

Hypercube Network Fault Tolerance: A Probabilistic Approach

JIANER CHEN

Department of Computer Science, Texas A&M University
College Station, TX 77843-3112, and
College of Information Science and Engineering,
Central-South University,
ChangSha, Hunan 410083, P.R. China.
Email: chen@cs.tamu.edu

IYAD A. KANJ*

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

GUOJUN WANG

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

Received (received date)

Revised (revised date)

*The corresponding author.

Extensive experiments and experience have shown that the well-known hypercube networks are highly fault tolerant. What is frustrating is that it seems very difficult to properly formulate and formally prove this important fact, despite extensive research efforts in the past two decades. Most proposed fault tolerance models for hypercube networks are only able to characterize very rare extreme situations thus significantly underestimating the fault tolerance power of hypercube networks, while for more realistic fault tolerance models, the analysis becomes much more complicated. In this paper, we develop new techniques that enable us to analyze a more realistic fault tolerance model and derive lower bounds for the probability of hypercube network fault tolerance in terms of node failure probability. Our results are both theoretically significant and practically important. From the theoretical point of view, our method offers very general and powerful techniques for formally proving lower bounds on the probability of network connectivity, while from the practical point of view, our results provide formally proven and precisely given upper bounds on node failure probabilities for manufacturers to achieve a desired probability for network connectivity. Our techniques are also useful and powerful for analysis of the performance of routing algorithms, and applicable to the study of other hierarchical network structures and to other network communication problems.

Keywords: parallel processing, interconnection network, hypercube, routing algorithm, fault tolerance.

1. Introduction

With the continuous increase in network size, dealing with large size networks with faults has become unavoidable [23, 24]. In particular, the problem of keeping all non-faulty nodes in a network connected and developing efficient and reliable routing algorithms in a network with faults has been extensively studied in the last two decades (e.g. [17, 21, 25]).

The current paper will focus on the study of fault tolerance of the well-known hypercube networks. Hypercube networks are among the earliest proposed network models and still remain as one of the most important and attractive ones. In particular, the high symmetry, the strong hierarchical structure, and the maximal fault tolerance are among the most attractive properties of hypercube networks [2, 18]. A number of research and commercial large-scale parallel machines have been built based on the hypercube network topology [14, 22, 28, 29, 31]. Very recently, research has also shown that fault tolerant hypercube networks of large size (e.g., over many thousand nodes) can be used as an effective control topology in supporting large-scale multicast applications in the Internet [19, 20].

Extensive experiments and experience have shown that hypercube networks can tolerate a large number of faulty nodes while still remain functioning. What is frustrating is that it seems very difficult to properly formulate and formally prove this important fact, despite extensive research efforts in the past two decades (see [9, 15] for comprehensive surveys). It is easy to see that n faulty nodes can disconnect the n -dimensional hypercube network H_n — when all n neighbors of a non-faulty node become faulty. Thus, the network H_n can tolerate no more than $n - 1$ faulty nodes in this case. However, this is the only way that n faulty nodes disconnect H_n [18, 27] and in practice the situation is very unlikely. Moreover, the ratio $(n - 1)/2^n$ of faulty nodes over the total nodes in H_n is impractically too small, which requires, for example for $n = 20$, the average failure probability of each individual node to be not larger than 0.002%. Much effort has been devoted attempting to introduce more realistic definitions to measure hypercube networks' ability of tolerating faults. Concepts such as *forbidden set* [7, 16, 17] and *cluster fault tolerance model* [10, 11] have been proposed. The concept of forbidden sets suggests to prohibit certain "very unlikely" failure patterns. A special model for forbidden sets, the *k-safeness* of networks, insists that each non-faulty node has at least k non-faulty neighbors [7, 12, 17, 32]. However, routing algorithms on the model of k -safe networks seem to become more complicated, and efficient routing algorithms have only been developed for 1-safe and 2-safe hypercube networks [7, 12, 32]. Since in the worst case a k -safe hypercube network H_n can tolerate no more than $2^k(n - k) - 1$ faulty nodes [17, 35], this line of research still assumes a bound $O(n)$ on the number of faulty nodes in H_n .

An alternative approach, which seems to more reasonably characterize "normal" failure situations and avoid being trapped by the unlikely rare situations, is to study the probability of connectivity of a network under an assumed probabilistic

distribution of individual node failures. Najjar and Gaudiot [21] have studied this model using the following approach. Let Q be an N node network in which all nodes have degree n . An h -cluster in Q is a connected subgraph of h nodes in Q . We say that a disconnection is *caused by an h -cluster C* if all nodes in C are non-faulty but all nodes adjacent to C are faulty. Najjar and Gaudiot [21] conjectured that the probability of network disconnection caused by 1-clusters should be much larger than that caused by other situations. Based on this conjecture, they studied the probability of disconnection caused by 1-clusters and used it to approximate the probability of real network disconnection. They also verified their conjecture based on a variety of assumptions, including that the number of possible h -clusters in Q must be bounded by $O(N)$ and that the number of nodes adjacent to an h -cluster must be bounded by $O(n)$. Obviously, hypercube networks (and actually most hierarchical networks) do not satisfy these assumptions: an n -dimensional hypercube H_n has $N = 2^n$ nodes and each node in H_n has degree n , while the number of edges in H_n , each of which makes a 2-cluster, is $nN/2$ (thus not $O(N)$), and the number of nodes in H_n adjacent to a k -dimensional subcube, which makes a 2^k -cluster, is $2^k(n - k)$ (thus can be much larger than $O(n)$).

Therefore, the conjecture given in [21] was not firmly verified and formally proved for hypercube networks, and an approximation of the probability of connectivity derived from this conjecture for hypercube networks will not be convincing. On the other hand, a precise calculation or a good approximation of the connectivity probability for hypercube networks seems to be very difficult. The hypercube networks may have a very large variety of different kinds of connected subgraphs of variant structures. A good approximation of the probability would require an effective characterization of these subgraphs, while it is known that even identifying a single connected subgraph of a hypercube network is already NP-hard [34]. The fact that no further improvements have been made over the results in [21] in the last decade also illustrates the difficulty of this approach.

The main contribution of the current paper is the development of systematic and powerful techniques for formal analysis of the above probabilistic fault tolerance model, which enables us to develop effective lower bounds for connectivity probability for hypercube networks. Our techniques make use of the concept of “local subcube connectivity” introduced in [3]. 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 effective lower bounds on the probability for the global connectivity for the entire hypercube network. Our results are both theoretically significant and practically important. From the theoretical point of view, our method offers very general and powerful techniques for formally proving lower bounds on the probability for network con-

nectivity, while from the practical point of view, our results provide formally proven and precisely given upper bounds on node failure probability for manufacturers to achieve a desired probability for the entire network connectivity. For example, our techniques give *formal proofs* that as long as the individual node failure probability is not larger than 10%, an nCUBE machine of 1024 nodes [22] remains connected with probability at least 99%, while when individual node failure probability is bounded by 1.7%, a Thinking Machines' CM-2 computer of 65,536 nodes [31] remains connected with probability larger than 99.99%. To our knowledge, these are the first group of *precisely stated* and *formally proven* conclusions that show the high fault tolerance of hypercube networks, although they had been previously "observed" by extensive practical experience and experiments.

To further illustrate the power of our techniques, we study the success probability of a simple routing algorithm on hypercube networks, which is a slight modification of the simplest greedy "dimension-order" routing algorithm [9]. This very simple routing algorithm is deterministic and local-information-based, running in optimal time, and constructing a routing path of length bounded by a small constant plus twice of the optimal length. Compared to most routing algorithms proposed in the extensive research in the literature (e.g., [1, 4, 8, 10, 11, 12, 13, 26, 30, 33], see [6, 9, 15, 18] and their bibliographies for further references), this algorithm is simpler and weaker. However, our new techniques enable us to formally prove that even with such a simple routing algorithm, and with a very large fraction of faulty nodes, the hypercube networks can still route successfully with a very high probability.

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 $b_1b_2 \cdots b_{n-k}$ 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 $b_1b_2 \cdots b_{n-k}x_{n-k+1} \cdots x_n$, where each x_j is either 0 or 1. The subcube H_k will also be written as $H_k = b_1b_2 \cdots b_{n-k} **$. 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 $b_1b_2 \cdots b_{n-k} **$.

The concept of *local k -subcube connectivity* of a hypercube network has been proposed in [3]. We first give a quick review on the related definitions and results.

Definition 1 ([3]) 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.

In particular, a k -subcube H_k of the n -cube is k -subcube connected if less than half of the nodes in H_k are faulty, 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 1 ([3]) *The non-faulty nodes in a locally k -subcube connected n -cube make a connected graph for all $k \leq n$.*

Theorem 1 shows a very 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 NL(H_k)

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

We have the following lemma which gives the probability of **Event NL(H_k)**.

Lemma 1 $\text{Prob}[\text{NL}(H_k)] = p^k U_k(p)$, 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 [18], 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 **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 $B_{k,i}$, $k \leq i \leq 2^{k-1} - 1$, 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. The lemma follows after a simple rearrangement of the formula. \square

Lemma 1 implies the following important theorem immediately.

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

Proof. Let H_k be any k -subcube of the n -cube H_n . By Lemma 1, the probability that the k -subcube H_k is not locally k -subcube connected is equal to $p^kU_k(p)$. Thus, the probability that the k -subcube H_k is locally k -subcube connected is equal to $1 - p^kU_k(p)$. 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

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

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

Now the theorem follows since by Theorem 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 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 $9.881 \dots$ at point $p = 0.14119 \dots$. Now applying Theorem 2 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 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.

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

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, enables us to conclude the following.

Theorem 3 *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 $1 - 32 \cdot 2^{n-5} p^5 = 1 - 2^n p^5$.*

We give a few remarks on Theorem 3.

From the theoretical point of view, Theorem 3 has developed powerful techniques for formally proving lower bounds on the probability of hypercube connectivity in terms of the probability of individual node failures. As described in the introductory

section, although it was well observed that the hypercube networks are highly fault tolerant, there was no formal proofs for this important fact. Theorem 3 provides a formal proof of a precisely stated bound for this measure.

From the practical point of view, Theorem 3 provides a formally proven threshold on the node failure probability for manufacturers to achieve a desired probability for the connectivity of hypercube networks. In particular, when the dimension n of the hypercube network and a designated probability p' for network connectivity are given, by setting $1 - 2^n p^5 \geq p'$, we can derive an upper bound for the individual node failure probability p in terms of n and p' , which will guarantee the desired probability for network connectivity. For example, if we want to build an nCUBE machine of 1024 nodes [22], which is a 10-cube, and achieve connectivity of probability at least 99%, we can derive from $1 - 2^{10} p^5 \geq 99\%$ that $p \leq 10\%$. Thus, as long as the manufacturers ensure that the individual node failure probability is not larger than 10%, the probability of network connectivity of the nCUBE machine of 1024 nodes is at least 99%. Similarly, if we want to achieve connectivity of probability 99.99% for a Thinking Machines' CM-2 computer with 65,536 processors [31], we just need to ensure that the individual node failure probability is not larger than 1.7%. Note that the above required individual node failure probabilities are feasibly achievable by today's manufacturing technology, which thus provide formal proofs for the possibilities of building highly fault tolerant hypercube computers. To the authors' knowledge, Theorem 3 provides the first *formally proven* conclusion for this important fact, 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 connectivity 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 1 and Theorem 2), we have been able to show the high probability of connectivity of a hypercube network with faulty nodes. In this section, we show that our scheme is also very useful and powerful in the analysis of routing algorithms on hypercube networks.

We consider a routing algorithm that is a slight modification of the simplest *dimension-order* routing algorithm. Given two nodes $u = u_1 u_2 \cdots u_n$ and $v = v_1 v_2 \cdots v_n$ in the n -cube, the dimension-order algorithm routes from u to v by "converting" each bit u_i into v_i , in the order $i = 1, 2, \dots, n$ [9]. It has been well known that the dimension-order routing algorithm, though simple, is extremely vulnerable to network faults [9].

We modify the dimension-order routing scheme as follows. We start by converting the first $n - k$ bits u_i into v_i , in the order $i = 1, 2, \dots, n - k$. In case a bit u_i cannot be directly converted into v_i because of network faults, we try to flip a bit among

Algorithm. H-Router

Input: an n -cube H_n and two non-faulty nodes $u = u_1u_2 \cdots u_n$ and $v = v_1v_2 \cdots v_n$ in H_n

Output: a path of non-faulty nodes in H_n from u to v

1. $w = u$, and initialize the path $P = [w]$;
2. **for** $i = 1$ **to** $n - k$ **do**
 - if** $w_i \neq v_i$ {so $v_i = \overline{w_i}$ } **then**
 - if** $w' = w_1 \cdots w_{i-1} \overline{w_i} w_{i+1} \cdots w_n$ is non-faulty
 - then** extend the path P to w' ; let $w = w'$;
 - else if** there is a pair of non-faulty nodes of the forms

$$q = w_1 \cdots w_{i-1} w_i w_{i+1} \cdots w_{n-k} w_{n-k+1} \cdots w_{j-1} \overline{w_j} w_{j+1} \cdots w_n$$
 and

$$q' = w_1 \cdots w_{i-1} \overline{w_i} w_{i+1} \cdots w_{n-k} w_{n-k+1} \cdots w_{j-1} \overline{w_j} w_{j+1} \cdots w_n$$
 - then** extend the path P to q then to q' ; let $w = q'$;
 - else** STOP('routing fails');
3. apply BFS in the k -subcube $w_1 \cdots w_{n-k} **$ to convert the last k bits of w .

Figure 1: A routing algorithm for hypercube networks

the last k bits to make the i th bit convertible. After converting the first $n - k$ bits, we apply a Breadth First Search process in a k -subcube to convert the last k bits. The formal algorithm is presented in Figure 1.

The algorithm **H-Router** is deterministic and local-information-based, requiring no global network faulty information. Compared to most routing algorithms proposed in the extensive literature (e.g., [1, 4, 8, 10, 11, 12, 13, 26, 30, 33], see [6, 9, 15, 18] and their bibliographies for further references), this algorithm is simpler and looks weaker. However, in the rest of this section, using our new techniques, we formally prove that even with such a simple routing algorithm, and with a very large fraction of faulty nodes, the hypercube networks can still route successfully with a very high probability.

Note that the loop body of step 2 in the algorithm **H-Router** is executed only when $u_i \neq v_i$. Let $H_{n-k}(u, v)$ be the Hamming distance between the substrings $u_1u_2 \cdots u_{n-k}$ and $v_1v_2 \cdots v_{n-k}$, then the loop body in step 2 is executed exactly $H_{n-k}(u, v)$ times. During each execution of the loop body, at most 2 bits are flipped. Thus, step 2 of the algorithm runs in time $O(n)$ and increases the path length by at most $2H_{n-k}(u, v)$. Finally, step 3 of the algorithm **H-Router** runs in time $O(2^k)$ and extends the routing path to the destination node v in the k -subcube $v_1v_2 \cdots v_{n-k} **$.

Thus, the algorithm **H-Router**, if succeeds, runs in optimal time and, when k is small, constructs routing paths of length bounded by a small constant plus twice of the optimal length. However, the algorithm **H-Router** may fail when fault-free routing paths exist between the two given nodes. In the following, we use our new developed techniques to show that this happens with very small probability.

We say that two k -subcubes $b_1b_2 \cdots b_{n-k} **$ and $b'_1b'_2 \cdots b'_{n-k} **$ in the n -cube H_n are *neighboring k -subcubes* if the two binary strings $b_1b_2 \cdots b_{n-k}$ and $b'_1b'_2 \cdots b'_{n-k}$ 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 $q \neq w$ (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 2 $\text{Prob}[\mathbf{L2}(H_k)]$ 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 q_1, q_2, \dots, q_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 q'_1, q'_2, \dots, q'_k be the nodes in the k -subcube H'_k that are adjacent to the nodes q_1, q_2, \dots, q_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 q_j of w in H_k then to q_j 's neighbor q'_j in H'_k , where $j = 1, 2, \dots, k$. 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 $\{q_j, q'_j\}$ of the k pairs of nodes, $j = 1, 2, \dots, k$, where q_j is a neighbor of w in H_k and q'_j is q_j 's neighbor in H'_k , must contain at least one faulty node. Note that the probability that the pair $\{q_j, q'_j\}$ 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 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, 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. \square

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 $\text{Prob}[\mathbf{FN}(H_k)]$ 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, [5], page 121). \square

Let $u = u_1 u_2 \cdots u_n$ and $v = v_1 v_2 \cdots v_n$ 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 $u_1 u_2 \cdots u_{n-k}$ and $v_1 v_2 \cdots v_{n-k}$. We have the following theorem.

Theorem 4 Suppose that each node in the n -cube H_n has a uniform and independent failure probability p , then for any $k \leq n$ 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 1, 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

$$\begin{aligned} \text{Prob}[\mathbf{Bad}(H_k)] &\leq \text{Prob}[\mathbf{NL}(H_k)] + \text{Prob}[\mathbf{L2}(H_k)] + \text{Prob}[\mathbf{FN}(H_k)] \\ &\leq p^k U_k(p) + 4^k (n-k) p^{k+1} + p^{2k-2} \binom{2^k}{2k-2} \\ &= p^k \left[U_k(p) + 4^k (n-k) p + p^{k-2} \binom{2^k}{2k-2} \right] \end{aligned} \quad (3.1)$$

where the second inequality is by Lemma 1, Lemma 2, and Lemma 3, and the polynomial $U_k(p)$ is given in Lemma 1.

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

Event $\mathbf{Good}(H_k)$

- (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

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

Now we define \mathbf{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 \mathbf{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

$$\begin{aligned} \text{Prob}[\mathbf{GOOD}] &= (\text{Prob}[\mathbf{Good}(H_k)])^{2^{n-k}} \\ &= (1 - \text{Prob}[\mathbf{Bad}(H_k)])^{2^{n-k}} \\ &\geq 1 - 2^{n-k} \text{Prob}[\mathbf{Bad}(H_k)] \end{aligned} \tag{3.2}$$

Combining (3.1) and (3.2), we get immediately

$$\text{Prob}[\mathbf{GOOD}] \geq 1 - 2^{n-k} p^k \left[U_k(p) + 4^k(n-k)p + p^{k-2} \binom{2^k}{2k-2} \right] \tag{3.3}$$

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

Consider step 2 of the algorithm. In case the i th bit w_i of the node $w = w_1 w_2 \cdots w_n$ is not equal to the i th bit v_i of the destination node $v = v_1 v_2 \cdots v_n$, we look for a subpath from the node w in the k -subcube $H_k = w_1 \cdots w_{i-1} w_i w_{i+1} \cdots w_{n-k} *$ to the neighboring k -subcube $H'_k = w_1 \cdots w_{i-1} v_i w_{i+1} \cdots w_{n-k} **$. Under the conditions of the event \mathbf{GOOD} , the neighboring k -subcube H'_k of the k -subcube H_k

is reachable from the node w in H_k in two steps. That is, either w 's neighbor $w' = w_1 \cdots w_{i-1} v_i w_{i+1} \cdots w_n$ in H'_k is non-faulty, or a neighbor $q = w_1 \cdots w_{i-1} w_i w_{i+1} \cdots w_{n-k} w_{n-k+1} \cdots w_{j-1} \bar{w}_j w_{j+1} \cdots w_n$ of w in H_k and q 's neighbor $q' = w_1 \cdots w_{i-1} v_i w_{i+1} \cdots w_{n-k} w_{n-k+1} \cdots w_{j-1} \bar{w}_j w_{j+1} \cdots w_n$ in H'_k are both non-faulty. Thus, the path P can be extended to the k -subcube H'_k by adding at most two edges. Finally, since the loop body in step 2 is executed only when $u_i \neq v_i$, we conclude that the length of the path P after step 2 of the algorithm is bounded by $2H_{n-k}(u, v)$. Moreover, the running time of step 2 is bounded by $O(kH_{n-k}(u, v) + (n - k)) = O(kn)$.

Finally, we consider step 3 of the algorithm. Now the node w and the destination node v are in the same k -subcube $H_k = v_1 v_2 \cdots v_{n-k} **$. 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 [16] 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 3 of the algorithm runs in time $O(2^k)$.

Therefore, under the conditions of event **GOOD**, whose probability is at least $1 - 2^{n-k} p^k [U_k(p) + 4^k (n - k) p + p^{k-2} \binom{2k}{2k-2}]$, 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

Remark. The running time of the algorithm **H-Router** in Theorem 4 can be further improved to $O(kn + k^2)$. In fact, Latifi [16] 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 3 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 1 *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 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 1.

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

Again Corollary 1 provides explicitly stated and formally proven upper bounds on the individual node failure probability in order to achieve a designated success probability for the routing algorithm **H-Router**. For instance, as long as the individual node failure probability is not larger than 4.7%, the routing algorithm **H-Router** routes successfully in an nCUBE machine of 1,024 processors [22] with probability at least 99%, while when the individual node failure probability is bounded by 1.7%, the routing algorithm **H-Router** routes successfully in a Thinking Machines' CM-2 computer with 65,536 processors [31] with probability larger than 99.9%. Again since the required individual node failure probabilities are feasible in today's manufacturing technology, Corollary 1 provides formally proven conclusions showing that the simple routing algorithm **H-Router** works practically well.

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 the theoretical point of view, the scheme offers very general and powerful techniques for establishing lower bounds on the probability for network connectivity, while from the practical point of view, our scheme has provided formally proven threshold on the node failure probability that guarantees very high probability for network connectivity and efficiency and effectiveness of routing algorithms. Before closing 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 [2, 18, 27] 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 3 could be further improved. In fact, since every locally k -subcube connected n -cube is also locally $(k + 1)$ -subcube connected [3], 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 most likely improve the bound given in Theorem 3. Therefore, Theorem 3 is really an example of an application of Theorem 2, 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 computational difficulties in order to improve Theorem 3: 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 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 [16], 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 .

Finally, we would like to explain the advantage of our scheme over the previously proposed fault tolerance models. Compared to Najjar and Gaudiot's results [21], our results are precisely stated and formally proved, while the results in [21] are approximations based on an unverified conjecture. Now consider the model of k -

safe networks [7, 12, 17, 32]. The k -safe networks require that each non-faulty node have at least k non-faulty neighbors. Using the techniques developed in this paper, it is 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 connectivity of the network unless we also bound the number of faulty nodes by $2^k(n - k) - 1$ [17, 35]. 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 connectivity of the entire network.

Acknowledgments

The author Jianer Chen was supported in part by NSF Grants CCR-0000206, CCR-0311590, and CCF-0430683, and by the National Natural Science Foundation of China Grants No. 60373083 and No. 60433020 for Distinguished Young Scholars and for the Changjiang Scholar Reward Project. The author Iyad Kanj was supported in part by a DePaul University Competitive Research Grant.

References

1. W. A. AIELLO, F. T. LEIGHTON, B. M. MAGGS, AND M. NEWMAN, Fast algorithms for bit serial routing on a hypercube, *Math. Syst. Theory* 24, (1991), pp. 253-271.
2. S. B. AKERS AND B. KRISHNAMURTHY, A group-theoretic model for symmetric interconnection networks, *IEEE Transactions on Computers* 38, (1989), pp. 555-565.
3. 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.
4. M. S. CHEN AND K. G. SHIN, Adaptive fault-tolerant routing in hypercube multiprocessors, *IEEE Transactions on Computers* 39, (1990), pp. 1406-1416.
5. T. H. CORMEN, C. E. LEISERSON, AND R. L. RIVEST, *Introduction to Algorithms*, McGraw-Hill Book Company, New York, 1992.
6. J. DUATO, S. YALAMANCHILI, AND L. NI, *Interconnection Networks: An Engineering Approach*, IEEE Comput. Soc., Los Alamitos, 1997.
7. A. H. ESFAHANIAN, Generalized measures of fault tolerance with application to n -cube networks, *IEEE Transactions on Computers* 38, (1989), pp. 1586-1591.
8. S. FELPERIN, P. RAGHAVAN, AND E. UPFAL, A theory of wormhole routing in parallel computers, *IEEE Transactions on Computers* 45, (1996), pp. 704-713.
9. M. D. GRAMMATIKAKIS, D. F. HSU, M. KRAETZL, AND J. F. SIBEYN, Packet routing in fixed-connection networks: a survey, *Journal of Parallel and Distributed Computing* 54, (1998), pp. 77-132.
10. 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.
11. Q.-P. GU AND S. PENG, k -pairwise cluster fault tolerant routing in hypercubes, *IEEE Transactions on Computers* 46, (1997), pp. 1042-1049.
12. 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.
13. J. HASTAD, F. T. LEIGHTON, AND M. NEWMAN, Fast computation using faulty hypercubes, *Proc. 21st ACM Symp. Theory Comput.*, (1989), pp. 251-263.
 14. INTEL CORP., *Intel iPSC/2*, Intel Scientific Computers, 1988.
 15. S. LAKSHMIVARAHAN AND S. K. DHALL, Ring, torus and hypercube architectures/algorithms for parallel computing, *Parallel Computing* 25, (1999), pp. 1877-1906.
 16. S. LATIFI, Combinatorial analysis of the fault diameter of the n -cube, *IEEE Transactions on Computers* 42, (1993), pp. 27-33.
 17. S. LATIFI, M. HEDGE, AND M. NARAGHI-POUR, Conditional connectivity measures for large multiprocessor systems, *IEEE Transactions on Computers* 43, (1994), pp. 218-222.
 18. F. T. LEIGHTON, *Introduction to Parallel Algorithms and Architectures: Arrays, Trees, Hypercubes*, Morgan Kaufmann, San Mateo, CA, 1992.
 19. J. LIEBEHERR AND B. S. SETHI, A scalable control topology for multicast communications, *Proc. IEEE INFOCOM'98*, (1998), pp. 1197-1203.
 20. J. LIEBEHERR AND T. K. BEAM, HyperCast: a protocol for maintaining multicast group members in a logical hypercube topology, *Proc. First International Workshop on Networked Group Communication (NGC'99)*, (1999), pp. 72-89.
 21. W. NAJJAR AND J. L. GAUDIOT, Network resilience: a measure of network fault tolerance, *IEEE Transactions on Computers* 39, (1990), pp. 174-181.
 22. nCUBE CORPORATION, *nCUBE 2 Processor Manual*, Dec. 1990.
 23. D. K. PRADHAN, *Fault-Tolerant Computing: Theory and Techniques*, Vols. I and II, Prentice Hall, 1986.
 24. D. K. PRADHAN, *Fault-Tolerant Computer System Design*, Prentice Hall, 1996.
 25. D. K. PRADHAN AND S. M. REDDY, A fault-tolerant communication architecture for distributed systems, *IEEE Transactions on Computers* 31, (1982), pp. 863-870.
 26. M. RABIN, Efficient dispersal of information for security, load balancing, and fault tolerance, *J. ACM* 36, (1989), pp. 335-348.
 27. Y. SAAD AND M. H. SCHULTZ, Topological properties of hypercubes, *IEEE Transactions on Computers* 37, (1988), pp. 867-872.
 28. C. L. SEITZ, The cosmic cube, *Communications of the ACM* 28, (1985), pp. 22-33.
 29. D. P. SIEWIOREK AND S. R. SWARZ, *The Connection Machine*, Cambridge, Mass., MIT Press, 1985.
 30. G. D. STAMOULIS AND J. N. TSITSIKLIS, Greedy routing in hypercubes and butterflies, *IEEE Transactions on Computers* 42, (1994), pp. 3051-3061.
 31. THINKING MACHINES CORPORATION, *Connection Machine, Model CM-2 Technical Summary*, Version 6.0 edition, November 1990.
 32. 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.
 33. L. G. VALIANT, A scheme for fast parallel communication, *SIAM J. Computing* 11, (1982), pp. 350-361.
 34. A. WAGNER AND D. CORNEIL, Embedding trees in a hypercube is NP-complete, *SIAM J. Computing* 19, (1990), pp. 570-590.
 35. 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.