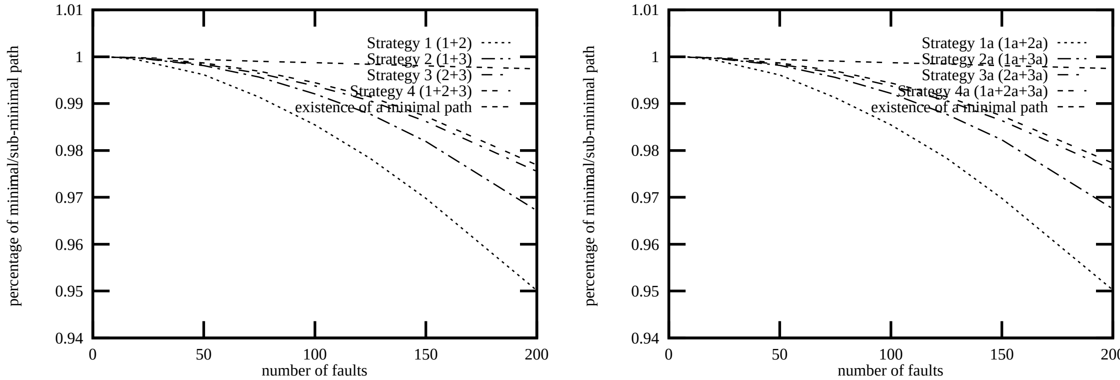

Figure 12: Percentage of a minimal path ensured at the source by combinations of different extensions: (a) Strategies 1, 2, 3 and 4. (b) Strategies 1a, 2a, 3a and 4a.

generated from different combinations of extensions. In routing strategy 1, extension 1 is first applied. If it cannot ensure a minimal path, extension 2 is then applied. Again, strategy 1a means strategy 1 under the MCC model and the same convention applies to other strategies. Here, the size of a segment in each region is set to 5. Figures 12 (a) and (b) show that most of cases (> 95%) have a minimal path by using routing strategy 1. In routing strategy 2, extension 3 is applied after extension 1 (if needed). Such a routing strategy can ensure more minimal paths (> 97%) than routing strategy 1. Here each pivot node in extension 3 is selected randomly in a submesh and the partition level is set to 3. Clearly, the number of pivot nodes is equal to 1 (at level 1) + 4 (at level 2) + 16 (at level 3) = 21. In routing strategy 3, extension 2 (with a segment size of 5) is applied first; and then, extension 3 (with a partition level of 3). In routing strategy 4, all the extensions are applied in the order of extensions 1, 2, and 3. Among all these four routing strategies, routing strategy 4 has the maximum percentage of the existence of a minimal path ensured at the source. The result from strategy 3 stays relatively close to the one from strategy 4. In Figure 12, the difference in percentages between any two strategies indicates the effectiveness of different combinations of extensions. Again, all strategies can be applied to the MCC model. In order to distinguish two models, strategies 1, 2, 3, and 4 are labeled as strategies 1a, 2a, 3a, and 4a, respectively under the MCC model.

From the above simulation, we conclude that in a 2-D mesh in which the number of faults is usually low, the sufficient safe condition and its extensions can ensure a minimal/sub-minimal path for most cases, especially extension 2 and extension 3, although the sufficient safe condition and extension 1 can be implemented much easily. The percentage of a minimal/sub-minimal path does not go down as much as the number of faults increases. Extension 2 and extension 1 each can ensure more minimal paths than the sufficient safe condition. Extension 3 can ensure more minimal paths than extension 2 (and extension 1). As the number of segments inside one region increases, extension 2 can ensure more minimal paths. However, the difference is not as significant as one when we increase the level of partition in extension 3. When the sufficient safe condition and its extensions are all applied, the corresponding routing strategy can ensure a minimal path for over 97.5% cases as long as the number of faults stays within 200 in a 200×200 2-D mesh. Again, we can conclude the similar results under the MCC model.

6 Conclusions

In this paper, we have given three extensions to a sufficient condition for the existence of a minimal path in a 2-D mesh with faulty blocks. Extensions have also been applied to 2-D meshes with the MCC model which is based on the faulty block model by activating some disabled (but healthy) nodes. Simulation results have confirmed the effectiveness of these extensions and their combinations. Results also show that the percentages of minimal routing under extensions are very close to the optimal case with global information. Our future work will focus on trade-offs between cost and effectiveness. Possible extensions to 3-D meshes and other high-dimensional mesh networks will be another focus.

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

[5] S. Dutt and J. P. Hayes. Some practical issues in the design of fault-tolerant multiprocessors. *IEEE Transactions on Computers.* 41, (5), May 1992, 588-598.

[6] 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.

[7] 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.

[8] S. P. Kim and T. Han. Fault-tolerant wormhole routing in meshes with overlapped solid fault regions. *Parallel Computing.* 23, 1997, 1937-1962.

[9] D. T. Lee, C. D. Yang, and C. K. Wong. Rectilinear paths among rectilinear obstacles. *Discrete Applied Mathematics.* 70, 1996, 185-215.

[10] S. L. Lillevik. The Touchstone 30 gigaflop DELTA prototype. *Proc. of the 6th Distributed Memory Computing Conference.* 1991, 671-677.

[11] C. L. Seitz. The architecture and programming of the Ametek Series 2010 multicomputer. *Proc. of the 3rd Conference on Hypercube Concurrent Computers and Applications.* Jan. 1988, I, 33 - 36.

[12] J. D. Shih. Adaptive fault-tolerant wormhole routing algorithms for hypercube and mesh interconnection networks. *Proc. of the 11th International Parallel Processing Symposium.* 1997, 333-340.

[13] P. H. Sui and S. D. Wang. Adaptive fault-tolerant deadlock-free routing in meshes and hypercubes. *IEEE Transactions on Parallel and Distributed Systems.* 11, 2000, 50-63.

[14] N. F. Tzeng and G. Lin. Maximum reconfiguration of 2-D mesh systems with faults. *Proc. of the 1996 International Conference on Parallel Processing.* 1996, I, 77-84.

[15] D. Wang. Minimal-connected-component (MCC) - a refined fault block model for fault-tolerant minimal routing in mesh. *Proc. of IASTED Int'l Conf. on Parallel and Distributed Computing and Systems.* Nov. 1999, 95-100.

[16] 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.

[17] 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.

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

[19] J. Zhou and F. Lau. Adaptive fault-tolerant wormhole routing in 2D meshes. *Proc. of the 15th International Parallel & Distributed Processing Symposium (IPDPS 2001).* 2001, 56.