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

Jie Wu and Zhen Jiang
Department of Computer Science and Engineering
Florida Atlantic University
Boca Raton, FL 33431

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. Fault information is coded in a 4-tuple called *extended safety level* associated with each node 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) fault model are also considered. Simulation results show substantial improvement in terms of higher percentage of minimal routing in 2-D meshes under both fault models.

1 Introduction

The performance of 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. We study fault-tolerant routing in the 2-dimensional (2-D) mesh and focus on achieving fault tolerance using the inherent redundancy present in 2-D meshes, without adding spare nodes and/or links. To this end, *fault model* and *information model* are the two keys to successfully extend the existing approaches.

Many studies have been done on routing in 2-D meshes based on the faulty block model [1, 2, 3]. 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 [6], 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. This work is based on an early work on *safety-level-based* routing [5]. Unlike many existing information models that require each node to have knowledge of the entire network, extended safety level as coded fault information associated with each node represents limited global information by exploring locality of disturbances in the network. 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 is also distributed to nodes along four boundary lines of each faulty block.

However, the sufficient safe condition in [6] is overly 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) fault model [4] are also considered. MMCs are rectilinear-monotone polygonal fault blocks of the polygonal shape 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.

2 Preliminaries

2-D meshes. Each node u in an $n \times n$ 2-D mesh has an address (x_u, y_u), where $0 \le x_u, y_u < n$. 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 and their dis-

Figure 1. (a) A faulty block consisting of faulty and disabled nodes. The corresponding type-one MCC (b) and type-two MCC (c).

Figure 2. (a) Constructing minimal paths from the destination. (b) Distribution of faulty-block-information (lines) and extended-safety-level-information (shadowed regions).

tance, $D(u,v)$, is equal to $\mid x_u - x_v \mid + \mid y_u - y_v \mid$. 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 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. A node-labeling scheme that identifies nodes is defined as follows: 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 nodes. 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 (black nodes) form a rectangle [2:6, 3:6].

Minimal-connected-components (MCCs). MCCs fault model [4] was proposed to reduce the size of a faulty block by "removing" four "corner sections" of the block based on 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 of a faulty block are removed and the resultant block is called *type-one MCC*; otherwise, the SW and NE corner sections are removed and the resultant block is called *type-two MCC*.

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 and East neighbors or South and West neighbors are faulty or disabled, u is labeled *disabled*. Connected faulty and disabled nodes form a type-one MCC. Type-two MCC can be defined in a similar way by exchanging the role of N (North), E, S, W by E, S, W, N, respectively. Figure 1 (b)

and (c) show two sample MCCs.

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. In Figures 1 (b) and (c), the status of node (4,3) is (fault-free, fault-free) and one for (2,6) is (fault-free, disabled). It is assumed that source and destination nodes both have (fault-free, fault-free) status.

Extended safety level. The extended safety level [6] 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 closet faulty block to its East. S, W, and N are defined in a similar way. To ensure a minimal path from source node, Wu [6] provided the following safe node definition: 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 [6]: *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 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. Wu [6] 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 (see Figure 2 (a)).

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
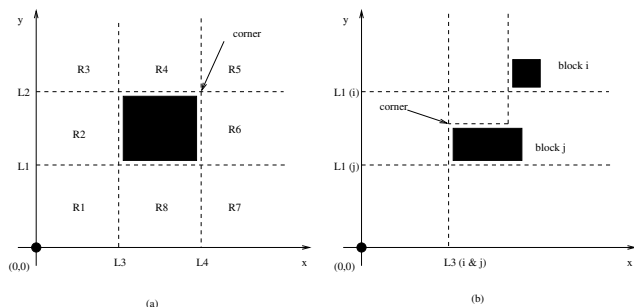
2

Figure 3. (a) Boundaries of a faulty block. (b) Boundaries of multiple faulty blocks.

corresponding result are still applicable to 2-D meshes with MCCs.

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 example of Figure 3 (a), source (0,0) is safe with respect to a destination in region R_4. A minimal route may not be guaranteed even if a preferred neighbor is selected at each hop towards the destination. For example, once the packet enters region R_8, no more minimal route is possible (all such routes are blocked by the faulty block).

In [6], 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 (see in Figure 2 (b)). 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 [6], 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 uses any adaptive minimal routing until the boundary of any faulty block is met. If the selec-

tion 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 based on the relative location of the destination.

WU'S PROTOCOL [6]:
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

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. 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. Faulty-block-information used in minimal routing is the same under the MCC model.

Necessary and sufficient conditions. Wang [4] 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. However, global information of fault distribution is needed to apply this necessary and sufficient condition.

3

3 Extended Sufficient Conditions

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 $x_d, y_d \geq 0$, we have the following three extensions.

Extension 1. The sufficient safe condition is enhanced by taking into consideration of neighbors' (preferred or spare) safety status. *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 constant (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.*

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. 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 (1) the source node is safe; that is, $x_d \leq E$ and $y_d \leq N$; (2) $x_d \leq E$ and node $(+k, 0)$ with $k \leq E$ is safe with respect to (x_d, y_d); that is, $y_d \leq N_{+k}$; (3) $y_d \leq N$ and node $(0, +k)$ with $k \leq N$ is safe with respect to (x_d, y_d); that is, $x_d \leq E_{+k}$.*

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 4 (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.
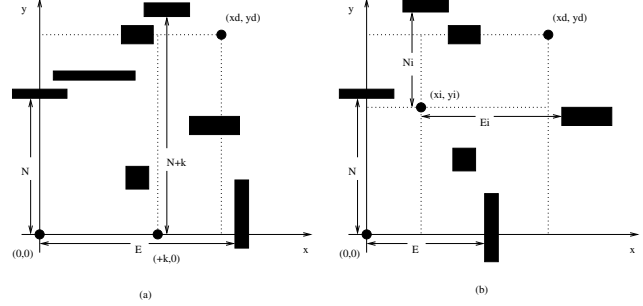

Figure 4. A two-phase routing process in 2-D meshes: (a) from $(0, 0)$ **to** $(+k, 0)$ **and from** $(+k, 0)$ **to** (x_d, y_d)**. (b) from** $(0, 0)$ **to** (x_i, y_i) **and from** (x_i, y_i) **to** (x_d, y_d)**.**

Theorem 1c: (E_i, S_i, W_i, N_i) *is the extended safety level of node* (x_i, y_j)*, where* $0 \leq x_i \leq x_d$ *and* $0 \leq y_i \leq y_d$. *Minimal routing exists if (a) the source node is safe; that is, $x_d \leq E$ and $y_d \leq N$; or (b) $(0, 0)$ 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 4 (b). In this extension, additional 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 where extensions 1, 2, and 3 are labeled as extensions 1a, 2a, and 3a, respectively.

4 Implementation Issues

Faulty-block-information (two opposite corners of a faulty block) needs to be distributed to boundary lines of faulty blocks (see lines in Figure 2 (b)) 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 2 (b)). The same result also applies to the MCC model.

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. Boundary lines are not affected rows or columns. The number of boundary lines is always four times the number of faulty blocks. 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. In extensions 3, one or more

4

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.

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 $n-i+1$ clean rows. Thus, for each fault in the ith stage, 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 (row) 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 $E[x]$ meet the following condition, $\min_x \{|k - \sum_{i=1}^{x} \frac{n}{n-i+1}|\}$.

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
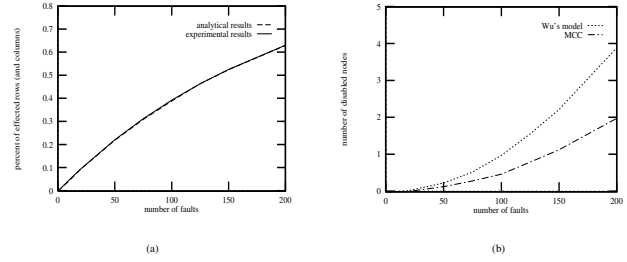

Figure 5. (a) The percentage of affected rows (and columns) under the analytical and simulated models for $n = 200$. (b) Average number of disabled nodes in a faulty block.

new hit on row or column. ∎

Figure 5 (a) shows the expected number of affected rows (and columns) under the analytical and simulated models for $n = 200$. The results for both analytical and experimental 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 5 (a), 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 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. 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. Figure 5(b) shows the average number of disabled nodes in a faulty block 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. Therefore, in the simulations (shown in Figure 6) the difference between the MCC model and the faulty block model is insignificant in terms of percentage of the existence of a minimal/sub-minimal path.

From the results of 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 exten-
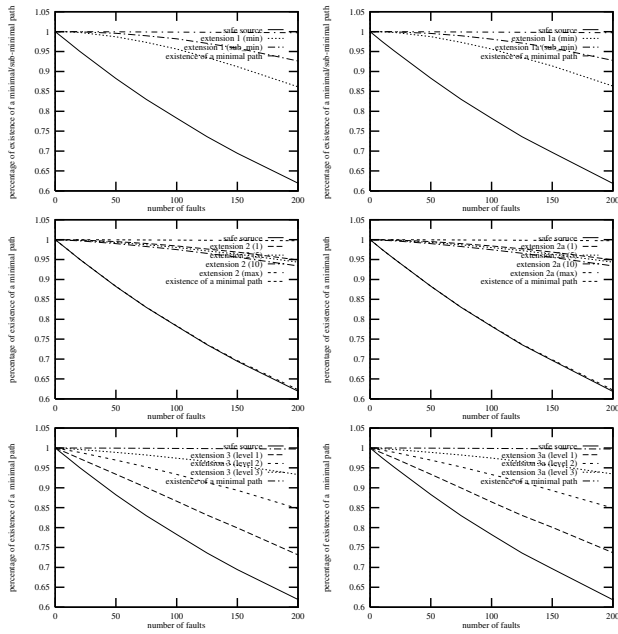
5

Figure 6. Percentage of a minimal/sub-minimal path ensured at the source by the sufficient safe condition and extensions 1 and 1a (first row), extensions 2 and 2a (second row), and extensions 3 and 3a (third row).
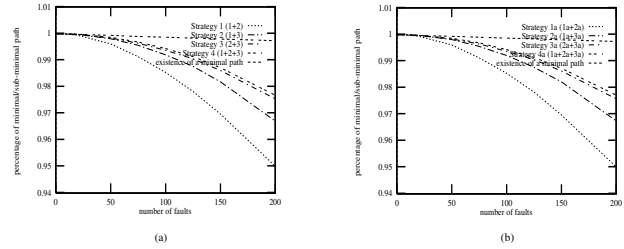

Figure 7. Percentage of a minimal path ensured at the source by combinations of different extensions: (a) Strategies 1, 2, 3 and 4. (b) Strategies 1a, 2a, 3a and 4a.

sion 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. Again, we can conclude the similar results under the MCC model.

Figures 7 (a) and (b) show the percentages of the existence of a minimal path ensured by strategies generated from different combinations of extensions. In routing strategy 1, $(1 + 2)$ means that extension 1 is first applied, and then, extension 2 is applied if extension 1 cannot ensure a minimal path. Other strategies we defined in a similar way. 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 7, the difference in percentages between any two strategies indicate 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.

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] 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.

[4] 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.

[5] 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.

[6] 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.