2007

On Achieving the Shortest-Path Routing in 2-D Meshes

Zhen Jiang
West Chester University of Pennsylvania, zjiang@wcupa.edu

Jie Wu
Florida Atlantic University

On Achieving the Shortest-Path Routing in 2-D Meshes *

Zhen Jiang
Dept. of Computer Sci.
West Chester University
West Chester, PA 19383
Email: zjiang@wcupa.edu

Jie Wu
Dept. of Computer Sci. and Eng.
Florida Atlantic University
Boca Raton, FL 33431
Email: jie@cse.fau.edu

Abstract

In this paper, we present a fully distributed process to collect and distribute the minimal connected component (MCC) fault information so that the shortest-path between the source and the destination can always be found in the corresponding information-based routing via routing decisions at each intermediate node. Considering the communication cost in the above information distribution, a more practical implementation is provided with only a low number of nodes along the boundary lines involved in the information propagation. The experimental results show the substantial improvement of our approach in terms of the success rate in finding the shortest-path and the average path length.

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. The routing time of packets is one of the key factors 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 routing path from the source to the destination is determined by the forwarding node selection at each intermediate node in a fully-distributed manner to make entire system more scalable. It is necessary to present a routing scheme using multiple-phase localized decisions that always route the package to the destination via the shortest-path, so that the destination can be reached in the quickest way.

The *mesh-connected topology* is one of the most thoroughly investigated network topologies for multicomputer systems. 2-dimensional (2-D) meshes are lower dimensional meshes that have been commonly discussed due to structural regularity for easy construction and high potential of legibility of various algorithms. Some multicomputers were built based on the 2-D meshes [3, 6, 7]. As the number of nodes in a mesh-connected multicomputer system 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 [2, 4, 9, 10].

The shortest-path is constructed among all the non-faulty nodes under the existing network configuration after the failure. Obviously, in a multiple-phase routing, the shortest-path in each phase does not imply the entire path is the shortest in presence of node faults (link faults can be treated as node faults by disabling the corresponding adjacent nodes). Appropriate fault information provided for routing decisions is the key to achieving the shortest-path routing. Most existing literature uses the simplest orthogonal convex region in the information model. To reduce the number of non-faulty nodes contained in rectangular faulty blocks, Wang [8] proposed the minimal connected component (MCC) model as a refinement of the rectangular faulty block model by considering the relative locations of source and destination nodes. The original idea is that a node will be included in an MCC only if using it in a routing will definitely make the route non-shortest. It turns out that each MCC is of the rectilinear-monotone-polygonal shape and is the absolutely minimal fault region in 2-D meshes. In [5], the information of each MCC is propagated along an edge of its "forbidden region", also called boundary. For each routing case, a feasibility check is conducted first at the source. It sends out two detection messages and waits for their responses to ensure the existence of a path with the Manhattan distance [1]. Then, the routing process starts if

*The work was supported in part by NSF grants ANI 0083836, CCR 9900646, CNS 0422762, CNS 0434533, and EIA 0130806.

1-4244-0910-1/07/$20.00

and only if such a path exists. The information saved along the boundary will be used in the routing decisions at those nodes to guarantee the success of a shortest-path routing. Such a routing that always routes the message along a Manhattan distance path is also called Manhattan routing.

In our new approach, the MCC information will propagate not only along the boundary, but also into the forbidden region. Therefore, the information can be used to avoid detour in those cases that do not have the Manhattan distance path. A routing is proposed under this new information model to form the shortest-path. Note that such a path is the Manhattan distance path if one exists. Considering the cost of broadcasting in forbidden region, a practical implementation is proposed here. Two boundaries initialized from opposite corners of each MCC, instead of only one in [5], are used to bound each forbidden region. The information is only needed to send to a limited number of nodes along those boundaries. Our simulation results show that the shortest-path can be achieved in most cases and only a very low number of detours are needed in those non-shortest-path routings.

The contribution of this paper is threefold. First, we focus on a way to collect and distribute the MCC information by information exchanges among neighbors so that the routing protocol based on such information can always form the shortest-path, even when the Manhattan distance path does not exist. It is noted that this new information model can be applied to any adaptive routing. This paper will prove that there is no path shorter than the one found under the MCC model. Second, to reduce the cost of information distribution, we extend the boundary model of 2-D meshes in [5]. The information only propagates to a limited number of nodes along the boundaries while keeping the length of routing path as short as possible. Third, we develop a simulation to show substantial improvement of our new approach in terms of the success rate of the shortest-path routing, the average length of routing paths, and the number of nodes involved in the information distribution.

2 Preliminary

A 2-dimensional (2-D) $n \times n$ mesh with n^2 nodes has an interior node degree of 4. Nodes along each dimension are connected as a linear array. Each node u has an address (x_u, y_u), where $0 \leq x_u, y_u < n$. Node (x_u+1, y_u) is called the $+X$ neighbor of u. Respectively, (x_u-1, y_u), (x_u, y_u-1), and (x_u, y_u+1) are $-X$, $-Y$ and $+Y$ neighbors of node u. The Manhattan distance between any two nodes u and v is the geographic distance $\mid x_u - x_v \mid + \mid y_u - y_v \mid$, denoted by $M(u, v)$. Assume node s is the source node, u is the current node, and d is the destination node. We consider the positions of the source and destination when the new faulty components are constructed. Without loss of generality, assume $x_s = y_s = 0$ and $x_d, y_d \geq 0$. Due to the effect of faulty nodes, a path with the length $M(s, d)$ may not exist. $D(s, d)$ denotes the length of a shortest-path between s and d and $D(s, d) \geq M(s, d)$. In general, $[x : x', y : y']$ represents a rectangular region with four vertexes: (x, y), (x, y'), (x', y'), and (x', y). Specifically, $[x : x, y : y']/[x : x', y : y]$ represents a line segment along the Y/X dimension.

The formation of MCC in 2-D meshes [8] is based on the notions of *useless* and *can't-reach* nodes: A useless node is such a node that once a routing enters it, the next move must take a $-X/-Y$ direction, making the routing non-shortest. A can't-reach node is such a node that for a routing to enter it, a $-X/-Y$ direction move must be taken, making the routing non-shortest. The node status faulty, useless, and can't-reach can be determined through a labeling procedure: Initially, label all faulty nodes as *faulty* and all non-faulty nodes as *safe*. If a node is safe, but its $+X$ neighbor and $+Y$ neighbor are faulty or useless, it is labeled *useless*. If the $-X$ neighbor and $-Y$ neighbor are faulty or can't-reach, such a safe node is labeled *can't-reach*. The nodes are iteratively labeled until there is no new useless or can't-reach node. All faulty, useless, and can't-reach nodes are also called *unsafe* nodes. The other nodes are called *safe nodes*. Neighboring unsafe nodes form an MCC. The labeling procedure can quickly identify the non-faulty nodes in MCCs. Each active node collects its neighbors' status and updates its status. Only those affected nodes update their status. Figure 1 (a) shows the idea of the definition of useless and can't-reach nodes. Figure 1 (b) shows a sample of MCC under the assumption that $x_s = y_s = 0$ and $x_d, y_d \geq 0$. In this paper, we focus on the case when the shortest-path exists. Therefore, we have the following assumption: (a) the entire network is connected, and (b) the source and the destination are safe nodes.

After the labeling process, a distributed process presented in [5] is conducted to collect the shape information of each MCC and distribute it to a limited number of nodes, also called the boundaries. This process starts from an *initialization corner*. The initialization corner is a safe node with two edge neighbors of the same MCC in the $+X$ and $+Y$ dimensions. Any safe node with an unsafe neighbor is called an *edge* node of the corresponding MCC. A safe node with two edge neighbors of the same MCC in the $-X$ and $-Y$ dimensions is called the *opposite corner*. From that initialization corner, two identification messages, one clockwise and one counter-clockwise, each carrying partial region information, will propagate along the edges of MCC and reach its opposite corner. By collecting the location information of each node these two messages passed through, the shape of this MCC can be identified at the opposite corner (see in Figure 1 (b)). After that, the propagation will continue and bring the identified shape information $F(c)$

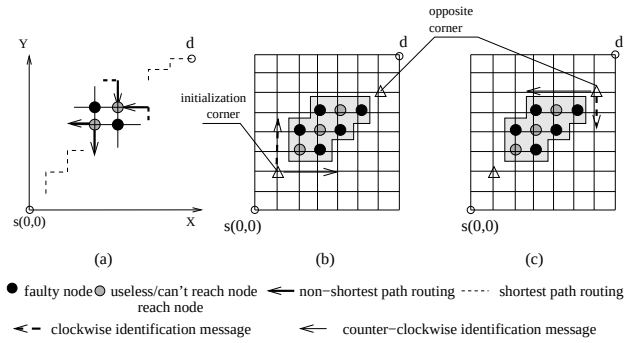

Figure 1. (a) Defintion of useless and can't-reach nodes. (b) A sample of MCC and its propagation activated at the initialization corner. (c) Identified information re-sending.

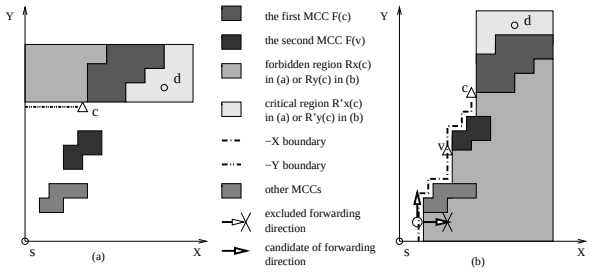

Figure 2. (a) Sample of MCC information propagation. (b) Sample of information propagation with intersection and the routing decision using the corresponding propagation information.

back to the initialization corner (see in Figure 1 (c)). The region right below $F(c)$ is called the *forbidden region*, noted by $R_Y(c)$. The region right above $F(c)$ is called *critical region*, noted by $R'_Y(c)$. To guide the routing process, one boundary message (also called $-X$ boundary) will carry the information $F(c)$, $R_Y(c)$, and $R'_Y(c)$ and propagate to all the nodes along the line $x = x_c$ until it reaches the edge of mesh. When this boundary line intersects with another MCC $F(v)$, a right turn is made. After that, it will go along the edges of $F(v)$ to join the same boundary of $F(v)$. From the corner v, $R_Y(v)$ merges into $R_Y(c)$ ($R_Y(c) = R_Y(c) \cup R_Y(v)$) (see in Figure 2 (b)). Similarly, another boundary propagation (construction of $-Y$ boundary) carrying $F(c)$, $R_X(c)$, and $R'_X(c)$ goes along $y = y_c$ (see in Figure 2 (a)) and will make a left turn if necessary. The whole procedure is shown in Algorithm 1.

Algorithm 1 [5]: Boundary construction of an MCC $F(c)$ ($B1$).

1. From the initialization corner c, two identification message (one clockwise and one counter-clockwise) are sent along the edges until they reach the opposite corner c'. The locations of all intermediate corners are collected to form the shape $F(c)$ at node c'. Then, the forbidden and critical regions ($R_X(c)$, $R_Y(c)$, $R'_X(c)$, $R'_Y(c)$) are identified.
2. From node c', the propagation will continue until the identified information reaches back to node c.
3. From node c, the triple ($F(c)$, $R_X(c)$, $R'_X(c)$)/($F(c)$, $R_Y(c)$, $R'_Y(c)$) is propagated along line $y = y_c$ / $x = x_c$. When the propagation intersects with another MCC, say $F(v)$, it will make a left/right turn to join the same boundary of $F(v)$. Since then, $R_X(v)$ / $R_Y(v)$ will merge into $R_X(c)$ / $R_Y(c)$.

Before the routing starts, a detection is activated at the source node s to ensure the existence of a Manhattan distance path. After that, the routing decision at each node along the path, including the source node s, basically has two forwarding directions: $+X$ and $+Y$ directions. The information saved along the boundaries helps to avoid the routing entering the forbidden region, by excluding the corresponding direction from the candidates of the forwarding direction. Then, any fully adaptive routing process could be applied to forward the message. The procedure of routing decision using boundary information is listed in Algorithm 2 and can be seen in the sample in Figure 2 (b).

Algorithm 2 [5]: Manhattan routing at the current node u with the information of $d(x_d, y_d)$

1. Add $+X$/$+Y$ direction into the set of candidates of forwarding directions P, if $x_d > 0$/$y_d > 0$ and the $+X$/$+Y$ neighbor is not fault.
2. For each triple ($F(c)$, $R(c)$, $R'(c)$) found at u, exclude the direction from P if using it will cause the routing enter region $R(c)$ while $d \in R'(c)$.
3. Apply any fully adaptive routing process to select a forwarding direction from set P.
4. Forward the routing message along the selected direction.

The more faults occur in the networks, the more and bigger MCCs will form. As a result, we have more routing cases that are blocked by MCCs and do not have the Manhattan distance path, i.e., $D(s,d) > M(s,d)$. The above routing process cannot find the existing connected path due to the failure in feasibility check. Therefore, the problem is how to find the shortest-path when $D(s,d) > M(s,d)$.

In E-cube routing [2], when the forwarding direction is blocked, it will detour around the fault region to reach the other side (see in Figure 3 (a)). With this kind of detours, the routing in Algorithm 2 can find a path from s to d when $D(s,d) > M(s,d)$. Therefore, the feasibility check is unnecessary. The routing process is rewritten in Algorithm 3.

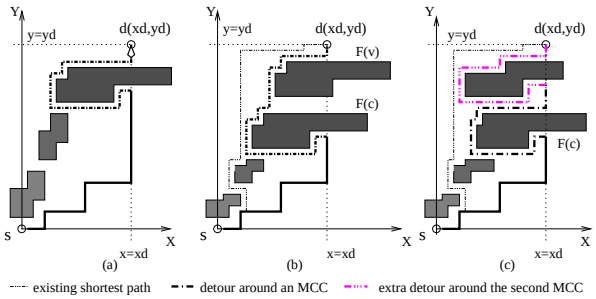

Figure 3. (a) Detour in E-cube routing [2]. (b) Detour of routing in Algorithm 3 when it is applied to the case $D(s,d) > M(s,d)$. (c) Extra detour.

When such a routing intersects with an MCC, say $F(c)$, it will route around the fault region in the clockwise direction. In this way, the joint boundary of the second MCC $F(v)$ can be used to save detours of $F(v)$ (see in Figure 3 (b)). However, this routing is not shortest-path routing. Moreover, when the whole detour path around $F(c)$ is inside the forbidden region of another MCC, the routing can be blocked again and need some extra detours (see in Figure 3 (c)). In the following section, we consider all the cases and proposed a new information model to help the routing achieve the shortest-path.

Algorithm 3: Routing in Algorithm 2 with E-cube routing detour ($RB1$), at the current node u with the information of $d(x_d, y_d)$

1. Add $+X/+Y$ direction into the set of candidates of forwarding directions P, if $x_d > 0/y_d > 0$ and the $+X/+Y$ neighbor is neither fault nor the preceding node.
2. Same as step 2 in Algorithm 2.
3. If $P = \phi$ (the routing is blocked by an MCC), select $-X$ or $-Y$ direction to route around the MCC in clockwise direction and go to step 5.
4. Same as step 3 in Algorithm 2.
5. Same as step 4 in Algorithm 2.

---

3 Fault Information Model for the Shortest-Path Routing

In this section, our method on achieving the shortest-path routing in 2-D meshes in presence of faults is introduced. We will prove that no path is shorter than the one found in our routing. Considering of the communication cost in the information distribution in the above method, its practical implementation is provided. We also analyze the performance of our information based routing under such an extension model.

In [5], it has been proved that the routing is blocked under MCC model if and only if the source is inside the forbidden region of one MCC while the destination is inside its critical region (see in Figure 4 (a)). To simplify the discussion, we focus on the situation when the Manhattan routing is blocked in the $+Y$ direction, i.e., $s \in R_Y \wedge d \in R'_Y$. For the remaining situation, the results can be obtained by simply rotating the mesh. As shown in the sample in Figure 4 (a), had the detours along the $-X$ direction been made to node v, a shortest-path would be found in Algorithm 3. The problem of routing for not making such detours is the lack of MCC information in the routing decision at the nodes inside forbidden region. In the new information model, the MCC information will broadcast to all the nodes inside the forbidden region so that the shortest-path can always be found.

Simply, for a certain MCC, $F(c)$, besides the $-X$ boundary initialized at the corner c, the $+X$ boundary is initialized at the opposite corner c' and propagated along line $y = y_{c'}$. If necessary, it will make a left turn to join the same $+X$ boundary of another MCC $F(v)$ at the corresponding opposite corner v'. After the joint point, the forbidden region of $F(v)$ will merge into that of $F(c)$. Figure 4 (b) show an example how the $+X$ boundary of $F(c)$ joins that of $F(c_2)$ at node c'_2. The area between these two boundaries is defined as the forbidden region of $F(c)$, denoted by $R_Y(c)$. After that, each node along the $-X$ boundary will form a triple $(F(c), R_{Y\_X}(c), R'_Y(c))$. The edge of R_Y, which is also the path of boundary construction, will be identified as one bound of $R_Y(c)$. Similarly, each node along the $+X$ boundary will form a triple $(F(c), R_{Y_+X}(c), R'_Y(c))$. Obviously, we have $R_Y(c) = R_{Y\_X}(c) \cup R_{Y_+X}(c)$. To obtain information of the other bound and further to identify $R_Y(c)$, the triple formed at a node along one boundary will be sent along the X dimension to reach the other boundary. At each intermediate node it passes, such information will also be sent in the $+Y$ direction. It is noted that the determined information is used in such a broadcasting and each node will not accept duplicates from its neighbors. Eventually, at each node inside the forbidden region $R_Y(c)$, the identified information $(F(c), R_Y(c), R'_Y(c))$ forms. The whole information propagation process is shown in Algorithm 4.

Algorithm 4: Complete boundary construction and information propagation for the forbidden region $R_Y(c)$ of an MCC $F(c)$ (proposed information model $B2$, replacing Algorithm 1)

1. Apply step 1 in Algorithm 1.
2. After the triple information is identified at its opposite corner c', it will propagate as $(F(c), R_{Y_+X}(c), R'_Y(c))$ to build the $+X$ boundary with the construction process of $-X$ boundary in step 3 in Algorithm 1, but always make a left turn.

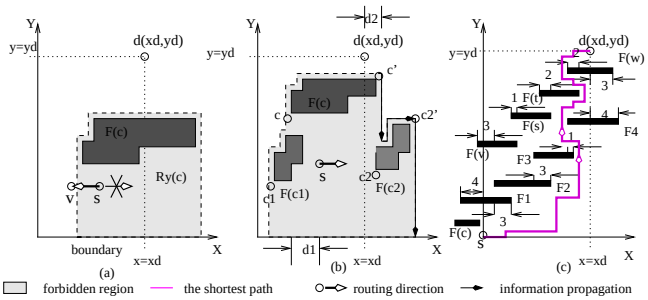

Figure 4. (a) A "must-take" detour. (b) A sample of complete information broadcast and a correct routing decision based on such information. (c) Multi-phase routing through the corner c_4 of F_4.

3. Apply step 2 in Algorithm 1.
4. Apply step 3 in Algorithm 1 to build $-X$ boundary and form the triple $(F(c), R_{Y\_X}(c), R'_Y(c))$ at each boundary node.
5. Simultaneously, the triple received at each boundary node will broadcast along the X dimension and then in the $+Y$ direction, until it reaches the other boundary. For each node we receive triples from both boundaries, $(F(c), R_Y(c), R'_Y(c))$ is identified.

Therefore, the routing decision at node s knows the existence of the blocking MCC. According to the location and the shape of such an MCC, the detour direction can be easily determined at node s: If $M(s,c)+M(c,d) \leq M(s,c')+M(c',d)$, the routing detours along the $-X$ direction. Otherwise, the $+X$ direction detour is taken (see in Figure 4 (b)).

As indicated in [8], the Manhattan routing is usually blocked by a sequence of MCCs, not only a single one. If the routing is blocked in the $+Y$ direction, the corresponding sequence (F_1, F_2, ..., F_n) is called type-I sequence. Respectively, we have type-II sequence blocking the routing in the $+X$ direction. Let c_i and c'_i denote the initialization corner and the opposite corner of each MCC F_i, $1 \leq i \leq n$, respectively. A type-I sequence can be identified by the following properties:

$$\begin{cases} (0, y_{c_1}) \in F_1 \wedge 0 < y_{c'_1} < y_d \\ (x_d, y_{c_n}) \in F_n \wedge 0 < y_{c_n} < y_d \\ x_{c_i} \leq x_{c_{i+1}} \leq x_{c'_i} \\ y_{c'_i} < y_{c'_{i+1}} \\ \nexists F(v), R_Y(c_i) \subset R_Y(v) \subset R_Y(c_{i+1}) \end{cases} \quad (1)$$

In Equation 1, the first four properties guarantee $R_Y(c_i) \subset R_Y(c_{i+1})$ for any $1 \leq i < n$. The last property guarantees that every shortest-path will be considered for the detour around the blocking MCC(s). Under our new information model with Algorithm 4, the routing at the source s ($u = s$) knows the information of each MCC of its blocking sequence when $D(s,d) > M(s,d)$. Furthermore, it knows the information of all MCCs blocking the Manhattan distance path to the destination d. Assume that F_1, F_2, $\cdots$, F_n is the closest sequence, with the initialization corner c_i and the opposite corner c'_i for each each F_i ($1 \leq i \leq n$). The routing has three options to detour around the sequence in the shortest-path: (a) through node c_1, denoted by P_0, (b) through node c'_n, denoted by P_n, and (c) through two consecutive MCCs, F_i and F_{i+1} ($1 \leq i < n$), denoted by P_i. The length of its shortest-path to the destination can be easily determined in a recursive function:

$$D(u,d) = \begin{cases} \min_{0 \leq i \leq n}\{P_i\} & the\ closest\ blocking \\ & sequence\ F_1, F_2, \cdots, \\ & F_n\ is\ found. \\ M(u,d) & otherwise \end{cases} \quad (2)$$

where

$$P_i = \begin{cases} M(u,c_1) & +D(c_1,d) & i = 0 \\ M(u,c'_i) & +M(c'_i,c_{i+1}) & 1 \leq i < n \\ & +D(c_{i+1},d) & \\ M(u,c'_n) & +D(c'_n,d) & i = n \end{cases} \quad (3)$$

After that, for the shortest-path P_i, the routing message will be forwarded to the corresponding corner, c_{i+1} ($0 \leq i < n$) or c'_n. And then, from that intermediate destination, the routing will continue until the destination d is reached. The detailed multi-phase routing process is shown in Algorithm 5.

Algorithm 5: Routing at the current node u with $d(x_d, y_d)$ ($RB2$)

1. If $u = d$, stop.
2. If no blocking sequence is found (with Equation 1), apply routing decision in Algorithm 2 to forward the routing message. Otherwise, apply the followings to detour.
3. Among all the sequences found in step 1, identify the closest one (F_1, F_2, $\cdots$, F_n).
4. Use Equation 2 to calculate the distance of the shortest-path $D(s,d)$ among the following paths: (a) through node c_1, denoted by P_0, (b) through node c'_n, denoted by P_n, and (c) through two consecutive MCCs (F_i and F_{i+1}, $1 \leq i < n$), denoted by P_i.
5. If P_0 / P_n is the selected shortest-path, apply routing decision in Algorithm 2 to reach the intermediate destination $d' = c_1$ / c'_n. Otherwise, for the selected shortest-path P_i ($1 \leq i < n$), apply routing decision in Algorithm 2 to form the Manhattan distance paths $M(s,c'_i)$ and $M(c'_i,c_{i+1})$ to reach the intermediate destination $d' = c_{i+1}$.

An example is shown in Figure 4 (c). At the source s ($u = s$), the closest sequence (F_1, F_2, F_3, and F_4) is found with Equation 1. Meanwhile, the information of all MCCs (F_1, F_2, F_3, F_4, $F(v)$, $F(s)$, $F(t)$, $F(w)$) and their initialization / opposite corners (c_1/c'_1, c_2/c'_2, c_3/c'_3, c_4/c'_4, v/v',

s/s', t/t', and w/w') can be obtained because s is inside their forbidden regions. Initially, we have $D(v,d) = M(v,d)$, $D(s,d) = M(s,d)$, $D(t,d) = M(t,d)$, $D(w,d) = M(w,d)$, $D(w',d) = M(w',d)$, $D(c_1,d) = M(c_1,d)$, and $D(c'_4,d) = M(c'_4,d)$. Then, we have $D(c_2,d) = M(d_2,v') + M(v',s) + D(s,d) = M(c_2,d) + 1 * 2$, $D(c_3,d) = M(c_3,s') + M(s',t) + D(t,d) = M(c_3,t') + M(t',w) + D(w,d) = M(c_3,d) + 2 * 2$, $D(c_4,d) = M(c_4,t') + M(t',w) + D(w,d) = M(c_4,d) + 2 * 2$. By calling the recursive function in Equation 2, the shortest-path P_3 is determined because $M(s,c'_3) + M(c'_3,c_4) + D(c_4,d) = M(s,d) + 6$ is the minimum. That is, the routing will form the paths $M(s,c'_3)$ and $M(c'_3,c_4)$ to reach the intermediate destination c_4. After that, at node c_4, by repeating the above process, the routing will find a path to d passing nodes t' and w. The recursive function guarantees the entire path is the shortest-path in such a multi-phase routing process. The following theorem ensures there is no path shorter than the one found in $RB2$ routing.

Theorem 1: *For a given pair of the safe source and the safe destination under MCC model, the $RB2$ routing in Algorithm 5 will find a path between them if such a path exists and there is no path shorter than this one.*

Proof: Assume that s and d are not disconnected by faults. If a Manhattan distance path between s and d exists, such a path can be easily found via step 2 of $RB2$ routing by applying routing decision in Algorithm 2. Otherwise, at least one sequence of either type-I or type-II can be found between s and d. Because s and d are safe, we cannot find both kinds of sequences. By applying the recursive function D in Equation 2, the path that detours around such a sequence can be found via step 5. In this way, the routing can advance in the blocking direction. Because the distance from s and d in such a blocking direction is fixed, the multi-phase routing will reach the node having Manhattan distance path to d and will eventually arrive node d.

The routing path found in Algorithm 5 only has detours around the MCCs in the blocking sequence(s). If there exists another shorter path from s to d, such a path must pass a healthy but MCC unsafe node. Because s and d are all safe, using any node inside in MCC will definitely cause detour. In other words, we must have another shorter path only using safe nodes. This contradicts with the results of Equation 2. ■

The above shortest-path routing needs information broadcasting. To reduce the broadcast overhead, a more practical information model is proposed for our shortest-path routing. In this extended model, the information of each MCC $F(c)$ is only propagated to the boundary nodes. The propagation will be split into two when it intersects with another MCC $F(v)$, instead of just making a turn. These two split propagations will route around $F(v)$, one in the clockwise direction and the other in the counter-clockwise direction. After that, the first one will merge to the $+X$ boundary of $F(v)$ at its opposite corner v'. The second one will merge to the $-X$ boundary of $F(v)$ at its initialization corner v. When it is the first intersection of the $-X$ boundary and $x_c > x_{v'}$, $F(c)$ will be identified as a candidate of the succeeding MCC of $F(v)$ in its type-I sequence. The relation $F(v) \rightarrow F(c)$ will also be sent by each split propagation. It is noted that the region information, $R_Y(c)$ and $R'_Y(c)$, are only needed and forwarded by those nodes along the $-X$ boundary of $F(c)$. The details of our extended information model are shown in Algorithm 6.

Algorithm 6: Information propagation for an MCC $F(c)$ (extended information model $B3$)

1. Apply steps 1 and 2 in Algorithm 1.
2. Apply step 3 in Algorithm 1 to construct the $-X$ boundary.
3. When the propagation intersects with another MCC, say $F(v)$, the shape information $F(c)$ will be propagated in the way of $+X$ boundary through its opposite corner v' (as shown in step 2 in Algorithm 4).
4. If the above intersection is the first one of $-X$ boundary and $x_c > x_{v'}$, $F(c)$ might be the succeeding MCC of $F(v)$ in a type-I sequence. Thus, the relation $F(v) \rightarrow F(c)$ is sent in both the above propagations (steps 2 and 3) from that intersection.

At the initialization corner c of an MCC $F(c)$, among all relation records received $I(c) = \{F(c) \rightarrow F(v)\}$, the succeeding MCC of $F(c)$ in a type-I sequence, $F(w)$, can be determined by:

$$\begin{cases} F(c) \rightarrow F(w) \in I(c) \\ \forall F(\tau) \in \{F(\zeta) \mid F(c) \rightarrow F(\zeta) \in I(c)\}, y_w \leq y_\tau \end{cases} \quad (4)$$

Based on the propagation process in Algorithm 6, a boundary node of $F(c)$ will obtain the same relation information as its initialization corner c does. That is, this $F(w)$ can also be determined at any boundary node of $F(c)$. Therefore, at any boundary node δ, with the information of the destination node d, a type-I blocking sequence (F_1, F_2, $\cdots$, F_n) can be determined by:

$$F_i = \begin{cases} F(\alpha),\ x_\alpha < x_\delta < x_{\alpha'},\ y_\delta < y_{\alpha'} & i = 1 \\ \textit{the succeeding MCC of } F_{i-1} & i > 1 \\ \textit{determined in Equation } 4 & \\ F(\beta), x_{c_\beta} < x_d < x_{c'_\beta},\ y_{c_\beta} < y_d & i = n \end{cases} \quad (5)$$

Therefore, the routing decision at boundary node δ can have the knowledge of all the blocking sequences and the corresponding MCC shape information. By using the same strategy in $RB2$ routing in Algorithm 5, the routing will find a path to detour the blocking sequence and then reach the destination. The following theorem shows that such a path from δ to d can be guaranteed as well as the one obtained with Algorithm 5. The detailed routing process based

on our extended information model, $RB3$, is shown in Algorithm 7.

Theorem 2: *When the source is a boundary node, the path found in $RB3$ routing in Algorithm 7 will not be longer than that found in $RB2$ routing in Algorithm 5.*

Proof: If the Manhattan distance path exists between s and d, both Algorithm 5 and Algorithm 7 will apply routing decision in Algorithm 2 to find it. Otherwise, the information propagation in Algorithm 6 ensures the same blocking sequence can be identified in Equation 1 and Equation 5. By applying Equation 2, these two routings will find the same shortest-path. ■

Algorithm 7: Routing at the current node u with $d(x_d, y_d)$ ($RB3$)

1. Same as step 1 in Algorithm 5.
2. With Equation 5, find the closest sequence blocking the Manhattan distance path from u to d.
3. If such a sequence is not found, apply routing decision in Algorithm 2 to forward the routing message. Otherwise, apply the following the detour.
4. For each MCC $F(\tau)$ in the sequence found in the above step, with the initialization corner τ and the opposite corner τ', find the sequence blocking the Manhattan distance path from τ to d or the Manhattan distance path from τ' to d (with Equation 5).
5. Repeat step 3 until there is no new sequence identified.
6. Same as step 4 in Algorithm 5.
7. Same as step 5 in Algorithm 5.

As in the sample in Figure 4 (c), s is a boundary node of an MCC $F(c)$. With the information collected, the closest MCC is F_1 and its sequence (F_1, F_2, F_3, F_4) can be determined. After that, it is identified that no sequence blocks the path from the initialization corner of F_1 to d. Meanwhile, the sequence ($F(v)$, $F(s)$, $F(t)$, $F(w)$) blocking the path from the initialization corner of F_2 to d is identified. This process will continue until the Manhattan distance paths $M(v,d)$, $M(s,d)$, $M(t,d)$, $M(w,d)$ and $M(w',d)$ can be identified. That is, no new MCC blocks the routing path. Then, by applying Equation 2, the distance of shortest-path $D(s,d)$ can be determined. Furthermore, the corresponding intermediate destination can be found and the routing message will be forwarded along the shortest-path in multiple phases.

If node s is not on any boundary line but is inside the forbidden region of an MCC, the above routing will miss the shortest-path in the case P_0. However, if each MCC can be controlled within a certain size, the routing will quickly reach the boundary line. That is, the number of detours is limited and the length of routing path is very close to the minimum. Moreover, the more MCCs that appear in the mesh, the greater are our chances the our routing finds the shortest-path among the remaining n cases from P_1 to P_n. Considering the communication cost saved under the extended information model, such a sub-optimal routing using only the boundary information is preferred with the performance still acceptable. In the next section, we use the experimental results to illustrate the substantial improvement of the extended information model on communication cost in terms of the number of nodes involved in the information propagation. Our experimental results also show the acceptable performance of routing under the extended information model in terms of (a) the success rate of the shortest-path routing, and (b) the relative error of the average length of routing path to the optimal result. These results will be compared with the best results so far in [2] and [5].

4 Simulation

In this section, we verify the improvement of our information based routing on the ability of achieving the shortest-path from a simulator, comparing with the best results so far. Such experimental results prove the effectiveness of our information models. The simulator also compares the implementation of our information model and its extension on the communication cost in the information propagation. The results show that the routing using only boundary information is cost-effective.

This simulator is conducted on a 100×100 mesh with numbers of faulty nodes randomly generated. It is noted that when more than 3000 faults occur in the mesh, the entire network will be disabled under MCC model. To have a fair comparison, we only show the results when the number of faults is no more than 3000. The MCC configuration situation for our test is shown in Figure 5 (a) and (b). After that, we implement the boundary information model $B1$[5], the proposed information model $B2$, and its extension model $B3$. Finally, we randomly pick up the source and destination and conduct the corresponding routings $RB1$, $RB2$ and $RB3$, respectively. We assume that the source has the path to the destination. Thus, we only conduct the test in the cases when the entire mesh is not disconnected by faults.

Figure 5 (c) shows the percentage of the number of nodes involved in the information propagation to the total safe nodes in the meshes in the information models $B1$, $B2$, and $B3$. The results show that $B2$ has the highest communication cost. However, it is still not as expensive as using global information. It is noted that when the entire mesh has up to 100 MCCs, the information only needs to broadcast to 20% of the safe nodes. $B1$ has the lowest communication cost. The results of $B3$ are very close to those of $B1$ because in most cases the $+X$ boundary of one MCC shares the nodes with the $-X$ boundaries of other MCCs. Figure 5 (d) shows the percentage of finding the shortest-path in differ-

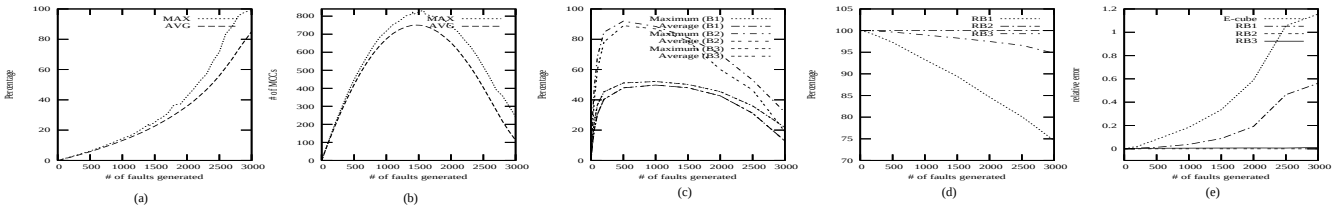

Figure 5. (a) Percentage of disabled area to the total area of the meshes (b) Number of MCCs. (c) Percentage of nodes involved in information propagation to the total safe nodes in the meshes. (d) Percentage of success in finding the shortest-path. (e) Relative error of routing path achieved to the shortest-path.

ent routings $RB1$, $RB2$ and $RB3$. The results show that with the information broadcasting, the routing $RB2$ always achieves the shortest-path ($= 100\%$). With the help of only $-X$ boundaries, the routing $RB1$ can successfully find the shortest-path in more than 75% of all cases. With the information model $B3$ proposed in this paper, the corresponding routing $RB3$ can find the shortest-path in more than 95% of all cases. However, under $B3$ model, there is no need for information broadcast. Figure 5 (e) shows the comparison of length of routing path achieved in each routing with the optimal result, i.e., the length of the shortest-path. It shows that the routing $RB1$ will experience many detours in the cases when the Manhattan distance path does not exist. The average length of the routing path is very close to that of E-cube routing in [2], which only requires the information of neighbors. Under the proposed information model $B2$, the corresponding routing $RB2$ will guarantee the shortest-path (relative error $= 0$). Under the proposed extension model $B3$, the routing $RB3$ will find the shortest-path in most cases, and for those non-shortest-path cases, the number of detours is limited. The results are very close to the optimal ones. This figure supports our statement on the significance of our new information model and its extension on achieving the shortest-path routing.

5 Conclusion

In this paper, we have proposed a fully distributed process to collect and distribute MCC block information in 2-D meshes for the corresponding information-based routing to achieve the shortest-path. The use of MCC model guarantees that there is no existing path shorter than the one we found. Its practical implementation with only a low number of nodes along the boundary lines involved in the information propagation is also presented. The simulation results have shown the substantial improvement of our methods on achieving the shortest-path routing in terms of the success rate and the average path length. In our future work, we will also extend our results to higher dimension networks and networks with irregular topology.

References

[1] Definition of the Manhattan distance, available at http://www.nist.gov/dads/HTML/manhattanDistance.html.

[2] R. V. Boppana and S. Chalasani. Fault-tolerant wormhole routing algorithm for mesh networks. *IEEE Transactions on Computers*, pages 848–864, July 1995. Vol. 44, No. 7.

[3] A. Gara and et al. Overview of the Blue Gene/L system architecture. *IBM Journal of Research and Development*. Vol. 49, No. 2, 2005, pp. 195-212.

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

[5] Z. Jiang, J. Wu, and D. Wang. A new fault information model for fault-tolerant adaptive and minimal routing in 3-D meshes. *Proc. of ICPP'05*. 2005, pp. 500-507.

[6] S. Kumar, A. Jantsch, M. Forsell J. Soininen, M. Millberg, J. Öberg, K. Tiensyrjä, and A. Hemani. A network on chip architecture and design methodology. *Proc. of VLSI Annual Symposium (ISVLSI'02)*. 2002, pp. 105-112.

[7] M. Schäfer, T. Hollstein, H. Zimmer, and M. Glesner. Deadlock-free routing and component placement for irregular mesh-based networks-on-chip. *Proc. of 2005 IEEE/ACM Int'l Conf. on Computer-aided Design*. 2005, pp. 238-245.

[8] D. Wang. A rectilinear-monotone polygonal fault block model for fault-tolerant minimal routing in meshes. *IEEE Transactions on Computer*. Vol. 52, No. 3, March 2003.

[9] J. Wu. Fault-tolerant adaptive and minimal routing in mesh-connected multicomputers using extended safety levels. *IEEE transactions on Parallel and Distributed Systems*. Vol. 11, No. 2, February 2000, pp. 149-159.

[10] J. Zhou and F. Lau. Multi-phase minimal fault-tolerant wormhole routing in meshes. *Parallel Computing*. Vol. 30, No. 3, 2004, pp. 423-442.