Hypercube Network Fault Tolerance: A Probabilistic Approach *

Jianer Chen[†] Iyad A. Kanj[‡] Guojun Wang[§]

Abstract

Extensive experiments and experience have shown that the well-known hypercube networks are highly fault tolerant. On the other hand, most proposed fault tolerance models for hypercube networks are only able to characterize very rare extreme situations thus significantly underestimating the power of hypercube network fault tolerance. In this paper, we develop a new scheme that enables us to derive lower bounds for the probability of hypercube network fault tolerance in terms of node failure probability. Our results provide formal proofs that the hypercube networks can sustain an extremely large number of node failures. Two typical examples are that a 10-cube network of 1024 nodes can sustain up to 10% faulty nodes (i.e., over 100 faulty nodes) while still keeping the non-faulty nodes connected with probability 99%, and that if the failure probability of each individual node is bounded by 0.1%, then all hypercube networks of practical size (e.g., up to a trillion nodes) are able to keep their non-faulty nodes connected with probability 99.9%. A simple routing algorithm, which is local-information-based, is proposed based on this scheme that, with roughly the same high probability, runs in optimal time and constructs a routing path of length bounded by a small constant plus twice the Hamming distance between any two given non-faulty nodes. Our scheme is also applicable to the study of other hierarchical network structures and of other network communication problems.

Keywords. interconnection network, hypercube, routing algorithm, fault tolerance

1 Introduction

With the continuous increasing in network size, dealing with large size networks with faults has become unavoidable. In particular, in a variety of network applications, it has become crucial to make sure that every pair of non-faulty nodes in a network are connected by a fault-free path so that they still can communicate. Let G be a network. The *fault tolerance* of G is defined to be the maximum integer k such that removing any k nodes in G does not disconnect the network [1].

The current paper will focus on the study of fault tolerance of the well-known hypercube networks. Hypercube networks are among the earliest proposed and still remain as one of the most important and attractive network models. In particular, the high symmetry, the strong hierarchical structure, and the maximal fault tolerance are among the most attractive properties

*This work is supported in part by the NSF Grant CCR-0000206, and by the China NNSF for Distinguished Young Scholars and the Changjiang Scholar Reward Project.

†The corresponding author. Department of Computer Science, Texas A&M University, College Station, TX 77843-3112. Email: chen@cs.tamu.edu.

‡School of Computer Science, Telecommunications and Information Systems, DePaul University, 243 S. Wabash Avenue, Chicago, IL 60604. Email: ikanj@cs.depaul.edu.

§College of Information Engineering, Central-South University, ChangSha, Hunan 410083, P. R. China. Email: Gordon434@163.net.

of the hypercube networks [1, 12]. A number of research and commercial hypercube machines have been built in recent years [4, 9, 14].

It is well-known [12, 15] that the fault tolerance of the n-dimensional hypercube network H_n is $n-1$. Indeed, since H_n is n-regular (i.e., each node in H_n has exactly n neighbors), removing the n neighbors of any node u will obviously disconnect u from the other part of the network. In fact, this is the only way to remove n nodes to disconnect H_n [12]. Thus, the fault tolerance $n-1$ of H_n just characterizes this extreme case.

The shortcomings of using the above definition to measure hypercube networks' ability of tolerating faults have been observed. First of all, the probability that the n faulty nodes in H_n are exactly the n neighbors of a non-faulty node is very low. Therefore, the fault tolerance of H_n actually characterizes a very unlikely situation. Secondly, the ratio $(n-1)/2^n$ of the number of faulty nodes to the total number of nodes in H_n is too small to have a practical impact, in the following sense. Suppose each node in the network H_n has an equal failure probability p, then the expected number of faulty nodes in the n-dimensional hypercube network H_n is $2^n p$. However, the fault tolerance $n-1$ of H_n is in general much smaller than $2^n p$ even for a very small value p and a moderate value n. For example, consider $n = 20$ and $p = 0.1\%$. Then $n-1 = 19$, which is too much smaller than $2^n p > 1000$. On the other hand, extensive research experience and experiments have given strong evidence that a hypercube network can tolerate a large number of faulty nodes under any "reasonable" node failure distribution, and that the definition above for the fault tolerance significantly underestimates the power of hypercube networks for tolerating network faults.

Much effort has been devoted that attempts to introduce more realistic definitions to measure networks' ability of tolerating faults. In particular, these studies try to allow the n-dimensional hypercube network to have more than $n-1$ faulty nodes. Several approaches, such as the concept of *forbidden set* [5, 10, 11] and *cluster fault tolerance model* [6, 7], have been proposed for this purpose. The concept of the forbidden sets suggests to prohibit certain failure patterns because "they are very unlikely in practice." In particular, based on the idea of the forbidden sets, the concept of *k-safeness* has been introduced and studied by several researchers [5, 8, 11, 16]. A hypercube network H_n is *k-safe* if each non-faulty node in H_n has at least k non-faulty neighbors.

It is not difficult to see that a k-safe n-dimensional hypercube network cannot tolerate more than $2^k(n-k)-1$ faulty nodes [11, 17]. Thus, for small value k, this still significantly limits the number of faulty nodes in the network. Moreover, so far only for 1-safe and 2-safe hypercube networks, efficient routing algorithms have been studied and developed [5, 8, 16]. Therefore, the study in this line of research still assumes that the number of faulty nodes in the n-dimensional hypercube network H_n is bounded by $O(n)$. On the other hand, a set of $2^k(n-k)$ faulty nodes that keeps k-safeness but disconnects the network H_n again describes a very rare situation: they must be exactly the $2^k(n-k)$ neighbor nodes of a k-dimensional subcube in the hypercube H_n.

Therefore, extensive research and efforts have been made in seeking a "reasonable" fault tolerance model that characterizes "normal" failure situations and avoids being trapped by those unlikely rare situations. A natural approach is to study the probability of connectedness of a network under an assumed probabilistic distribution of individual node failures. However, a direct implementation of this approach seems infeasible since a network may have a very large variety of different kinds of connected components of variant structures. A general probabilistic analysis on these structures seems difficult.

The main contribution of the current paper is to make the above suggested approach possible. We will focus on the uniform and independent probabilistic distribution of individual node failures. Our probabilistic analysis makes use of the concept of "local subcube connectivity" introduced in [2]. The study of local subcube connectivity shows a very nice and important property for the hypercube networks: a properly defined local connectivity of small subcubes implies the global connectivity of the entire network. Since small subcubes have much smaller size, a probabilistic analysis on the local subcube connectivity becomes possible and feasible. From the probability derived for the local subcube connectivity, we then are able to obtain the probability for the global connectivity for the entire hypercube network. Our techniques derive very strong conclusions on the fault tolerance for hypercube networks of practical size. For example, if the failure probability of each individual node is bounded by 10%, then for hypercube networks of up to 1024 nodes, the non-faulty nodes in the network remain connected with probability at least 99%, while if the failure probability of each individual node is bounded by 0.1%, then for hypercube networks of all practical size (e.g., with up to more than a trillion nodes), the non-faulty nodes in the network remain connected with probability at least 99.9%. To the authors' knowledge, these are the first group of formally proven conclusions showing that the hypercube networks are very strongly fault tolerant and can sustain a very large number of faulty nodes with a very high probability.

Another contribution of this paper is the development and probabilistic analysis of a simple routing algorithm on hypercube networks with a large number of faulty nodes. Our routing algorithm is local-information-based and requires no global information for the network status. Taking the advantage of the concept of local subcube connectivity, our routing algorithm is guided by the subcube structures. Because of the small size of the subcubes, our routing algorithm is very efficient and effective. More interestingly, the simpler subcube structures make it possible to apply a probabilistic analysis on the performance of the routing algorithm. In particular, we show that under the same conditions and with roughly the same high probability as we have in the study of the hypercube connectivity, for any two given non-faulty nodes in a hypercube network, our routing algorithm runs in optimal time and constructs a routing path of length bounded by a small constant plus twice the Hamming distance of the two nodes, regardless of the fact that the network may contain a very large number of faulty nodes.

The techniques developed in the current paper provide us with a scheme for probabilistic analysis of faulty tolerance and network routing for hypercube networks. By further investigating the local subcube connectivity for subcubes of different size, we can also trade the probability of network connectivity we can derive with the effectiveness and efficiency of our routing algorithm.

Finally, we point out that our techniques can be extended to other node failure probabilities, to other hierarchical network structures, and to other network applications.

2 The probability of hypercube connectivity

The *n-dimensional hypercube* H_n (or shortly the *n-cube*) consists of 2^n nodes, each is labeled by a distinguished binary string of length n. Two nodes in H_n are adjacent if the binary labels of them differ by exactly one bit. Each binary string $b_1 b_2 \cdots b_{n-k}$ of length $n-k$ corresponds to a

k-dimensional subcube H_k in H_n (or shortly a *k-subcube*) of 2^k nodes whose labels are of the form $b_1 b_2 \cdots b_{n-k} x_{n-k+1} \cdots x_n$, where each x_j is either 0 or 1. The subcube H_k will also be written as $H_k = b_1 b_2 \cdots b_{n-k} **$. It is easy to see that each k-subcube of H_n is isomorphic to the k-cube. Note that there are other subgraphs of H_n that are isomorphic to the k-cube. However, in this paper, we only consider "basic" k-subcubes of the form $b_1 b_2 \cdots b_{n-k} * *$.

The concept of *local k-subcube connectivity* of a hypercube network has been proposed in [2]. We first give a quick review on the related definitions and results.

Definition ([2]) The n-cube H_n is *locally k-subcube connected* if in every k-subcube H_k of H_n, less than half of the nodes in H_k are faulty and the non-faulty nodes of H_k make a connected graph.

The concept of local k-subcube connectivity can be extended in an obvious way to subcubes of the n-cube. In particular, a k-subcube H_k is locally k-subcube connected if the number of faulty nodes in H_k is less than 2^{k-1}, and the non-faulty nodes in H_k make a connected graph.

We point out that the conditions of the local k-subcube connectivity of the n-cube H_n can be detected and maintained in a distributed manner based on localized management: for each k-subcube H_k in H_n, we can set up a managing program, executed on a processor in H_k, which manages and reports the status of the k-subcube H_k.

Theorem 2.1 ([2]) *The non-faulty nodes in a locally k-subcube connected n-cube make a connected graph for all $k \leq n$.*

Theorem 2.1 shows a nice property for the hypercube networks: local connectivity in a hypercube network implies global connectivity of the network. Based on this observation, we are able to derive lower bounds on the probability that an n-cube remains connected when it contains faulty nodes, as follows.

Throughout this paper, we assume that in the n-cube H_n, the node failures are independent, and every node has the same failure probability p. For a given event $\mathbf{E}$, we denote by $\text{Prob}[\mathbf{E}]$ the probability of the event $\mathbf{E}$. We first derive lower bounds, in terms of p and the size 2^n of the n-cube, on the probability that the n-cube H_n is locally k-subcube connected.

Let H_k be a fixed k-subcube of the n-cube H_n. We define the following event:

Event $\mathbf{NL}(H_k)$

The k-subcube H_k is not locally k-subcube connected.

We have the following lemma.

Lemma 2.2 $\text{Prob}[\mathbf{NL}(H_k)] = p^k U_k(p)$, *where*

$$U_k(p) = \sum_{i=k}^{2^{k-1}-1} B_{k,i} p^{i-k}(1-p)^{2^k - i} + \sum_{j=2^{k-1}}^{2^k} \binom{2^k}{j} p^{j-k}(1-p)^{2^k - j}$$

and $B_{k,i}$, $k \leq i \leq 2^{k-1} - 1$, is the number of ways to remove i nodes from the k-subcube H_k so that the remaining nodes in H_k do not make a connected graph.

PROOF. Suppose H_k is not locally k-subcube connected, then one of the following situations must hold: (1) the number i of faulty nodes in H_k is less than 2^{k-1} and the i faulty nodes disconnect the k-subcube H_k; or (2) the number j of faulty nodes in H_k is larger than or equal to 2^{k-1}. Note that since the k-subcube H_k is k-connected [12], when the number i of faulty nodes is less than k, the remaining non-faulty nodes must make a connected graph. Therefore, the probability of the event $\mathbf{NL}(H_k)$ is equal to

$$\sum_{i=k}^{2^{k-1}-1} B_{k,i} p^i (1-p)^{2^k-i} + \sum_{j=2^{k-1}}^{2^k} \binom{2^k}{j} p^j (1-p)^{2^k-j}$$

where $B_{k,i}$, $k \le i \le 2^{k-1}-1$, is the number of ways to remove i nodes from the k-subcube H_k so that the remaining nodes in H_k do not make a connected graph. The lemma follows after a simple rearrangement of the formula. $\square$

Lemma 2.2 implies the following important theorem immediately.

Theorem 2.3 *Suppose that each node in the n-cube H_n has a uniform and independent failure probability p and that $k \le n$. Then the probability that the non-faulty nodes in H_n remain connected is at least $1 - 2^{n-k} p^k U_k(p)$, where $U_k(p)$ is the polynomial of p given in Lemma 2.2.*

PROOF. Let H_k be any k-subcube of the n-cube H_n. By Lemma 2.2, the probability that the k-subcube H_k is not locally k-subcube connected is equal to $p^k U_k(p)$. Thus, the probability that the k-subcube H_k is locally k-subcube connected is equal to $1 - p^k U_k(p)$. In consequence, the probability that the n-cube H_n is locally k-subcube connected, i.e., the probability that all 2^{n-k} k-subcubes in H_n are locally k-subcube connected, is equal to

$$(1 - p^k U_k(p))^{2^{n-k}}$$

which is at least as large as $1 - 2^{n-k} p^k U_k(p)$.

Now the theorem follows since by Theorem 2.1, the local k-subcube connectivity of H_n implies that the non-faulty nodes in H_n are connected. $\square$

It turns out that Theorem 2.3 is very powerful in the discussion of fault tolerance of hypercube networks. Consider the local 3-subcube connectivity. It is not hard to see that $B_{3,3} = 8$ since the only way to remove 3 nodes to disconnect the 3-cube H_3 is to remove the 3 neighbors of a node in H_3, and there are totally 8 nodes in H_3. Therefore, the polynomial $U_3(p)$ is given by

$$\begin{aligned} U_3(p) &= B_{3,3}(1-p)^5 + \sum_{j=4}^{8} \binom{8}{j} p^{j-3} (1-p)^{8-j} \\ &= 8(1-p)^5 + 70p(1-p)^4 + 56p^2(1-p)^3 + 28p^3(1-p)^2 + 8p^4(1-p) + p^5 \end{aligned}$$

Using the standard techniques in calculus, i.e., by computing the zeros of the derivative of $U_3(p)$, we can check that in the interval $[0,1]$, the polynomial $U_3(p)$ is positive and reaches its maximum

$9.881\cdots$ at point $0.14119\cdots$. Now applying Theorem 2.3 enables us to conclude that the probability that the non-faulty nodes in the n-cube H_n remain connected is at least

$$1 - 2^{n-3}p^3U_3(p) \geq 1 - 10 \cdot 2^{n-3}p^3$$

This implies that, for example, if the failure probability of each individual node is bounded by 0.1%, then a 20-cube, which has about a million nodes, can sustain up to $2^{20} \cdot 0.1\% > 1,000$ faulty nodes while still keeping its non-faulty nodes connected with a probability larger than 99.8%. Note that this is a very significant improvement over the traditional definition of fault tolerance, which allows the 20-cube to tolerate at most 19 faulty nodes, and over the concepts of 1-safeness and 2-safeness, which allow the 20-cube to tolerate at most 37 and 71 faulty nodes, respectively.

The above example shows that Theorem 2.3 can be used as a scheme in evaluating the power of hypercubes' fault tolerance. In general, by considering local k-subcube connectivity, we try to derive an upper bound c_k for the polynomial $U_k(p)$ in the interval $[0,1]$, which will immediately make us able to conclude that the probability that the non-faulty nodes of an n-cube H_n, where $n \geq k$, remain connected is at least $1 - c_k 2^{n-k}p^k$. This gives us a potential to derive a larger and larger probability for the connectivity for the n-cube H_n if $c_k 2^{n-k}p^k$ is decreasing when k increases. Unfortunately, even for local 4-subcube connectivity, the analysis for deriving a precise formula for the polynomial $U_4(p)$ becomes extremely tedious. On the other hand, the construction of the polynomial $U_k(p)$ is routine: for each i, $k \leq i \leq 2^{k-1} - 1$, simply enumerate all possible removals of i nodes from H_k, and count the number of removals that disconnect H_k. This will directly give us the number $B_{k,i}$, and hence the polynomial $U_k(p)$. This procedure can obviously be automated and fed to a computer. Based on this idea, we have programmed this procedure for local 5-subcube connectivity and run the program on a parallel Sun workstation system, which gives us the polynomial $U_5(p)$ as follows.

$$\begin{aligned}
U_5(p) \quad = \quad & 32(1-p)^{27} + 832p(1-p)^{26} + 10400p^2(1-p)^{25} + \\
& +83120p^3(1-p)^{24} + 476640p^4(1-p)^{23} + 2086896p^5(1-p)^{22} + \\
& +7251584p^6(1-p)^{21} + 20524992p^7(1-p)^{20} + 48190080p^8(1-p)^{19} + \\
& +95095920p^9(1-p)^{18} + 159252160p^{10}(1-p)^{17} + 601080390p^{11}(1-p)^{16} + \\
& +565722720p^{12}(1-p)^{15} + 471435600p^{13}(1-p)^{14} + 347373600p^{14}(1-p)^{13} + \\
& +225792840p^{15}(1-p)^{12} + 129024480p^{16}(1-p)^{11} + 64512240p^{17}(1-p)^{10} + \\
& +28048800p^{18}(1-p)^{9} + 10518300p^{19}(1-p)^{8} + 3365856p^{20}(1-p)^{7} + \\
& +906192p^{21}(1-p)^{6} + 201376p^{22}(1-p)^{5} + 35960p^{23}(1-p)^{4} + \\
& +4960p^{24}(1-p)^{3} + 496p^{25}(1-p)^{2} + 32p^{26}(1-p) + p^{27}
\end{aligned}$$

Again using the standard techniques in calculus, we verify that the polynomial $U_5(p)$ in the interval $[0,1]$ is positive and bounded by 32 (this now involves solving a polynomial of degree 26, which was also done by a computer). This computation, together with Theorem 2.3, enables us to conclude the following.

$p = 10\%$		$p = 0.1\%$	
dimension n	prob. H_n is connected	dimension n	prob. H_n is connected
5	99.9%	≤ 38	99.9%
6	99.9%	39	99.9%
7	99.9%	40	99.9%
8	99.7%	41	99.8%
9	99.5%	42	99.6%
10	99.0%	43	99.1%

Figure 1: Probability that H_n is connected

Theorem 2.4 *Suppose that each node in the n-cube H_n has a uniform and independent failure probability p and that $n \geq 5$. Then the probability that the non-faulty nodes in H_n remain connected is at least $1 - 32 \cdot 2^{n-5}p^5 = 1 - 2^n p^5$.*

To illustrate the significance of Theorem 2.4, we have computed the probability that the non-faulty nodes in the n-cube H_n make a connected graph for a variety of values of the dimension n, and for $p = 10\%$ and $p = 0.1\%$. The results are given in Figure 1. From the table, we can see that even for very large node failure probability ($p = 10\%$), the hypercube networks of dimension up to 10 remain connected with a very high probability (at least 99%), while if we can bound the node failure probability moderately (say not larger than 0.1%), then all hypercube networks of practical sizes (up to 2^{43} nodes, which is over eight trillion nodes) remain connected with a probability larger than 99%. In particular, for hypercube networks of up to a trillion nodes (i.e., the 40-cube with 2^{40} nodes), the connectedness probability is larger than 99.9%.

To the authors' knowledge, the table in Figure 1 presents the first group of *formally proven* results showing the strong fault tolerance of the hypercube networks, although previous extensive experience and experiments have given people strong impression that the hypercube networks "should be" very highly fault tolerant.

3 The probabilistic analysis for a routing algorithm

One important reason to require the connectedness of non-faulty nodes in a network is to ensure the existence of fault-free routing paths between any two non-faulty nodes in the network. Using the scheme we established in the previous section (Theorem 2.1 and Theorem 2.3), we have been able to show the high probability of connectedness of a hypercube network with faulty nodes, based on its local k-subcube connectivity. In this section, we show that our scheme is also very useful in the analysis of routing algorithms on hypercube networks.

The routing algorithm given in Figure 2 is a revision of the algorithm presented in [2]. The only difference is that in step 3 and step 4 when we search for a path from a node in a k-subcube, we now require explicitly applying the Breadth First Search process (BFS).

Theorem 3.1 *If the input n-cube H_n is locally k-subcube connected, then the algorithm* **H-Router** *constructs a path of non-faulty nodes from u to v.*

Algorithm. H-Router

Input: an n-cube H_n with faults and two non-faulty nodes u and v in H_n

Output: a path of non-faulty nodes in H_n that connects u and v

1. let $u = u_1 u_2 \cdots u_n$ and $v = v_1 v_2 \cdots v_n$, where u_i and v_i are binary bits;
2. $w = u$, and initialize the path $P = [w]$;
3. **for** $i = 1$ **to** $n - k$ **do**

   {at this point, we have, inductively, constructed a path P of non-faulty nodes from $u = u_1 u_2 \cdots u_n$ to $w = w_1 w_2 \cdots w_n$, where $w_j = v_j$ for $j = 1, 2, \ldots, i-1$, and $w_j = u_j$ for $j = i, i+1, \ldots, n-k$. }

   if $w_i \neq v_i$ {so $v_i = \overline{w_i}$} **then**

   apply BFS in the k-subcube $w_1 w_2 \cdots w_{n-k} **$, starting from the node w, to find two adjacent non-faulty nodes $w' = w_1 \cdots w_{i-1} w_i w_{i+1} \cdots w_{n-k} x_{n-k+1} \cdots x_n$ and $v' = w_1 \cdots w_{i-1} v_i w_{i+1} \cdots w_{n-k} x_{n-k+1} \cdots x_n$;

   extend the path P from w to w' along the BFS tree, then to v';

   let $w = v'$;
4. {at this point, we have $w_j = v_j$ for all $j = 1, 2, \ldots, n-k$}

   apply BFS to extend the path P from w to v in the k-subcube $v_1 v_2 \cdots v_{n-k} **$.

{**Remark.** if at any point the algorithm could not proceed, then **stop:** the n-cube H_n is not locally k-subcube-connected. }

Figure 2: A routing algorithm for hypercube networks

PROOF. The constructed path P from the source node $u = u_1 u_2 \cdots u_n$ to the destination node $v = v_1 v_2 \cdots v_n$ starts with a single node u. Inductively, assume that before the ith execution of the loop in step 3, the path P of non-faulty nodes is from the source node u to a node $w = w_1 w_2 \cdots w_n$ such that $w_j = v_j$ for $j = 1, 2, \ldots, i-1$ and $w_j = u_j$ for $j = i, i+1, \ldots, n-k$. In the ith execution of the loop in step 3, the algorithm "converts" the ith bit w_i of w into v_i. If the ith bits of w and v are identical, then we do not have to do anything. Otherwise, consider the two k-subcubes $H_k^0 = w_1 \cdots w_{i-1} w_i w_{i+1} \cdots w_{n-k} **$ and $H_k^1 = w_1 \cdots w_{i-1} v_i w_{i+1} \cdots w_{n-k} **$ (note that $v_i = \overline{w_i}$). Since the n-cube H_n is locally k-subcube connected, each of H_k^0 and H_k^1 has less than half its nodes faulty. Thus, we must be able to find two adjacent non-faulty nodes, $w' = w_1 \cdots w_{i-1} w_i w_{i+1} \cdots w_{n-k} x_{n-k+1} \cdots x_n$ and $v' = w_1 \cdots w_{i-1} v_i w_{i+1} \cdots w_{n-k} x_{n-k+1} \cdots x_n$, in H_k^0 and H_k^1, respectively. Again because of the local k-subcube connectivity of H_n, the non-faulty nodes in the k-subcube H_k^0 are connected. Thus, the Breadth First Search process in H_k^0 starting from the non-faulty node w must reach the non-faulty node w'. Thus, the path P can now be extended in the k-subcube H_k^0 from w to w'. Moreover, since the nodes w' and v' are adjacent, the path P can be further extended from w' to v'. We then let $w = v'$. Note that now the node w has its first i bits identical to that of the destination node v. Moreover, the new node $w = v'$ is in the k-subcube $H_k^1 = w_1 \cdots w_{i-1} v_i w_{i+1} \cdots w_{n-k} **$, which, by induction, also has its $(i+1)$st, $(i+2)$nd, ..., and $(n-k)$th bits identical to that of the source node u. Therefore, the induction goes through.

After the $n-k$ executions of the loop, the node w has its first $n-k$ bits identical to that of v. Thus, the nodes w and v now are in the same k-subcube $H_k = v_1 v_2 \cdots v_{n-k} **$ of H_n. Since the non-faulty nodes in H_k are connected, we can spend another Breadth First Search, in the k-subcube H_k, to further extend the path P from w to v, and complete a path of non-faulty nodes in H_n from u to v.

In conclusion, if the input n-cube H_n is locally k-subcube connected, then the algorithm **H-Router** constructs a path of non-faulty nodes from u to v. □

We point out that the algorithm **H-Router** is distributed and local-information-based, in the following sense. No global information about the network H_n is required by the algorithm, i.e., the algorithm does not require any information in advance about the faulty nodes. The only thing we assume is that from each non-faulty node y, our algorithm can request the status of the neighbors of y. Step 3 of the algorithm may need a little more explanation. When we look for a non-faulty pair $\{w', v'\}$ in step 3, we actually only search for non-faulty nodes in the k-subcube $H_k^0 = w_1 \cdots w_{i-1} w_i w_{i+1} \cdots w_{n-k} **$, based only on the neighbors' status. For each non-faulty node y we reach in H_k^0, we immediately check its neighbor y' in the k-subcube $H_k^1 = w_1 \cdots w_{i-1} v_i w_{i+1} \cdots w_{n-k} **$. Therefore, the non-faulty pair $\{w', v'\}$ in the algorithm can be found using only local information.

Before the ith execution of the loop body in step 3, we have $w_i = u_i$. Thus, the loop body is executed only when $u_i \neq v_i$. Let $H_{n-k}(u, v)$ be the Hamming distance between the substrings $u_1 u_2 \cdots u_{n-k}$ and $v_1 v_2 \cdots v_{n-k}$, i.e., $H_{n-k}(u, v)$ is the number of bit positions at which the substrings $u_1 u_2 \cdots u_{n-k}$ and $v_1 v_2 \cdots v_{n-k}$ differ. Then the loop body in step 3 is executed exactly $H_{n-k}(u, v)$ times. During each execution of the loop body, the Breadth First Search in a k-subcube H_k takes time $O(2^k)$ and increases the path length by at most 2^k. Therefore, for the two non-faulty nodes u and v in H_n, the running time of the algorithm **H-Router** is bounded by $O(2^k(H_{n-k}(u, v) + 1))$ and the constructed routing path has length bounded by $2^k(H_{n-k}(u, v) + 1)$. This gives the bounds in the worst case. In the following, we show that with very high probability, both the running time of the algorithm and the routing path length are much better than their worst case bounds.

We say that two k-subcubes $b_1 b_2 \cdots b_{n-k} **$ and $b'_1 b'_2 \cdots b'_{n-k} **$ in the n-cube H_n are *neighboring k-subcubes* if the two binary strings $b_1 b_2 \cdots b_{n-k}$ and $b'_1 b'_2 \cdots b'_{n-k}$ differ by exactly one bit. Note that each k-subcube in the n-cube H_n has exactly $n-k$ neighboring k-subcubes.

Let w be a node in a k-subcube H_k and let H'_k be a neighboring k-subcube of H_k. We say that *the k-subcube H'_k is not reachable from w in two steps* if every path in $H_k \cup H'_k$ of length bounded by 2 from w to a node in H'_k contains at least one faulty node $u \neq w$ (note that the node w could be either faulty or non-faulty). We define the following event.

Event L2(H_k)

There is a node w in the k-subcube H_k and there is a neighboring k-subcube H'_k of H_k such that the k-subcube H'_k is not reachable from w in two steps.

Lemma 3.2 Prob[**L2**(H_k)] *is bounded by* $4^k(n-k)p^{k+1}$.

PROOF. Let w be any node in the k-subcube H_k and let H'_k be any neighboring k-subcube of H_k.

Let u_1, u_2, ..., u_k be the k neighbors of the node w in the k-subcube H_k, and w' be the node in the k-subcube H'_k that is adjacent to w. Finally, let u'_1, u'_2, ..., u'_k be the nodes in the k-subcube H'_k that are adjacent to the nodes u_1, u_2, ..., u_k in H_k, respectively. Since each node in H_k has a unique neighbor in H'_k, a path of length at most 2 in $H_k \cup H'_k$ from w to H'_k must either go directly from the node w to its neighbor w' in H'_k, or go from w to a neighbor u_j of w in H_k then to u_j's neighbor u'_j in H'_k, where $j = 1, 2, \ldots, k$. Therefore, if the k-subcube H'_k is not reachable from w in two steps, then we must have the following conditions: (1) the node w' is faulty; and (2) each $\{u_j, u'_j\}$ of the k pairs of nodes, $j = 1, 2, \ldots, k$, where u_j is a neighbor of w in H_k and u'_j is u_j's neighbor in H'_k, must contain at least one faulty node. Note that the probability that the pair $\{u_j, u'_j\}$ contains at least one faulty node is $1 - (1-p)^2 = 2p - p^2$. Therefore, the probability that the k-subcube H'_k is not reachable from w in two steps is bounded by

$$p(2p - p^2)^k = p^{k+1}(2-p)^k$$

Since the k-subcube H_k has exactly $n-k$ neighboring k-subcubes, the probability that there is a neighboring k-subcube H'_k of H_k such that H'_k is not reachable from w in two steps is bounded by

$$(n-k)p^{k+1}(2-p)^k$$

Finally, since there are 2^k nodes in the k-subcube H_k, the probability that there exist a node u in H_k and a neighboring k-subcube H'_k of H_k such that H'_k is not reachable from u in two steps, i.e., the probability of the event $\mathbf{L2}(H_k)$, is bounded by

$$2^k(n-k)p^{k+1}(2-p)^k \leq 4^k(n-k)p^{k+1}$$

This proves the lemma. □

Now we consider the following event.

Event $\mathbf{FN}(H_k)$

The k-subcube H_k has at least $2k-2$ faulty nodes.

We have the following lemma.

Lemma 3.3 Prob$[\mathbf{FN}(H_k)]$ *is bounded by* $p^{2k-2}\binom{2^k}{2k-2}$.

PROOF. Since the k-subcube H_k has totally 2^k nodes, the probability that H_k has exactly i faulty nodes is $\binom{2^k}{i}p^i(1-p)^{2^k-i}$. Thus, the probability that H_k has at least $2k-2$ faulty nodes is

$$\sum_{i=2k-2}^{2^k} \binom{2^k}{i} p^i (1-p)^{2^k-i}$$

which is bounded by $\binom{2^k}{2k-2}p^{2k-2}$ (for a proof of this inequality, see, for example, [3], page 121). □

Let $u = u_1u_2\cdots u_n$ and $v = v_1v_2\cdots v_n$ be any two nodes in the n-cube H_n. Recall that $H_{n-k}(u,v)$ is the Hamming distance between the two substrings $u_1u_2\cdots u_{n-k}$ and $v_1v_2\cdots v_{n-k}$. We have the following theorem.

Theorem 3.4 *Suppose that each node in the n-cube H_n has a uniform and independent failure probability p, then for any $k \le n$ and for any two given non-faulty nodes u and v in H_n, with probability at least $1 - 2^{n-k}p^k[U_k(p) + 4^k(n-k)p + p^{k-2}\binom{2^k}{2k-2}]$, where the polynomial $U_k(p)$ is given in Lemma 2.2, the algorithm* **H-Router** *runs in time $O(kn + 2^k)$ and constructs a routing path from u to v of length bounded by $2H_{n-k}(u, v) + k + 2$.*

PROOF. Let H_k be a fixed k-subcube in the n-cube H_n. Recall that $\mathbf{NL}(H_k)$ is the event that H_k is not locally k-subcube connected, $\mathbf{L2}(H_k)$ is the event that there exist a node w in H_k and a neighboring k-subcube H_k' of H_k such that H_k' is not reachable from w in two steps, and $\mathbf{FN}(H_k)$ is the event that H_k has at least $2k-2$ faulty nodes. Now let $\mathbf{Bad}(H_k)$ be the union of these events:

$$\mathbf{Bad}(H_k) = \mathbf{NL}(H_k) \cup \mathbf{L2}(H_k) \cup \mathbf{FN}(H_k)$$

then we have

$$\begin{aligned}
\mathrm{Prob}[\mathbf{Bad}(H_k)] &\le \mathrm{Prob}[\mathbf{NL}(H_k)] + \mathrm{Prob}[\mathbf{L2}(H_k)] + \mathrm{Prob}[\mathbf{FN}(H_k)] \\
&\le p^k U_k(p) + 4^k(n-k)p^{k+1} + p^{2k-2}\binom{2^k}{2k-2} \\
&= p^k \left[U_k(p) + 4^k(n-k)p + p^{k-2}\binom{2^k}{2k-2} \right] \qquad (1)
\end{aligned}$$

where the second inequality is by Lemma 2.2, Lemma 3.2, and Lemma 3.3, and the polynomial $U_k(p)$ is given in Lemma 2.2.

Let the complement event of the event $\mathbf{Bad}(H_k)$ be $\mathbf{Good}(H_k)$, which is stated as:

Event Good(H_k)

(1) the k-subcube H_k is locally k-subcube connected; (2) for every node w in H_k, every neighboring k-subcube H_k' of H_k is reachable from w in two steps; and (3) the number of faulty nodes in H_k is bounded by $2k-3$.

Then the probability of the event $\mathbf{Good}(H_k)$ is

$$\mathrm{Prob}[\mathbf{Good}(H_k)] = 1 - \mathrm{Prob}[\mathbf{Bad}(H_k)]$$

Now we define **GOOD** to be the intersection of the events $\mathbf{Good}(H_k)$ over all k-subcubes H_k in the n-cube H_n, thus

Event GOOD

For every k-subcube H_k in the n-cube H_n: (1) H_k is locally k-subcube connected; (2) for every node w in H_k, every neighboring k-subcube H_k' of H_k is reachable from w in two steps; and (3) the number of faulty nodes in H_k is bounded by $2k-3$.

Since there are totally 2^{n-k} k-subcubes in the n-cube H_n, the node failure probability is independent, and any two k-subcubes in H_n are disjoint (thus the events $\mathbf{Good}(H_k)$ and $\mathbf{Good}(H_k')$ are independent for two different k-subcubes H_k and H'_k), we have

$$\begin{aligned} \text{Prob}[\mathbf{GOOD}] &= (\text{Prob}[\mathbf{GOOD}(H_k)])^{2^{n-k}} \\ &= (1 - \text{Prob}[\mathbf{Bad}(H_k)])^{2^{n-k}} \\ &\geq 1 - 2^{n-k}\text{Prob}[\mathbf{Bad}(H_k)] \end{aligned} \tag{2}$$

Combining (1) and (2), we get immediately

$$\text{Prob}[\mathbf{GOOD}] \geq 1 - 2^{n-k}p^k \left[U_k(p) + 4^k(n-k)p + p^{k-2}\binom{2^k}{2k-2} \right] \tag{3}$$

Therefore, it suffices to show that under the conditions of the event **GOOD**, the algorithm **H-Router** achieves the performance described in the theorem.

Under the conditions of the event **GOOD**, every k-subcube in H_n is locally k-subcube connected. That is, the n-cube H_n is locally k-subcube connected. By Theorem 3.1, the algorithm **H-Router** constructs a routing path P connecting the two given non-faulty nodes $u = u_1u_2 \cdots u_n$ and $v = v_1v_2 \cdots v_n$.

Now consider step 3 of the algorithm. In case the ith bit w_i of the node $w = w_1w_2 \cdots w_n$ (note that $w_i = u_i$) is not equal to the ith bit v_i of the destination node $v = v_1v_2 \cdots v_n$, we look for a subpath from the node w in the k-subcube $H_k^0 = w_1 \cdots w_{i-1}w_iw_{i+1} \cdots w_{n-k} * *$ to the neighboring k-subcube $H_k^1 = w_1 \cdots w_{i-1}v_iw_{i+1} \cdots w_{n-k} * *$. Under the conditions of the event **GOOD**, the neighboring k-subcube H_k^1 of the k-subcube H_k^0 is reachable from the node w in H_k^0 in two steps. That is, either w's neighbor in H_k^1 is non-faulty, or a neighbor w' of w in H_k^0 and w's neighbor v' in H_k^1 are both non-faulty. Since the algorithm **H-Router** uses Breadth First Search from w in H_k^0, in time $O(k)$ the path P should be extended to H_k^1 by adding at most two edges. Finally, since the loop body in step 3 is executed only when $u_i \neq v_i$, we conclude that the length of the path P after step 3 of the algorithm is bounded by $2H_{n-k}(u, v)$. Moreover, the running time of step 3 is bounded by $O(kH_{n-k}(u, v) + (n-k)) = O(kn)$.

Finally, we consider step 4 of the algorithm. Now the node w and the destination node v are in the same k-subcube H_k. Under the conditions of the event **GOOD**, the k-subcube H_k is locally k-subcube connected and the number of faulty nodes in H_k is bounded by $2k - 3$. Since the non-faulty nodes in H_k are connected, every non-faulty node in H_k has at least one non-faulty neighbor in H_k. Latifi [10] has shown that if each non-faulty node in H_k has at least one non-faulty neighbor and the total number of faulty nodes in H_k is bounded by $2k - 3$, then every pair of non-faulty nodes in H_k are connected by a path of length bounded by $k + 2$ that consists of only non-faulty nodes. Therefore, the Breadth First Search process from the node w will construct a path from w to v in H_k whose length is bounded by $k + 2$. This completes the routing path from the source node u to the destination node v in the n-cube H_n. Moreover, the Breadth First Search in step 4 of the algorithm runs in time $O(2^k)$.

Therefore, under the conditions of event **GOOD**, whose probability is at least $1 - 2^{n-k}p^k[U_k(p) + 4^k(n-k)p + p^{k-2}\binom{2^k}{2k-2}]$, the routing algorithm **H-Router** runs in time $O(kn + 2^k)$ and constructs a routing path of length bounded by $2H_{n-k}(u, v) + k + 2$. $\square$

dimension n	prob. **H-Router** performs well
≤ 38	99.9%
39	99.9%
40	99.8%
41	99.5%
42	99.0%
43	98.1%

Figure 3: The probability that the algorithm **H-Router** runs in time $O(n)$ and constructs a routing path of length bounded by $2H_{n-5}(u, v) + 7$ (assuming the node failure probability $p = 0.1\%$).

Remark. The running time of the algorithm **H-Router** in Theorem 3.4 can be further improved to $O(kn + k^2)$. In fact, Latifi [10] has described a construction that runs in time $O(k^2)$ and constructs a routing path of length bounded by $k + 2$ for any two non-faulty nodes in a k-cube, under the conditions that each non-faulty node has at least one non-faulty neighbor and that the total number of faulty nodes in H_k is bounded by $2k - 3$. In consequence, the running time of step 4 of the algorithm **H-Router** can be reduced to $O(k^2)$, which gives the improved running time $O(kn + k^2)$ for the algorithm **H-Router**.

Corollary 3.5 *Suppose that each node in the n-cube H_n has a uniform and independent failure probability p, then for any two given non-faulty nodes u and v in H_n, with probability at least $1 - 2^{n-5}p^5[32 + 1024(n - 5)p + 10518300p^3]$, the algorithm* **H-Router** *runs in time $O(n)$ and constructs a routing path from u to v of length bounded by $2H_{n-5}(u, v) + 7$.*

PROOF. Let $k = 5$ in Theorem 3.4, we conclude that with probability at least $1 - 2^{n-5}p^5[U_5(p) + 4^5(n-5)p + p^3\binom{32}{8}]$, the routing algorithm **H-Router** constructs in time $O(n)$ a routing path from u to v of length bounded by $2H_{n-5}(u, v) + 5 + 2 = 2H_{n-5}(u, v) + 7$, where the polynomial $U_5(p)$ is given in Lemma 2.2.

As we have verified in section 2, in the interval $[0, 1]$, the polynomial $U_5(p)$ is bounded by 32. This, plus $\binom{32}{8} = 10518300$, proves the corollary. $\square$

To illustrate the significance of Corollary 3.5, we consider the case when $p = 0.1\%$, and calculate the probability that the routing algorithm **H-Router** runs in time $O(n)$ and constructs a routing path of length bounded by $2H_{n-5}(u, v) + 7$ from u to v. The results are given in Figure 3.

From the table in Figure 3 (compare it with the table in Figure 1), we can see that for hypercube networks of dimension up to 43, the routing algorithm **H-Router** has very good performance with probability larger than 98%. In particular, for n-dimensional hypercube networks with $n \leq 39$, the algorithm achieves the performance with probability larger than 99.9%. Note that the 39-dimensional hypercube network has more than half trillion nodes, which seems to have covered all practical cases.

4 Conclusions and final remarks

Fault tolerance and network routing have been among the most studied topics in the research of parallel processing and computer networking. In this paper, we have established a scheme that enables us to study the probability of network fault tolerance in terms of individual node failure probability. Our results are both theoretically significant and practically important. From a theoretical point of view, the scheme (see Theorem 2.3) offers very general and powerful techniques for establishing lower bounds on the probability for network connectedness, while from the practical point of view, our scheme has provided (see the tables in Figure 1 and Figure 3) formally proven threshold on the node failure probability that guarantees very high probability for network connectedness and efficiency and effectiveness of our routing algorithm. Before we close this paper, we would like to make a few remarks.

The scheme established in this paper is obviously not only restricted to hypercube networks. In fact, the technique developed here is very general and can be applied to any hierarchical network structures (i.e., the network structures in which larger networks can be decomposed into smaller sub-networks of similar structure). Examples of hierarchical networks include mesh networks, a variety of hypercube variations, and many network structures based on Cayley graphs (see [1, 12, 15] for a survey). Roughly speaking, the scheme suggests that for hierarchical network structures, the study of global connectivity and routings can be reduced to the study of smaller substructures. In most cases, the study of small substructures is much easier since they have much fewer nodes and simpler structure. Sometimes certain properties of the small substructures can even be obtained using exhaustive enumerations.

Our scheme should also suggest a possible approach for the study of network fault tolerance under other probability distributions of node failures. For example, clustered node failure distributions have also become very popular under the belief that a node tends to fail when many of its neighbors have failed. We believe that based on our approach, the probabilistic study of network fault tolerance in terms of this kind of node failure distributions should become possible. Other extensions of our scheme include the study of other network communication problems, such as broadcasting, multicasting, and parallel routing.

We would also like to remark on our scheme for more technical details. To make our discussion more specific, we will concentrate on hypercube networks. However, the readers are reminded that most of these discussions are also applicable to other hierarchical network structures.

Theorem 2.4 could be further improved. In fact, since every locally k-subcube connected n-cube is also locally $(k+1)$-subcube connected [2], under the same uniform and independent node failure distribution, the probability that an n-cube is locally $(k+1)$-subcube connected is at least as large as the probability that the n-cube is locally k-subcube connected. Therefore, deriving the probability for, say, local 6-subcube connectivity will certainly improve the bound given in Theorem 2.4. Therefore, Theorem 2.4 is really an example of an application of Theorem 2.3, which offers a very general scheme for the study of hypercube fault tolerance. On the other hand, it does require to overcome certain analysis and computation difficulties in order to improve Theorem 2.4: when k increases, the construction of the polynomial $U_k(p)$ quickly becomes computationally infeasible even with a high speed computer. For example, to construct the polynomial $U_6(p)$ using a routine method, we need to enumerate about half of the $2^{2^6} > 10^{19}$ subsets of nodes in the 6-cube. For further larger k, this enumeration quickly becomes infeasible. Better and deeper mathematical analysis seems needed for deriving and estimating the value for the polynomial $U_k(p)$.

There is another reason that we may not want to consider local k-subcube connectivity for large k. Consider Theorem 3.4, the running time of the algorithm **H-Router** is bounded by $O(kn + 2^k)$, which will become inefficient when k is very large. Indeed, the local n-subcube connectivity of an n-cube basically just tells that the non-faulty nodes in the n-cube are connected without giving any other useful information. Thus, what a routing algorithm can do is to exhaustively search among the non-faulty nodes, which obviously takes time $O(2^n)$. Note that Latifi's method [10], which routes in a k-subcube in time $O(k^2)$, cannot help for large k either since it requires the number of faulty nodes to be bounded by $2k - 3$. When k is large, the ratio $(2k - 3)/2^k$ will require an impractically small bound for individual node failure probability p.

We should mention that Najjar and Gaudiot [13] have studied the probability of fault tolerance of regular networks based on uniform and independent node failures. Let Q be an n-regular network of N nodes. Suppose that Q contains faulty nodes. An *h-cluster* is a connected component of h nodes in Q after all faulty nodes are removed. Najjar and Gaudiot [13] conjectured that the probability of network disconnectedness caused by 1-clusters should be much larger than the probability of network disconnectedness caused by h-clusters for any $h > 1$. Based on this conjecture, they studied the probability of 1-clusters and used it to approximate the probability of network disconnectedness. They also verified their conjecture based on a variety of assumptions, in particular, on the assumptions that the number of possible h-clusters is bounded by $O(N)$ and that the number of faulty neighbors of each h-cluster is bounded by $O(n)$. Unfortunately, their approach seems not applicable to the case of hypercube networks and most hierarchical network structures. Take hypercube networks as an example, even the number of 2-clusters, which are edges, in an n-cube is already $O(nN) = O(2^n n)$. It can also be seen easily that when h gets larger, the number of possible h-clusters in the n-cube can be much larger than $N = 2^n$ while the number of faulty neighbors of an h-cluster can be much larger than n. Moreover, the analysis in [13] provides no guideline on how a routing algorithm can be developed when node failures occur in a network.

Finally, we would like to explain the advantage of our scheme over the previously proposed fault tolerance models. Take the k-safeness as an example. The model of k-safeness requires that each non-faulty node have at least k non-faulty neighbors. Using the techniques developed in this paper, it is also possible to prove that for small k, a hypercube network is k-safe with high probability. However, the k-safeness of the hypercube does not guarantee the connectedness of the network unless we also bound the number of faulty nodes by $2^k(n-k) - 1$ [11, 17]. This constraint then significantly bounds the node failure probability to the order $2^k(n - k)/2^n$, which would be impractically small even for a reasonable value k and a moderate value n. On the other hand, our scheme allows a much larger percentage of faulty-nodes and guarantees the connectedness of the entire network.

References

[1] S. B. Akers and B. Krishnamurthy, A group-theoretic model for symmetric intercon-

nection networks, *IEEE Transactions on Computers 38*, (1989), pp. 555-565.

[2] J. Chen, G. Wang, and S. Chen, Routing in hypercube networks with a constant fraction of faulty nodes, *Proc. 4th International Conference on Algorithms and Architecture for Parallel Processing* (ICA3PP'2000), (2000), pp. 605-616.

[3] T. H. Cormen, C. E. Leiserson, and R. L. Rivest, *Introduction to Algorithms*, McGraw-Hill Book Company, New York, 1992.

[4] A. L. DeCegama, *The Technology of Parallel Processing – Parallel Processing Architectures and VLSI Hardware*, Prentice Hall, 1989.

[5] A. H. Esfahanian, Generalized measures of fault tolerance with application to n-cube networks, *IEEE Transactions on Computers 38*, (1989), pp. 1586-1591.

[6] Q.-P. Gu and S. Peng, Optimal algorithms for node-to-node fault tolerant routing in hypercubes, *The Computer Journal 39*, (1996), pp. 626-629.

[7] Q.-P. Gu and S. Peng, k-pairwise cluster fault tolerant routing in hypercubes, *IEEE Transactions on Computers 46*, (1997), pp. 1042-1049.

[8] Q.-P. Gu and S. Peng, Unicast in hypercubes with large number of faulty nodes, *IEEE Transactions on Parallel and Distributed Systems 10*, (1999), pp. 964-975.

[9] Intel Corp., *Intel iPSC System Overview*, Jan. 1986.

[10] S. Latifi, Combinatorial analysis of the fault diameter of the n-cube, *IEEE Transactions on Computers 42*, (1993), pp. 27-33.

[11] S. Latifi, M. Hedge, and M. Naraghi-Pour, Conditional connectivity measures for large multiprocessor systems, *IEEE Transactions on Computers 43*, (1994), pp. 218-222.

[12] F. T. Leighton, *Introduction to Parallel Algorithms and Architectures: Arrays, Trees, Hypercubes* , Morgan Kaufmann, San Mateo, CA, 1992.

[13] W. Najjar and J. L. Gaudiot, Network resilience: a measure of network fault tolerance, *IEEE Transactions on Computers 39*, (1990), pp. 174-181.

[14] NCUBE Corp., *NCUBE/ten: An Overview*, Beaverton, OR, Nov. 1985.

[15] Y. Saad and M. H. Schultz, Topological properties of hypercubes, *IEEE Transactions on Computers 37*, (1988), pp. 867-872.

[16] S. B. Tien and C. S. Raghavendra, Algorithms and bounds for shortest paths and diameter in faulty hypercubes, *IEEE Transactions on Parallel and Distributed Systems 4*, (1993), pp. 713-718.

[17] J. Wu and G. Guo, Fault tolerance measures for m-ary n-dimensional hypercubes based on forbidden faulty sets, *IEEE Transactions on Computers 47*, (1998), pp. 888-893.