

Reliable Unicasting in Faulty Hypercubes Using Safety Levels

Jie Wu, *Senior Member, IEEE*

Abstract—We propose a unicasting algorithm for faulty hypercubes (including disconnected hypercubes) using the safety level concept. A faulty hypercube is a hypercube with faulty nodes and unicasting is a one-to-one communication between two nodes in the hypercube. Each node is associated with a safety level which is an approximated measure of the number and distribution of faulty nodes in the neighborhood. The safety level of each node in an n -dimensional hypercube (or n -cube) can be easily calculated through $n - 1$ rounds of information exchange among neighboring nodes. Optimal unicasting between two nodes is guaranteed if the safety level of the source node is no less than the Hamming distance between the source and the destination. The proposed unicasting algorithm can also be used in disconnected hypercubes, where nodes in a hypercube are disjointed into two or more parts. The feasibility of an optimal or suboptimal unicasting can be easily determined at the source node by comparing its safety level, its neighbors' safety levels, and the Hamming distance between the source and the destination. The proposed scheme is the first attempt to address the unicasting problem in disconnected hypercubes. The safety level concept is also extended to be used in hypercubes with both faulty nodes and links and in generalized hypercubes.

Index Terms—Disconnected networks, fault tolerance, hypercubes, reliable communication, unicast.

1 INTRODUCTION

WITH its numerous attractive features, the binary unicasting hypercube has been one of the commonly used topological structures for distributed-memory systems. Efficient interprocessor communication is a key to the performance of a hypercube system. Unicast is a one-to-one communication between a source and a destination, and it has been extensively studied in fault-free hypercubes.

As the number of processors in a hypercube system increases, the probability of processor failure also increases. There has been a number of fault-tolerant unicasting schemes proposed in previous work [3], [6], [7], [8]. Most of these models assume that each node knows either only the neighbors' status or the status of all the nodes. A model that uses the former assumption is called *local-information-based*, while a model that uses the later assumption is called *global-information-based*. In general, a global-information-based model can obtain an optimal or suboptimal result; however, it requires a complex process that collects global information. Normally, global information is presented in a tabular format and it is not easy to use.

Local-information-models use a weaker but a more reasonable assumption; however, local information can only be used to achieve local optimization and most of approaches based on this model are heuristic in nature. Therefore, the length of a routing path is unpredictable in general, and global optimization, such as time and traffic in routing, is impossible. For example, in [5], a sidetracking approach is used to route messages in a faulty hypercube with node faults. A message is rerouted to a randomly chosen fault-free neighboring node when there exists no fault-free

neighbors that are along optimal paths to the destination node. In [3], Chen and Shin proposed a scheme based on depth-first search in which backtracking is required when all the forward links are blocked by faulty components. However, a *history* of visited nodes has to be kept as part of message. A simplified version of this approach that tolerates fewer faults was presented in [2], where routing is progressive without backtracking. Still routing paths are not optimal in general. An optimal unicasting using only local information was reported in [6] under a restricted model of fault distribution. Note that most of the existing fault-tolerant unicasting algorithms cannot be applied in *disconnected hypercubes*, where nonfaulty nodes in a faulty hypercube are partitioned into several disjoint parts.

We proposed in [9] a novel concept called *safety level* which is an integer associated with each node in the system. Safety level is a concise representation of the distribution of faulty nodes in the system. It is also considered as a special type of *limited global information*, a compromise between local-information and global-information based approaches. Because this type of information is easy to update and maintain and routing optimality is still preserved when this information is used to direct messages in the unicasting process, it is more cost effective than the others. Basically, each node in an n -cube is assigned a safety level k , where $0 \leq k \leq n$, and this node is called k -safe. A k -safe node indicates that there exists at least one Hamming distance path (i.e., the optimal path) from this node to any node within k distance. The safety level of each node can be calculated using a simple $(n - 1)$ -round iterative algorithm which is independent of the number and distribution of faults in the hypercube.

Among other unicasting schemes based on limited global information of fault distribution, Lee and Hayes [7] proposed the concept of *safe node* which requires a stronger condition than the one that defines the safety level. Therefore, the safe node set is a subset of the one based on our definition. The safe node set is decided in $O(n^2)$ rounds of information exchange among neighboring nodes. This algorithm can route a message via a path of length no longer than two plus the Hamming distance between the source and the destination as long as the hypercube is not fully unsafe, which is guaranteed provided that the number of faults is no more than $\lceil \frac{n}{2} \rceil$ in an n -cube.

Wu and Fernandez [10] extended the Lee and Hayes' safe node definition by relaxing certain conditions and hence increase the size of the safe nodes set and the degree of fault tolerance. A process that identifies the node status was also proposed in [10] and it needs fewer rounds than the one in [7] in general. However, it still requires $O(n^2)$ rounds in the worst case. Recently, a unicasting algorithm based on this enhanced safe node concept was proposed by Chiu and Wu [4]. They showed that a path of length no more than the Hamming distance between the source and the destination plus four can always be established in their algorithm as long as the hypercube is not fully unsafe.

The safety level concept is one further step to extend the safe node concept. In this paper, we show that the safety level concept covers a larger set of safe nodes than both Lee-Hayes' and Wu-Fernandez' definitions and it can be used for unicasting in various faulty hypercubes (including disconnected hypercubes) more effectively. We also prove that under both Lee-Hayes' and Wu-Fernandez' safe node definitions, the safe node set is empty for any disconnected hypercube; that is, the unicasting algorithms proposed by Lee-Hayes [7] and Chiu-Wu [4] are not applicable to disconnected hypercubes. When a faulty n -cube has fewer than n faulty nodes, the proposed unicasting algorithm ensures an optimal unicasting (generating an optimal path) or suboptimal unicasting (generating a path with a length of Hamming distance between the source and the destination plus two). Unlike most of existing

• The author is with the Department of Computer Science and Engineering, Florida Atlantic University, Boca Raton, FL 33431.
E-mail: jie@cse.fau.edu.

Manuscript received Mar. 28, 1995; revised Apr. 4, 1996.

For information on obtaining reprints of this article, please send e-mail to: transcom@computer.org, and reference IEEECS Log Number C96095.

algorithms, the proposed algorithm can be used for n -cubes with more than $n - 1$ faults without additional cost. Only a simple feasibility check is needed at the source node before using the unicast algorithm and the result is still optimal or suboptimal. We also demonstrate that the safety level concept can be used in a general faulty hypercube with both faulty nodes and links by treating two end nodes of each faulty link as faulty nodes. The safety level concept can be further extended to the *generalized hypercube* [1].

We make the following assumptions for the techniques used in this paper:

- 1) All the node faults are fault-stop, i.e., there are no malicious faults.
- 2) Fault detection and diagnosis algorithms exist, but we do not require such algorithms to be perfect.

We do assume that each node knows exactly the safety status of all its neighbors.

2 NOTATION AND PRELIMINARIES

2.1 Hypercubes

The n -dimensional hypercube (or n -cube) Q_n is a graph having 2^n nodes labeled from 0 to $2^n - 1$. Two nodes are joined by an edge if their addresses, as binary integers, differ in exactly one bit position. More specifically, every node a has address $a_{n-1}a_{n-2} \dots a_0$ with $a_i \in \{0, 1\}$, $0 \leq i \leq n - 1$, and a_i is called the i th bit (also called the i th dimension) of the address. We denote node a^i the neighbor of a along dimension i . Symbol \oplus denotes the bitwise exclusive OR operation on binary addresses of two nodes. Let $e^k = e_{n-1}e_{n-2} \dots e_0$ where $e_k = 1$ and $e_j = 0$, $\forall j \neq k$. For example $1101 \oplus e^2 = 1001$. Clearly, $a \oplus e^i$ represents setting or resetting the i th bit of a . The distance between two nodes s and d is equal to the Hamming distance between their binary addresses, denoted by $H(s, d)$.

A path connecting two nodes s and d is termed an *optimal path* (also called *Hamming distance path*) if its length is equal to the Hamming distance between these two nodes. A *shortest path* is a path of minimal length among all the available paths between these two nodes. Clearly, $s \oplus d$ has value 1 at $H(s, d)$ bit positions corresponding to $H(s, d)$ distinct dimensions. These $H(s, d)$ dimensions are called *preferred dimensions* and the corresponding nodes are termed *preferred neighbors*. The remaining $n - H(s, d)$ dimensions are called *spare dimensions* and the corresponding nodes are *spare neighbors*.

2.2 Safety Levels

In a given n -cube, the safety level of each node ranges from 0 to n . The safety level associated with a node is an approximation of the number and distribution of faulty nodes in the neighborhood, rather than just the number of faulty nodes. Let $S(a) = k$ be the safety status of node a , where k is referred to as the level of safety, node a is called k -safe. A faulty node is 0-safe which corresponds to the lowest level of safety, while an n -safe node (also called a *safe node*) corresponds to the highest level of safety. A node with k -safe status is called *unsafe* if $k \neq n$.

DEFINITION 1 [9]. *The safety level of a faulty node is 0. For a nonfaulty node a , let $(S_0, S_1, S_2, \dots, S_{n-1})$, $0 \leq S_i \leq n$, be the nondecreasing safety level sequence of node a 's n neighboring nodes in an n -cube, such that $S_i \leq S_{i+1}$, $0 \leq i \leq n - 2$. The safety level of node a is defined as: if $(S_0, S_1, S_2, \dots, S_{n-1}) \geq (0, 1, 2, \dots, n - 1)$,¹ then $S(a) = n$ else if $(S_0, S_1, S_2, \dots, S_{k-1}) \geq (0, 1, 2, \dots, k - 1) \wedge (S_k = k - 1)$ then $S(a) = k$.*

The safety level of a nonfaulty node is recursively defined in

1. $seq_1 \geq seq_2$ if and only if each element in seq_1 is greater or equal to the corresponding element in seq_2 .

terms of its neighbors' safety levels. The following theorem shows that for any faulty hypercube there is one and only one safety level for each node that satisfies the condition in Definition 1.

THEOREM 1. *For any given faulty hypercube, there is one and only one way to assign safety levels to nodes that satisfies the safety level condition in Definition 1.*

PROOF. We first show that there exists a feasible safety level assignment for any given faulty hypercube and then we show that it is the only possible one.

For a given faulty n -cube, Q_n , we perform n -rounds of safety level assignment. Each nonfaulty node is assigned once and only once. All the assignments in the k th round are of safety level k . In the first round, any nonfaulty node that has two or more faulty neighbors is assigned safety level 1. In the second round, any unassigned nonfaulty node that has three or more neighbors whose safety levels is no more than 1 is assigned a safety level 2. This process continues from round to round. At the k th round, any unassigned nonfaulty node that has $k + 1$ or more neighbors whose safety levels are no more than $k - 1$ is assigned a safety level k . Clearly, the safety level k assigned to node a at the k th round meets the safety level condition. Also, any later assignment (with a safety level larger than k) of an unassigned nonfaulty neighbor of a will not affect a 's safety status. That is, once a nonfaulty node is assigned a safety level, this level remains feasible. At the last round, a safety level n is given to all the unassigned nonfaulty nodes (if there are any).

To prove there is only one safety level assignment, let's assume that there are two different assignments $S^{(1)}, S^{(2)}$ both satisfying the safety level condition. Select a node a that has two different safety levels and $\min\{S^{(1)}(a), S^{(2)}(a)\} = \min\{\min\{S^{(1)}(s), S^{(2)}(s)\} \mid S^{(1)}(s) \neq S^{(2)}(s), \forall s \in Q_n\}$; that is, among nodes that have different safety level assignments, node a has the lowest safety level in one of two assignments.

Without loss of generality, let $S^{(1)}(a) = k_1$, $S^{(2)}(a) = k_2$ and $k_1 < k_2$.

Assume that $(S_0^{(1)}, S_1^{(1)}, S_2^{(1)}, \dots, S_{k_1-1}^{(1)}, S_{k_1}^{(1)}, \dots, S_{n-1}^{(1)})$ and $(d_0, d_1,$

$d_2, \dots, d_{k_1-1}, d_{k_1}, \dots, d_{n-1})$ are a nondecreasing neighbor sequence of node a and the corresponding dimension sequence based on $S^{(1)}$. Based on the safety level definition,

$(S_0^{(1)}, S_1^{(1)}, S_2^{(1)}, \dots, S_{k_1-1}^{(1)}) \geq (0, 1, 2, \dots, k_1 - 1)$ and $S_{k_1}^{(1)} = k_1 - 1$.

Let $(S_0^{(2)}, S_1^{(2)}, S_2^{(2)}, \dots, S_{k_1-1}^{(2)}, S_{k_1}^{(2)}, \dots, S_{n-1}^{(2)})$ be the neighbor

status sequence (not necessarily nondecreasing) based on the safety assignment $S^{(2)}$ and the dimension sequence

$(d_0, d_1, d_2, \dots, d_{k_1-1}, d_{k_1}, \dots, d_{n-1})$. Because node a has the lowest safety level among all the nodes that have different safety level assignments, $S_i^{(1)} = S_i^{(2)}$ for all $0 \leq i \leq k_1$, and

$S_i^{(1)} = S_j^{(2)}$ or $S_j^{(2)} \geq k_1$ for all $k_1 < j \leq n - 1$. Based on the safety

level definition, the safety level of a based on $S^{(2)}$ is also k_1 ,

i.e., $k_1 = k_2$. This brings up a contradiction to the assumption

that $k_1 \neq k_2$. \square

The proof of Theorem 1 also suggests a simple way to identify the safety level of every node in a given faulty hypercube. The following iterative algorithm (GS) calculates the safety level of each node in an n -cube. For simplicity, we show here only the synchronous version of GS, although it can be implemented

asynchronously. We assume that all nonfaulty nodes in Q_n have n as their initial safety levels (in this way the n th round in the proof of Theorem 1 is not necessary and there is no need to calculate safety levels when the hypercube is fault-free) and N is the set of all the nonfaulty nodes in Q_n . The selection of Δ , the number of iterations used in GS, will be discussed in the next section.

Algorithm GLOBAL_STATUS (GS)

{Initially all nonfaulty nodes are n -safe, faulty nodes are 0-safe, and $round = 1$.}

```

begin
  while round ≤ Δ
    parbegin
      NODE_STATUS(a), ∀ a ∈ N
    parend;
    round := round + 1
  end.

```

Procedure NODE_STATUS(a)

```

begin
  at node a determine the nondecreasing status sequence of
  neighboring nodes ( $S_0, S_1, S_2, \dots, S_{n-1}$ )
  if ( $S_0, S_1, S_2, \dots, S_{n-1} \geq (0, 1, 2, \dots, n-1)$ )
    then mark a as n-safe (or safe);
  if ( $(S_0, S_1, S_2, \dots, S_{k-1}) \geq (0, 1, 2, \dots, k-1) \wedge (S_k = k-1)$ )
    then mark a as k-safe;
end.

```

Fig. 1 shows the safety level of each node in a faulty four-cube with four faulty nodes (represented as black nodes): 0011, 0100, 0110, and 1001. Based on the safety level definition, the safety levels of all the nodes that have two (or more) faulty neighbors will be changed to 1 after the first round, as in the case for nodes 0001, 0010, 0111, 1011 in Fig. 1. That is, the effect of 0-safe status of faulty nodes first propagated to their neighbors, then neighbors' neighbors and so on. For example, after the second round the safety level of node 0101 changes to 2, because this node has three 1-safe or 0-safe neighbors. Similarly, the safety level of node 0000 changes to 2. The safety level of each node remains stable after two rounds and each value represents the safety level of the corresponding node. Note that in the absence of faulty nodes, all the nodes are safe and no extra overhead is introduced using this approach.

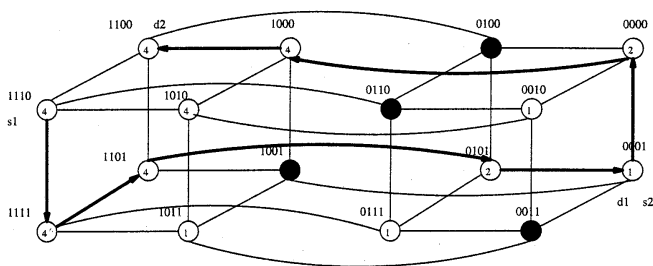


Fig. 1. A four-cube with four faulty nodes.

There are several ways to keep safety level information up-to-date.

- 1) *Demand-driven*: The GS is applied only when a node detects an inaccurate safety level of one of its neighbors caused by the occurrence (or recovery) of faulty nodes in the neighborhood during a unicast. The recovery of a faulty node will not cause disruption of a unicast. However, in case of occurrence of a new faulty node that affects a unicast, this unicast might either be aborted or be re-routed from the current node after all the safety levels are stabilized.
- 2) *Periodic*: Each node exchanges safety information periodically with its neighbors. Note that this approach does not adapt the activity to the failure rate of nodes. For example,

all (or most) exchanges are wasted when all (or most) of nodes' status remain stable.

- 3) *State-change-driven*: A node initiates a GS whenever it detects that a neighboring node fails (or recovers). In this case, the GS algorithm can be implemented asynchronously as in the demand-driven approach.

2.3 Properties of Safety Levels

We consider here properties of safety levels that are useful for efficient unicasting in a faulty hypercube.

PROPERTY 1 [9]. *The GS algorithm identifies a k-safe ($k \neq n$) node of an n-cube in k rounds, i.e., at the kth round this node reaches a stable status.*

COROLLARY. *To identify the status of all the nonfaulty nodes in any faulty hypercube (which might be a disconnected hypercube), the number of rounds (Δ in GS) is $n - 1$, where n is the dimension of the faulty hypercube.*

We compare the proposed method with other definitions of safe nodes. Two measures are used:

- 1) the number of steps required to determine the status of a node,
- 2) the size of safe node set.

The following are two other definitions of safety status of a node. Again, we assume that initially all nonfaulty nodes' are safe.

DEFINITION 2 (Lee and Hayes [7]). *A nonfaulty node is unsafe if and only if there are at least two unsafe or faulty neighbors.*

DEFINITION 3 (Wu and Fernandez [10]). *A nonfaulty node is unsafe if and only if one of the following conditions is true: There are two faulty neighbors, or there are at least three unsafe or faulty neighbors.*

In general, it is difficult, if not impossible, to compare the size of safe node sets because each safe node set depends on the distribution of faulty nodes. However, it is clear that, for each distribution of faulty nodes, the safe node set obtained using the definition in this paper contains the set using the definition in [10], which in turn contains the set using the definition in [7]. Consider a four-cube with three faulty nodes: 0000, 0110, and 1111. Each node has three safety status based on the three different definitions. Using the safety definition proposed in this paper, we obtain the safe node set which is $\{0001, 0011, 0101, 1000, 1001, 1010, 1011, 1100, 1101\}$. Using Definition 3 [10], we have $\{0001, 0011, 0101, 1000, 1001, 1010, 1011, 1101\}$ as the safe node set with the absence of node 1100. The safe node set is empty using Definition 2 [7]. In summary, the safety level defined here provides more accurate information than the previous ones. Surprisingly, it takes fewer rounds ($n - 1$ by the above Corollary) to determine the safety level of each node than using both of the other definitions which require $O(n^2)$ rounds of information exchanges in the worst case.

Simulation results show that when the number of faulty nodes is less than the dimension of the hypercube, the average number of rounds is much lower than the worst case bound (i.e., dimension minus one). Fig. 2 shows that the number of rounds for seven-cubes with various number of faults is much lower than the worst case number. When the number of faulty nodes is less than the number of rounds for seven-cubes is less than 2.

PROPERTY 2 [9]. *In a faulty n-cube with fewer than n faulty nodes, each nonfaulty but unsafe node has a safe neighbor.*

For example, in the faulty four-cube with three faulty nodes: 0000, 0110, and 1101, all nonfaulty but unsafe nodes have at least one safe neighbor.

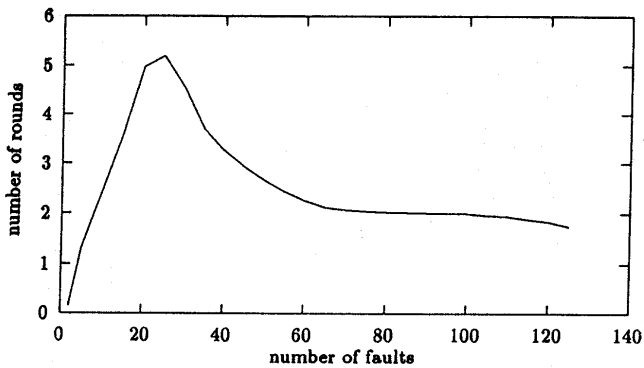


Fig. 2. Average number of rounds of information exchange for seven-cubes.

3 UNICASTING IN FAULTY HYPERCUBES

3.1 Basic Ideas

In this section, we propose an optimal unicasting scheme for faulty hypercubes. The scheme is a distributed one in which each intermediate (including the source) node routes the unicast message based on its own safety level and its neighbors'. Optimality (or suboptimality²) is ensured when the safety level of the source or one of its neighbors meets certain safety level decided by the Hamming distance between the source and the destination. The selection of a neighbor, among all its preferred neighbors, to forward the unicast message is based on the safety level of this neighbor. The following result serves as a basis for our approach.

THEOREM 2. *If the safety level of a node is k ($0 < k \leq n$), then there is at least one Hamming distance path from this node to any node within k Hamming distance.*

PROOF. We prove this theorem using the mathematical induction on j , the distance between the source and the destination.

When $j = 1$, the source node can clearly reach all the neighbors, faulty and nonfaulty. Assume that this theorem holds true for all $j < i$. When $j = i$, based on the hypercube property that there are j node-disjoint optimal paths between two nodes separated by j Hamming distance, there are j preferred dimensions of a source node to any destination node which is j distance away. Based on the safety level definition and assuming that the safety level is $k (\geq j)$, the nondecreasing safety level sequence of the source node's neighboring nodes satisfies the following condition: $(S_0, S_1, S_2, \dots, S_{k-1}) \geq (0, 1, 2, \dots, k-1) \wedge (S_k = k-1)$. Therefore, among $j (\leq k)$ preferred neighbors, there is at least one neighbor whose safety level is at least $j-1$. Because the distance between this neighbor and the destination node is $j-1$ and based on the induction assumption, there exists an optimal path from the source node to the destination node. \square

The above result also provides a simple way to identify an optimal path between two nodes: An optimal path is generated by selecting a preferred neighbor with the highest safety level at each routing step. We consider two unicasting algorithms: one is an optimal algorithm in which a message is guaranteed to be forwarded to the destination node along an optimal path and the other is a suboptimal algorithm in which a message is forwarded to the destination node along a path with a length of the Hamming

2. A unicasting is called suboptimal if it generates a path connecting a given pair of source and destination nodes that has a length of Hamming distance plus two.

distance between the source and the destination plus two. The selection between the optimal and the suboptimal algorithms can be decided locally at the source node, using the following information:

- 1) The Hamming distance of the source s and the destination d , $H(s, d) = |s \oplus d|$. The preferred neighbor sets and the spare neighbor sets are obtained based on $s \oplus d$.
- 2) The safety level of the source node $S(s)$.
- 3) The safety levels of the neighboring nodes of the source node $S(s^0), S(s^1), S(s^2), \dots, S(s^{n-1})$ for neighbors along dimensions $0, 1, 2, \dots, n-1$, respectively.

Note that safety levels of neighbors are available at each node and they are updated at each application of GS.

A navigation vector $N = s \oplus d$ is introduced here which is calculated at the source node and is passed to a selected neighbor after resetting or setting the corresponding bit depending on whether this neighbor is a preferred one or a spare one. The safety level requirement for optimal unicasting is as follows: The safety level of the source is at least equal to the Hamming distance between the source and the destination or one preferred neighbor's safety level is at least equal to the Hamming distance minus one. If there is no such preferred neighbor, then a spare neighbor whose safety level is at least equal to the Hamming distance plus one (the condition for suboptimal unicasting) is selected and the corresponding algorithm is suboptimal. At each intermediate node, a preferred neighbor with the highest safety level is selected (for both optimal and suboptimal algorithms). Each intermediate node knows its preferred and spare neighbors upon receiving the unicast message with a navigation vector. A unicasting completes when the navigation vector becomes zero; that is, each bit is zero. Note that, at the source node, if both conditions for optimal and suboptimal unicasting fail then the routing algorithm fails. This failure state can be easily detected at the source node. The cause of failure can be either too many faulty nodes in the neighborhood or a network partition (this case will be discussed in the next section). As shown in Property 2, when there are fewer than n faulty nodes in an n -cube, the proposed unicasting algorithm never fails (each unsafe but nonfaulty node has a safe neighbor); that is, the proposed algorithm is either optimal or suboptimal. When there are more than $n-1$ faulty nodes, the proposed unicasting algorithm may still work (at certain source nodes with appropriate routing distances as in Fig. 1) depending on the distribution of faults.

3.2 Unicasting Algorithm

At the source node s with unicast message m and destination d :

Algorithm UNICASTING_AT_SOURCE_NODE

$N = s \oplus d; H = H(s, d);$

/ calculate the navigation vector and the Hamming distance */*

if $(C_1 : S(s) \geq H) \vee (C_2 : \exists i(S(s^i) \geq H-1 \wedge N(i) = 1))$

/ the safety level of the source is at least equal to the Hamming distance or a preferred neighbor's safety level is at least equal to the distance minus one */*

then OPTIMAL_UNICASTING:

send $(m, N \oplus e^i)$ to s^i , where $S(s^i) = \max\{S(s^j) \mid N(j) = 1\}$

/ send the message to the preferred neighbor s^i with the highest safety level together with N after resetting bit i */*

else if $C_3 : \exists i(S(s^i) \geq H+1 \wedge N_i = 0)$

/ one spare neighbor's safety level is at least equal to the distance plus one */*

then SUBOPTIMAL_UNICASTING:

send $(m, N \oplus e^i)$ to s^i , where $S(s^i) = \max\{S(s^j) \mid N(j) = 0\}$

/ send the message to the spare neighbor s^i with the highest safety level together with*

```

N after setting bit i*/
else failure

```

At any intermediate node a with unicast message m and navigation vector N :

Algorithm UNICASTING_AT_INTERMEDIATE_NODE

```

if  $N=0$  /* the navigation vector is empty */
then stop /* the current node is the destination */
else send  $(m, N \oplus e^i)$  to  $a^i$ , where  $S(a^i) = \max\{S(a^j) \mid N(j) = 1\}$ 
/* send the message to the preferred neighbor  $a^i$ 
with the highest safety level together with  $N$ 
after resetting bit  $i$  */

```

Consider again the example in Fig. 1. The number in each circle (node) represents the safety level of this node. In a unicast where $s_1 = 1110$ and $d_1 = 0001$ are the source and the destination nodes, respectively, the navigation vector is $N_1 = s_1 \oplus d_1 = 1111$; hence, $H(s_1, d_1) = 4$. Also, the safety level of the source s_1 is 4. Therefore, the optimal algorithm is applied. Among preferred neighbors of the source, nodes 1010, 1100, and 1111 have a safety level 4 and node 0110 has a safety level 0. A neighbor with the highest safety level, say 1111 along dimension 0, is selected. The navigation vector N is sent together with the unicast message after resetting bit 0. At intermediate node 1111, based on the navigation vector 1110, the preferred neighbor set is calculated which is {0111, 1011, 1101}. Among preferred neighbors, node 1101 has the highest safety level (which is 4); therefore, 1101 is the next intermediate node with a navigation vector 1100. At node 1101, the preferred node 0101 (with a safety level 2) is selected among two preferred neighbors (the other one is the faulty neighbor 1001). At node 0101 with navigation vector 0100, there is only one preferred neighbor which is 0001. Upon receiving the unicast message with a navigation vector 0000, node 1100 identifies itself as the destination node and terminates the unicast algorithm.

Consider another unicast example in the faulty four-cube of Fig. 1, where $s_2 = 0001$ and $d_2 = 1100$ are the source and the destination. In this case, the safety level of the source (which is 1) is less than the Hamming distance between the source and the destination (which is 3). However, there are two preferred neighbors (0000 and 0101) whose safety levels are 2 (which is the Hamming distance minus one). Therefore, optimal unicast is still possible by selecting one of these two preferred neighbors, say 0000. The corresponding routing path $0001 \rightarrow 0000 \rightarrow 1000 \rightarrow 1100$ is shown in Fig. 1.

Note that if the source node is safe (the highest safety level), optimality is automatically guaranteed for any unicast. When one neighbor of the source is safe at least suboptimality is guaranteed. If this neighbor is a preferred one, then the unicast is optimal. If this neighbor is a spare one, then the unicast is suboptimal. Based on Property 2, any unsafe node has a safe neighbor in a faulty n -cube with no more than $n - 1$ faulty nodes. Suboptimality is guaranteed in such faulty hypercubes.

THEOREM 3. *Suppose the Hamming distance between the source and the destination is j for a given unicast. When the source is k ($\geq j$)-safe or there is an l ($\geq j - 1$)-safe preferred neighbor of the source node, optimality for any unicast is guaranteed using the proposed unicast algorithm. When there is an m ($\geq j + 1$)-safe spare neighbor of the source node, at least suboptimality is guaranteed.*

This theorem can be directly derived based on Theorem 2 and the proposed unicast algorithm.

3.3 Unicast in Disconnected Hypercubes

In this section, we show that the proposed unicast algorithm can be applied to various faulty hypercubes, including disconnected hypercubes. To our best knowledge, our approach is the

first one that addresses routing in disconnected hypercubes.

We first show an example of a disconnected hypercube and see how the proposed algorithm works in this case. Fig. 3 shows a disconnected four-cube with four faulty nodes: 0110, 1010, 1100, and 1111. Clearly, any unicast initiated at node 1110 will fail. The source node 1110 detects this situation by checking its neighbors' safety levels and its own safety level. However, unicast is possible if it is initiated from the other part of the partition. For example, in a unicast with source $s_1 = 0101$ and destination $d_1 = 0000$, the Hamming distance between the source and the destination is 2 and the safety level of the source is 2. Therefore, optimal unicast is possible and the corresponding path is shown in Fig. 3. In another example where $s_2 = 0111$ and $d_2 = 1011$, although the safety level of the source (1) is less than the Hamming distance (2), the preferred neighbor 0011's safety level is 2, which is more than the the Hamming distance minus one. Again, the optimal routing is possible in this case (see Fig. 3). When the destination is 1110, any unicast fails and this can be easily detected at the source node, say 0111. At 0111, the safety level of the source is 1, which is less than the Hamming distance $H(0111, 1110) = 2$ (condition C_1 fails) and none of the preferred neighbors' (0110 and 1111) safety level is more than the Hamming distance minus one (condition C_2 fails), hence there is no optimal unicast. Both spare neighbors' (0101 and 0011) safety levels are 2 which is less than the Hamming distance plus one (condition C_3 also fails), and hence there is no suboptimal unicast. Therefore, this unicast is aborted at the source node.

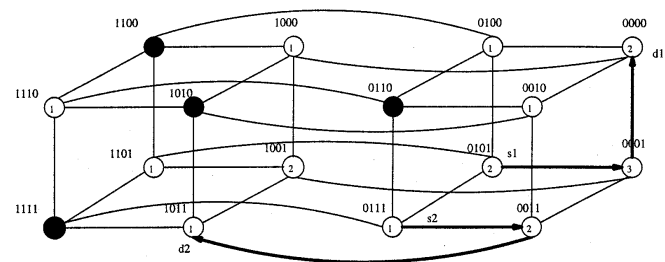


Fig. 3. Unicast in a disconnected four-cube with four faulty nodes.

The following theorem shows that in any disconnected hypercube all the nonfaulty nodes are unsafe based on the safe node definitions of both Lee-Hayes [7] and Wu-Fernandez [10]. Because both unicast algorithms in [7] and [4] need at least one safe node in the neighborhood of the source node, they cannot be used for cubes with no safe nodes. That is, they are not applicable in any disconnected hypercube. Because for any given faulty hypercube the safe node set based on Lee-Hayes' definition is a subset of the one based on Wu-Fernandez' definition, it suffices to prove that the safe node set based on Wu-Fernandez's definition is empty.

THEOREM 4. *Based on Wu and Fernandez's safe node definition, the safe node set is empty for any disconnected hypercube.*

PROOF. Assume that the nonfaulty node set of a given faulty n -cube is partitioned into two disjoint sets V and W . That is, any two nodes, one from each of V and W , are disconnected in the faulty n -cube.

Select a node v from V and assume that its address is 0000 (otherwise we can always remap nodes to make the address of 0000). We then classify nodes in the n -cube into n levels. A node with i one bits in its address belongs to level i . Note that any node at level i has i neighbors at level $i - 1$. Assume that level k is the lowest (closest to node v : 0000) level among the nodes in W , then k should be larger than 1;

otherwise, W and V are connected. Randomly select a node w from W at level k , there are at least $k \geq 2$ neighbors at level $k - 1$. Based on the fact that W and V are disconnected and k is the lowest level among the nodes in W , all k neighbors at level $k - 1$ are faulty. Therefore, node w is unsafe. Since node w is randomly selected, all nodes at level k that belong to W are unsafe.

At level $k + 1$, again we randomly select a node w' in W . Node w' has $k + 1 \geq 3$ neighbors at level k , among them some are unsafe nodes in W and the rest are faulty. Based on the Wu-Fernandez's definition (Definition 3) of safe node, a node is unsafe if there are three unsafe or faulty neighbors, all the nodes at level $k + 1$ that are in W are unsafe. The same argument can be applied level by level until reaching the highest level among the nodes in W . In this way, we prove that all the nodes in W are unsafe. Note that the highest level may or may not be n . By interchanging the role of V and W , we can prove that all the nodes in V are unsafe.

A disconnected hypercube with multiple disjoint parts can be considered as the result of a series of partitions of a disconnected hypercube with two disjoint parts. Clearly, the above conclusion applies to any disconnected hypercube with multiple disjoint parts. \square

Based on the above result, we conclude that the unicasting algorithms proposed by Lee-Hayes [7] and Chiu-Wu [4] are not applicable to any disconnected hypercube.

4 EXTENSIONS

4.1 Hypercubes with Both Faulty Links and Nodes

We first extend the safety level concept to a general faulty hypercube with both faulty links and nodes. The approach is based on the following idea: Nonfaulty nodes are classified into

- 1) nonfaulty nodes without adjacent faulty link(s) and set N_1 includes all these nodes, and
- 2) nonfaulty nodes with adjacent faulty link(s) and set N_2 includes all these nodes.

The safety levels of nodes from these two sets (N_1 and N_2) are defined differently representing two different views of the same system. From the view of a nonfaulty node in N_1 , each nonfaulty node in N_2 is treated as a faulty node. However, a nonfaulty node in N_2 considers itself as a regular healthy node but treats all the other end node(s) of adjacent faulty link(s) as faulty nodes. We assume that any nonfaulty node can distinguish an adjacent faulty link from an adjacent faulty nodes.

The GLOBAL_STATUS algorithm can be modified as follows: Each faulty node (F is the set of faulty nodes) is labeled as 0, each nonfaulty node with adjacent faulty link(s) declares itself faulty with an initial safety level 0, and each nonfaulty node without adjacent faulty link(s) has an initial safety level n . Apply first the regular GS algorithm only to nonfaulty nodes without adjacent faulty link(s) (nodes in N_1) and each nonfaulty node with adjacent faulty link(s) (a node in N_2) runs the NODE_STATUS algorithm once only in the last round.

Algorithm EXTENDED_GLOBAL_STATUS (EGS)

{Initially all nonfaulty nodes in N_1 are n -safe, all faulty nodes in F and nonfaulty nodes in N_2 are 0-safe, and $round = 1$.}

begin

while $round \leq n - 1$

parbegin

 NODE_STATUS(a), $\forall a \in N_1$;

if $round = n - 1$ **then** NODE_STATUS(a), $\forall a \in N_2$

parent;

round := $round + 1$

end.

Fig. 4 shows a faulty four-cube with four faulty nodes and one faulty link, and lists the safety levels of nonfaulty nodes after executing the GS algorithm. Node 1000 is 1-safe and node 1001 is 2-safe. However, both are treated as faulty³ by all the other nodes. Each node in N_2 (nodes 1000 and 1001) considers itself regular and uses the associated safety level to perform a unicasting. Assume that node 1101 needs to forward a message to node 1000. Because both preferred neighbors of node 1101 are faulty, there is no Hamming distance path between 1101 and 1000. However, the spare neighbor 1111 has a safety level of 4 which is more than the Hamming distance (which is 2) plus 1. Therefore, suboptimal routing is possible and the routing path is $1101 \rightarrow 1111 \rightarrow 1011 \rightarrow 1010 \rightarrow 1000$ as shown in Fig. 4.

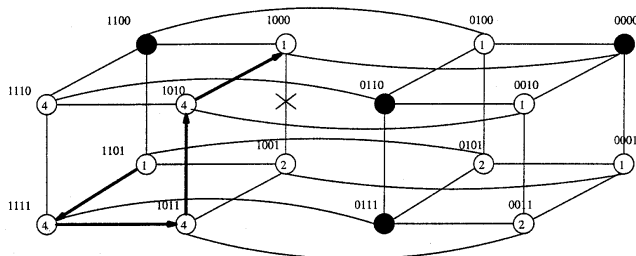


Fig. 4. A faulty four-cube with four faulty nodes and one faulty link.

The proposed routing algorithm (in Section 3) can also be used at nonfaulty nodes with adjacent faulty link(s). The rule is as follows: Suppose node a is the source node with adjacent faulty link(s), except for the end node(s) of adjacent faulty link(s) of a , there exists a Hamming distance path to any nodes with k distance, where k is the safety level of a nonfaulty node with adjacent faulty link(s). The same rule for suboptimality applies here.

4.2 Generalized n -Dimensional Hypercubes

Let N be the total number of processors (nodes) and be represented as a product of m_i 's, $m_i > 1$ for $0 \leq i < n$, i.e., $N = m_{n-1} \times m_{n-2} \times \dots \times m_1 \times m_0$. Each node corresponds to an n -vector $(a_{n-1}, a_{n-2}, \dots, a_1, a_0)$, where $0 \leq a_i \leq m_i - 1$. In a generalized n -dimensional hypercubes (GH_n) [1], two nodes are linked by an edge if they differ in exactly one coordinate.

The safety level of a node in GH_n is defined as follows: Each node's safety level (S) depends on the neighbors' safety levels. Neighbors are grouped by dimensions and each node needs only one safety level from nodes in the same dimension. Therefore, each node in a generalized n -dimensional hypercube still has an n -vector of neighbors' safety level. The definition of the safety level is the same as the one defined in regular hypercubes, which is based on the n -vector of neighbor's safety level. However, the definition of safety level along each dimension is different. The safety level along a dimension (say i) of node a (analogous to the safety level along this dimension in a regular hypercube) is the minimum safety level of all the nodes ($m_i - 1$ in total) along this dimension, i.e., the minimum of all the nodes that have the same addresses as node a except at coordinate (dimension) i .

DEFINITION 4. The safety level of a faulty node is 0. For a nonfaulty node a , let $(S_0, S_1, S_2, \dots, S_{n-1})$, $0 \leq S_i \leq n$, be the nondecreasing safety level sequence of node a 's n neighboring nodes in an n -cube, such

3. It is treated as a special fault. Although such a node may not be used as an intermediate node to forward a message, the message still needs to be forwarded to this node if it is a destination node.

that $S_i \leq S_{i+1}$, $0 \leq i \leq n - 2$. Let $(d_0, d_1, \dots, d_{n-1})$ be the corresponding dimension sequence of the safety level sequence. Each S_i is defined as follows: S_i is the minimum of safety levels among all the nodes that have the same coordinates as node a except at dimension d_i . The safety level of node a is defined as: if $(S_0, S_1, S_2, \dots, S_{n-1}) \geq (0, 1, 2, \dots, n - 1)$ then $S(a) = n$ else if $(S_0, S_1, S_2, \dots, S_{k-1}) \geq (0, 1, 2, \dots, k - 1) \wedge (S_k = k - 1)$ then $S(a) = k$.

The GLOBAL_STATUS algorithm can be easily extended to be applied to the generalized hypercube. The key is to obtain each element of $(S_0, S_1, \dots, S_{n-1})$. Because all the nodes along the same dimension are directly connected, the minimum safety level of all the nodes in the same dimension can be obtained in one step. Therefore, it still requires a total of $(n - 1)$ steps to obtain the safety status of each node in GH_n . The following is the extended NODE_STATUS algorithm (we will use the same GS algorithm as shown in Section 2).

Procedure EXTENDED_NODE_STATUS(a)

begin

{at node a determine the ascending status sequence of neighboring nodes $(S_0, S_1, S_2, \dots, S_{n-1})$ where $S_i = \min\{S(a^i) \mid a^i \text{ and } a \text{ differ only in the } i\text{th coordinate}\}$

if $(S_0, S_1, S_2, \dots, S_{n-1}) \geq (0, 1, 2, \dots, n - 1)$

then mark a as n -safe (or safe);

if $((S_0, S_1, S_2, \dots, S_{k-1}) \geq (0, 1, 2, \dots, k - 1) \wedge (S_k = k - 1))$

then mark a as k -safe;

end.

THEOREM 2'. If the safety level of a node (say a) in a generalized hypercube is k ($0 \leq k \leq n$), then there is at least one optimal path from this node to any node that differ from node a by at most k coordinates.

Routing in GH_n is exactly the same as in a regular hypercube, because all the nodes are directly connected along the same dimension. Fig. 5 shows a $2 \times 3 \times 2$ generalized hypercube with four faulty nodes. The number inside each cycle represents the safety level of the node. There are four nodes whose safety levels are 3, i.e., safe. Therefore, routing from any of these four nodes are optimal. Because each unsafe but nonfaulty node has a safe neighbor, routing from any of these nodes is at least suboptimal (the length of such a path is the distance between the source and destination plus two). Suppose node 010 is the source and node 101 is the destination (their addresses differ in three coordinates). At the first step, node 010 can forward the unicast message to one of four neighbors along one of three possible dimensions. The neighbor along dimension 0 is faulty and it is not eligible. The neighbor along dimension 2 has a safety level of 1 which is less than $3 - 1 = 2$ and again is not eligible. Therefore, the only eligible neighbors are the ones along dimension 1. Based on the relative distance between the source and the destination along dimension 1, node 000 is selected (as part of ring routing along this dimension). Once node 000 receives the message, it then forwards the message to one of its two eligible neighbors, in this case, the only choice is node 001 (with a safety level 1) which in turn forwards the message to the destination node 101. Another possible optimal path is $010 \rightarrow 020 \rightarrow 021 \rightarrow 121 \rightarrow 101$.

5 CONCLUSIONS

We have proposed a unicasting algorithm for faulty hypercubes. This algorithm uses limited global information captured by a safety level associated with each node. The safety level can be calculated through a simple $(n - 1)$ -round information exchange among neighboring nodes in an n -cube. A source node can easily decide to perform either an optimal or a suboptimal unicast, based on its safety level, its neighbors' safety levels, and the Hamming distance between the source and the destination. A source node

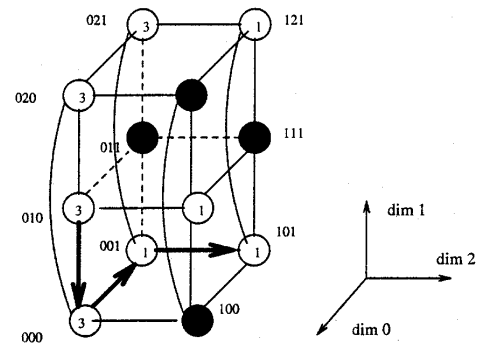


Fig. 5. Routing in a $2 \times 3 \times 2$ GH_3 .

can also identify cases when optimal and suboptimal paths are blocked by faulty nodes and when the corresponding unicast tries to forward a message to another part in a disconnected faulty hypercube. The proposed approach is the first attempt to address reliable unicasting in disconnected hypercubes. We have also extended the safety level concept to cover faulty links and have shown its application in generalized hypercubes.

REFERENCES

- [1] L.N. Bhuyan and D.P. Agrawal, "Generalized Hypercube and Hyperbus Structures for a Computer Network," *IEEE Trans. Computers*, vol. 32, no. 4, pp. 323-333, Apr. 1984.
- [2] M.S. Chen and K.G. Shin, "Adaptive Fault-Tolerant Routing in Hypercube Multicomputers," *IEEE Trans. Computers*, vol. 39, no. 12, pp. 1,406-1,416, Dec. 1990.
- [3] M.S. Chen and K.G. Shin, "Depth-First Search Approach for Fault-Tolerant Routing in Hypercube Multicomputers," *IEEE Trans. Parallel and Distributed Systems*, vol. 1, no. 2, pp. 152-159, Apr. 1990.
- [4] G.M. Chiu and S.P. Wu, "A Fault-Tolerant Routing Strategy in Hypercube Multicomputers," *IEEE Trans. Computers*, vol. 45, no. 2, pp. 143-156, Feb. 1996.
- [5] J.M. Gordon and Q.F. Stout, "Hypercube Message Routing in the Presence of Faults," *Proc. Third Conf. Hypercube Concurrent Computers and Applications*, pp. 251-263, Jan. 1988.
- [6] Y. Lan, "A Fault-Tolerant Routing Algorithm in Hypercubes," *Proc. 1994 Int'l Conf. Parallel Processing*, pp. III 163-III 166, Aug. 1994.
- [7] T.C. Lee and J.P. Hayes, "A Fault-Tolerant Communication Scheme for Hypercube Computers," *IEEE Trans. Computers*, vol. 41, no. 10, pp. 1,242-1,256, Oct. 1992.
- [8] C.S. Raghavendra, P.J. Yang, and S.B. Tien, "Free Dimensions—An Effective Approach to Achieving Fault Tolerance in Hypercubes," *Proc. 22nd Int'l Symp. Fault-Tolerant Computing*, pp. 170-177, 1992.
- [9] J. Wu, "Safety Level—An Efficient Mechanism for Achieving Reliable Broadcasting in Hypercubes," *IEEE Trans. Computers*, vol. 44, no. 5, pp. 702-705, May 1995.
- [10] J. Wu and E.B. Fernandez, "Reliable Broadcasting in Faulty Hypercube Computers," *Microprocessing and Microprogramming*, vol. 39, pp. 43-53, 1993.