A Graph Partitioning Approach to Sequential Diagnosis

Sanjeev Khanna*   W. Kent Fuchs, *Fellow, IEEE*†

June 15, 1996

Abstract

This paper describes a generalized sequential diagnosis algorithm whose analysis leads to strong diagnosability results for a variety of multiprocessor interconnection topologies. The overall complexity of this algorithm in terms of total testing and syndrome decoding time is linear in the number of edges in the interconnection graph and the total number of iterations of diagnosis and repair needed by the algorithm is bounded by the diameter of the interconnection graph. The degree of diagnosability of this algorithm for a given interconnection graph is shown to be directly related to a graph parameter which we refer to as the partition number. We approximate this graph parameter for several interconnection topologies and thereby obtain lower bounds on degree of diagnosability achieved by our algorithm on these topologies. If we let N denote total number of vertices in the interconnection graph and Δ denote the maximum degree of any vertex in it, then our results may be summarized as follows. We show that a symmetric d-dimensional grid graph is sequentially $\Omega(N^{\frac{d}{d+1}})$-diagnosable for any fixed d. For hypercubes, symmeteric $\log N$-dimensional grid graphs, it is shown that our algorithm leads to a surprising $\Omega(\frac{N \log\log N}{\log N})$ degree of diagnosability. Next we show that the degree of diagnosability of an arbitrary interconnection graph by our algorithm is $\Omega(\sqrt{\frac{N}{\Delta}})$. This bound translates to an $\Omega(\sqrt{N})$ degree of diagnosability for cube-connected cycles and an $\Omega(\sqrt{\frac{N}{k}})$ degree of diagnosability for k-ary trees. Finally, we augment our algorithm with another algorithm to show that every topology is $\Omega(N^{\frac{1}{3}})$-diagnosable.

Keywords– Analysis of algorithms, degree of diagnosability, fault-tolerance, graph partitioning, multiprocessor systems, sequential diagnosis, system-level diagnosis.

*Bell Laboratories, Lucent Technologies, 700 Mountain Avenue, Murray Hill, NJ 07974-0636 (`sanjeev@theory.stanford.edu`).

†School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907-1285.

I. Introduction

The problem of identifying faulty processors in a multiprocessor system, known as *system-level diagnosis*, has been extensively studied in the literature [1, 2, 3, 4, 5]. The foundations of this area and the original diagnostic model were established in a classic paper by Preparata, Metze and Chien [1]. This model, known as the PMC model, has been widely studied [1, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15]. It assumes a system to be composed of units or processors capable of testing each other along the available communication channels. Once a unit u_1 has tested another unit u_2, it declares u_2 as fault-free or faulty. The outcome of the test is considered reliable iff unit u_1 is fault-free. Only permanent faults are considered in this model. In recent years, probabilistic fault models which allow intermittent faults have also been actively studied [16, 17, 18].

The process of interpreting the test results so as to correctly determine the status of various processors is known as *syndrome-decoding*. This can be done either by a central observer [1, 6, 8, 4, 7, 9] or it can be done in a distributed manner [11, 12, 13, 14, 15]. While the main criticism of the former approach is the bottleneck created by the central observer, the weakness of the distributed approach lies in the amount of message traffic generated and the global storage requirements for the diagnostic software and information. Motivated by these observations, the notion of semi-distributed diagnosis was introduced [19], where a group of processors is used to coordinate the diagnosis process. In this paper, we use the PMC model and assume existence of a central observer to coordinate the diagnosis and to process the outcomes of various tests. Furthermore, we assume that the set of faulty processors does not change during the execution of the algorithm.

Preparata et al. also introduced the notions of *one-step* and *sequential diagnosis* [1]. In the first approach, called one-step diagnosis or *diagnosis without repair*, the objective is to identify all the faulty units before any faulty unit is replaced or repaired. The latter approach, known as sequential diagnosis or *diagnosis with repair*, on the other hand, aims at iteratively identifying subsets of faulty units. At the end of each iteration, the identified subset of faulty units is repaired or replaced before the next iteration is initiated. This process is repeated until the system has been completely diagnosed and repaired. Given a diagnosis algorithm $\mathcal{A}$ for a system S, one-step or sequential, the largest integer t such that for any set of faults $\mathcal{F}$ with $|\mathcal{F}| \leq t$, the system can be correctly diagnosed, is referred to as the *degree of diagnosability* of $\mathcal{A}$ for S. A system is called *one-step t-diagnosable* if there exists an algorithm to correctly locate all faulty processors whenever the total number of faulty processors does not exceed t. Similarly, a system is called *sequentially t-diagnosable* if there exists an algorithm to correctly locate at least one faulty processor provided the total number of faulty processors does not exceed t. Hakimi and Amin gave a characterization of one-step t-diagnosable systems [6]. Huang et al. presented a characterization theorem for sequentially t-diagnosable systems [9].

A multiprocessor system S can be modeled as a graph G which we refer to as the *interconnection graph.* The vertices of G correspond to the processors and the edges correspond to direct communication channels available between pairs of processors. The graph G will be undirected as we assume the communication channels to be bi-directional. In our discussion, we use the terms processor and vertex interchangeably.

A well-known result implies that the maximum diagnosability of a one-step diagnosis algorithm for any system S, is bounded from above by the minimum vertex degree in its interconnection graph [1]. This result has pessimistic implications for systems whose interconnection graphs have one or more vertices of small degree. However, a variety of commercial multiprocessor systems are based on topologies which result in interconnection graphs containing many vertices of small degree. Some examples of such topologies include multi-dimensional grids (a special case is the hypercube[1]), cube-connected cycles and trees. Thus for example, in any tree interconnection graph, since there must exist a leaf vertex, there does not exist a one-step diagnosis algorithm which can always correctly diagnose in the presence of more than a single fault. This bound is independent of the total number of processors in the system.

For such topologies, as we explore in this paper, sequential diagnosis appears to be a significantly more powerful alternative. So far little previous work has been done in the area of sequential diagnosis algorithms. Kavianpour and Kim have argued that a hypercube is sequentially $\Theta(\sqrt{N \log N})$[2]-diagnosable where N is the total number of processors [10]. The crux of their argument lies in the existence of a simple cycle in the hypercube which contains a sufficiently small number of faulty units to be sequentially diagnosable using the results of [1] on single-loop systems. However, it is not clear how one might identify such a cycle.

The authors of the present paper have also developed a sequential diagnosis algorithm for the hypercube topology which achieves $\Omega(\sqrt{N \log N})$ degree of diagnosability [20]. The algorithm has $O(N)$ time complexity and requires $O(\log N)$ iterations of diagnosis and repair.

In this paper, we develop a generalized sequential diagnosis algorithm. The degree of

[1]Throughout this paper, the word hypercube refers to a binary hypercube.
[2]All logarithms used in this paper have base 2.

diagnosability of this algorithm for any given interconnection topology, is directly related to a graph parameter which we refer to as the *partition number*. Our approach yields surprisingly high degree of diagnosability for several interconnection topologies. Specifically, we show that a d-dimensional grid graph with each dimension of length $N^{\frac{1}{d}}$ is sequentially $\Omega(N^{\frac{d}{d+1}})$-diagnosable for any constant d. For hypercubes, grid graphs in $\log N$ dimensions where each dimension has length two, we prove an $\Omega(\frac{N \log\log N}{\log N})$ degree of diagnosability. We show that this algorithm has degree of diagnosability $\Omega(\sqrt{\frac{N}{\Delta}})$ for any interconnection graph with maximum vertex degree Δ. This result implies that the degree of diagnosability of a cube-connected cycles graph is $\Omega(\sqrt{N})$ and that a k-ary tree can be correctly diagnosed in presence of upto $\Omega(\sqrt{\frac{N}{k}})$ faulty processors. However, a lower bound of $\Omega(\sqrt{\frac{N}{\Delta}})$ is not effective when Δ is large. To handle such graphs effectively, we augment this algorithm to obtain an $\Omega(N^{\frac{1}{3}})$ lower bound on the degree of diagnosability of any arbitrary graph. The time complexity of the augmented version remains the same. The total time taken by our algorithm in testing and syndrome decoding for any interconnection graph is linear in the total number of edges in the graph. Lastly, the maximum number of iterations of diagnosis and repair needed by the algorithm are shown to be bounded by the diameter of the interconnection graph and one more than the diameter for the augmented version.

The remainder of this paper is organized as follows. Section II develops the essential relationship between the partition numbers and degree of diagnosability of a graph. We also design and analyze the complexity of the generalized sequential diagnosis algorithm. This section is concluded with a sketch of the basic approach to derive lower bounds on the degree of diagnosability achieved by our algorithm for various interconnection graphs. In

Sections III through VI, we estimate the partition numbers and derive diagnosability results for the symmetric grid graphs in a fixed dimension, hypercubes, cube-connected cycles, k-ary trees and arbitrary topologies, respectively. Section VII describes an approach to improve the time complexity for dense interconnection graphs.

II. Graph Partitioning and Sequential Diagnosis

A. Notation

For a given undirected graph G, we use $V(G)$ and $E(G)$ to denote its set of vertices and edges, respectively. $\mathcal{D}(G)$ denotes the diameter of the graph G and $\Delta(G)$ denotes the maximum degree of any vertex in G.

In order to concisely represent the performance characteristics of our algorithm for a given interconnection graph, a 3-tuple notation of the form $\langle t_F, t_T, t_I \rangle$ is used where t_F is a lower bound on the degree of diagnosability, t_T is an upper bound on the total testing and syndrome decoding time needed and t_I denotes an upper bound on the number of iterations of diagnosis and repair needed by the algorithm.

B. Partition Numbers and Sequential Diagnosability

Let G be a given interconnection graph. We will assume G to be a connected graph. For each edge $(x, y) \in E(G)$, let processor x test y and vice versa. The outcomes of the $2|E(G)|$ tests thus conducted can be abstracted into a labeled undirected graph called the *syndrome graph.* If we let G_S denote the syndrome graph, then $\mathcal{V}(G_S) = \mathcal{V}(G)$ and $\mathcal{E}(G_S)$ simply consists of the edges in $\mathcal{E}(G)$ with labels. An edge (x, y) is given label "pass" if x

declared y to be fault-free and vice versa. Similarly, we label an edge (x, y) as "fail" if x and y declare each other to be faulty. Any other edges are labeled as "conflict". The following simple lemma characterizes a useful property of this graph.

Lemma 1 *Let G_P be the subgraph of the syndrome graph G_S induced by the edges labeled as "pass". Then in each connected component of G_P, either all vertices are fault-free or all of them are faulty.* □

An immediate corollary is as follows :

Corollary 1 *Let t be an upper bound on the total number of faulty processors in the system. If the graph G_P contains a connected component of size $t+1$ or larger, then it must be the case that all these vertices correspond to fault-free processors.* □

Corollary 1 forms the basis of our generalized sequential diagnosis algorithm. Our approach is based on considering a certain parameter for the interconnection graph G which we refer to as the partition number.

Definition 1 *Given a connected graph G, we define the k-*partition number *of G as the largest integer p such that for all p-element subsets $S \subseteq V(G)$, the subgraph of G induced by the vertices in $V(G) - S$ has a connected component of size k or larger.* □

The k-partition number of G is denoted by $\phi_G(k)$ and in general for a given G, it will be a function of k and $|\mathcal{V}(G)|$. This function is undefined for $k > |V(G)|$.

Suppose we allow a maximum of t faulty processors in our system. Let U be the set of vertices in G corresponding to the fault-free processors and let E_U be defined as below :

$$E_U = \{(x, y) \in \mathcal{E}(G) | x, y \in U\} \tag{1}$$

Observe that $\mathcal{E}(G_P)$ must contain all the edges in E_U. Therefore, if the value t was chosen such that $\phi_G(t+1) \geq t$, then the graph G_P must have a component of size $t+1$ or larger. Moreover, by Corollary 1, any such component must consist solely of vertices corresponding to fault-free processors.

C. The Generalized Sequential Diagnosis Algorithm

We now use the above observations to design a generalized sequential diagnosis algorithm, referred to as the PARTITION algorithm. Let G be the interconnection graph for a given system S and let t be a non-negative integer such that $\phi_G(t+1) \geq t$. Then we can use the following algorithm to correctly diagnose all the faults in S provided there are no more than t faulty processors in S. For a clear exposition, we describe the algorithm as composed of two phases :

- *Phase 1 : Fault-free Subset Identification*

  The objective of this phase is to identify a subset of fault-free processors. Each processor is asked to test each one of its neighbors. The outcomes of these tests are used to construct the syndrome graph G_S. Let G_P be the subgraph of G_S induced by the edges labeled "pass". We do a depth-first search to locate a connected component of size at least $t+1$ in G_P. By our choice of t, we are guaranteed to find such a component and by Corollary 1, all the processors in this component must be fault-free.

- *Phase 2 : Iterative Diagnosis and Repair*

  This phase is aimed at iteratively diagnosing and repairing the faulty processors in the system using the subset identified in phase 1. Select an arbitrary fault-free processor, say u, from the component identified in phase 1 and construct a breadth-first search tree of G rooted at this processor. Let h denote the height of breadth-first search tree and let L_i, $0 \leq i \leq h$ denote the set of processors at distance i from u in this tree. We now use the following iterative scheme. Initialize an index variable, say i, to 0. Use the processors in set L_i to test the processors in the set L_{i+1}. If any processors in L_{i+1} are identified as faulty then repair or replace them. Increment the index i by 1 and repeat this till i equals h. Since in the $i^{\underline{\text{th}}}$ iteration, all the processors in the set L_i are known to be fault-free, they always diagnose the processors in the set L_{i+1} correctly.

Now let us analyze the total testing and syndrome decoding time taken by the PARTITION algorithm. Phase 1 is easily seen to be performed in $O(|E(G)|)$ time. In phase 2, the breadth-first search tree is constructed in $O(|E(G)|)$ time. The total number of tests conducted over all iterations is simply the number of edges in the breadth-first tree, that is, $|V(G)| - 1$. Aggregating over all the iterations, the total time taken in testing and forwarding this test information to the central observer for syndrome decoding, is $O(|V(G)|)$. Therefore, the overall complexity of this algorithm is $O(|E(G)|)$. Finally, since the height of the breadth-first search tree constructed in phase 2 above can be no more than $\mathcal{D}(G)$, the algorithm needs at most so many iterations of diagnosis and repair.

For a given graph G, $\phi_G(t)$ must be a monotonically non-increasing function of t. Therefore, the optimal value of t for the PARTITION algorithm is the largest integer satisfying the inequality $\phi_G(t+1) \geq t$. The following theorem summarizes the performance characteristics of the PARTITION algorithm.

Theorem 1 *For a given interconnection graph G, let t denote the largest integer such that $\phi_G(t+1) \geq t$. Then* PARTITION *is a*

$$\langle t, O(|E(G)|), \mathcal{D}(G) \rangle$$

sequential diagnosis algorithm for G. □

The PARTITION algorithm as described above, can be suitably modified to improve the performance in practice. For instance, in phase 2 of the algorithm, one may use a collection of breadth-first search trees which is constructed by doing a simultaneous breadth-first search from all the fault-free processors identified in phase 1. However, such modifications leave the asymptotic worst case performance of the algorithm unchanged.

D. Estimating the Degree of Diagnosability

We have not yet considered the issue of determining the function $\phi_G(.)$ for a given graph G. In general, it might be very hard to determine this function exactly. However, for our application, it will be sufficient to closely approximate this function by another function $\phi_G^*(k)$ such that for a given G and any non-negative integer $k \leq |V(G)|$, $\phi_G^*(k) \leq \phi_G(k)$. A two-step approach is used to approximate the function $\phi_G(k)$.

First, we concentrate on a closely related function $\psi_G(k)$ which we define as the smallest integer p such that there exists a p-element set $S \subseteq V(G)$ so that the subgraph of G induced by the vertices in $V(G) - S$ has no connected component of size k or larger. The following simple lemma relates the two functions.

Lemma 2 *For any connected graph G and a non-negative integer $k \leq V(G)$, $\phi_G(k) = \psi_G(k) - 1$.* □

As we will see, it is more convenient to estimate the function $\psi_G(k)$. We first approximate $\psi_G(k)$ by $\psi^*_G(k)$ such that $\psi^*_G(k) \leq \psi_G(k)$, and then use Lemma 2 to obtain a candidate solution for $\phi^*_G(k)$. It is easily seen that the largest integer t^* satisfying $\phi^*_G(t^* + 1) \geq t^*$ is always less than or equal to the largest integer t satisfying $\phi_G(t + 1) \geq t$.

III. Symmetric Grid Graphs

A d-dimensional grid graph is defined as the graph obtained by $P_{n_1} \times P_{n_2} \times ... \times P_{n_d}$ where $\times$ is the graph cartesian product operation, P_{n_i} denotes a path on n vertices and n_i, $1 \leq i \leq d$ denotes the length of dimension i. Let this graph be denoted by $\mathcal{G}_d(n_1, n_2, .., n_d)$. A grid graph can alternatively be viewed as a recursively constructed graph. More specifically, it can be recursively defined as $\mathcal{G}_d(n_1, n_2, .., n_d) = \mathcal{G}_{d-1}(n_1, n_2, .., n_{d-1}) \times P_{n_d}$ where $\mathcal{G}_1(n_1)$ is simply a copy of P_{n_1}. It is easy to see that $\mathcal{G}_d(n_1, n_2, .., n_d)$ contains $\Pi_{i=1}^d n_i$ vertices. A grid graph is called *symmetric* if each dimension of the graph has the same length. A symmetric d-dimensional grid graph with each dimension n units long, is denoted by $\mathcal{G}_d(n)$.

In this section, we will derive diagnosability results for d-dimensional grid graphs when d is a constant. The following lemma gives a useful property of subgraphs of such graphs.

Lemma 3 *Let H be a subgraph of $\mathcal{G}_d(n)$ where d is a constant. Then the vertex degree sum* *of H is bounded from above by*

$$2d(|V(H)| - |V(H)|^{\frac{d-1}{d}})$$

Furthermore, this bound is exact if H is isomorphic to $\mathcal{G}_d(p)$ for some integer $p \geq 2$.

Proof : We prove by induction on d. The basis case with $d = 1$ is trivial.

Let k denote $|V(H)|$ and let $S(d,k)$ denote the maximum degree sum for any k-vertex subgraph of a d-dimensional grid graph. Using the recursive definition, we know that the graph $\mathcal{G}_d(n)$ is composed of n copies of $\mathcal{G}_{d-1}(n)$, say $G_1, G_2, ..., G_n$. Let $x_i = |V(H) \bigcap V(G_i)|$, $1 \leq i \leq n$ and let $\mathcal{P}(n,k)$ denote the set of all n-tuples $\vec{X} = (x_1, x_2, ..., x_n)$ such that $x_i's$ are non-negative integers and $\sum_{i=1}^{n} x_i = k$. Using this notation, we can write the following recurrence to get an upper bound on $S(d,k)$:

$$S(d,k) \leq \max_{\vec{X} \in \mathcal{P}(n,k)} \{\sum_{i=1}^{n} S(d-1, x_i) + 2\sum_{i=1}^{n} x_i - 2 \max_{1 \leq i \leq n} x_i\} \tag{2}$$

The term $\sum_{i=1}^{n} S(d-1, x_i)$ expresses the maximum vertex degree sum contribution due to the edges in H with each end-point inside a G_i, $1 \leq i \leq n$. The expression $2\sum_{i=1}^{n} x_i - 2\max_{1 \leq i \leq n} x_i$ on the other hand, indicates the maximum possible contribution to the vertex degree sum due to edges connecting vertices of G_i with G_{i+1} where $1 \leq i < n$. To see this, observe that the total number of such edges cannot exceed $\sum_{i=1}^{n-1} \min(x_i, x_{i+1})$, and this sum is clearly bounded by $\sum_{i=1}^{n} x_i - \max_{1 \leq i \leq n} x_i$.

Using the induction hypothesis, we can rewrite the above inequality as below :

$$S(d,k) \leq \begin{cases} \max_{\vec{X}\in\mathcal{P}(n,k)}\{4k - 2\sum_{i=1}^{n}\min(1,x_i) - 2\max_{1\leq i\leq n} x_i\} \text{ if } d = 2 \\ \max_{\vec{X}\in\mathcal{P}(n,k)}\{2dk - 2(d-1)\sum_{i=1}^{n} x_i^{\frac{d-2}{d-1}} - 2\max_{1\leq i\leq n} x_i\} \text{ if } d > 2 \end{cases} \tag{3}$$

We only describe the analysis needed to get the upper bound on $S(d,k)$ when $d > 2$. An essentially identical argument can be used to show that $S(2,k) \leq 4(k-\sqrt{k})$. Observe that to get this upper bound when $d > 2$, it is sufficient to solve the following minimization problem :

$$\min_{\vec{X}\in\mathcal{P}(n,k)} (2(d-1)\sum_{i=1}^{n} x_i^{\frac{d-2}{d-1}} + 2\max_{1\leq i\leq n} x_i) \tag{4}$$

A two-step approach is used to solve this minimization problem. Let $x = \max_{1\leq i\leq n} x_i$. For a given value of x, we first determine the minimum value for the objective function in equation 4. Next we minimize this value over all values of x. So our first step is to solve the following minimization problem :

$$\min_{\vec{X}\in\mathcal{P}(n,k),\ \max_{1\leq i\leq n} x_i = x} \sum_{i=1}^{n} x_i^{\frac{d-2}{d-1}} \tag{5}$$

To determine the solution to this problem, we analyze the distribution of values which must be satisfied by pairs of variables in the optimal solution. Let us consider any two variables x_i and x_j. Suppose $x_i + x_j = y$. Now consider the function $f(x_i) = x_i^p + (y - x_i)^p$ where $0 < p = \frac{d-2}{d-1} < 1$. Since $f''(x_i) < 0$ in the interval $[0, y]$, $f(x_i)$ is concave over this interval. Given that $x_i \leq x$ and $x_j \leq x$, the minima for $f(x_i)$ subject to this constraint occurs uniquely at the boundary $x_i = \min(x,y)$ or $x_i = y - \min(x,y)$. If $x > y$, the minimum value is obtained by setting $x_i = y$ or $x_i = 0$. Otherwise, the minimum value is obtained by setting $x_i = x$ or $x_i = y - x$. This argument implies that if $(x_1^*, x_2^*, ..., x_n^*)$ is the solution to the minimization problem specified in equation 5, then there cannot exist $1 \le i < j \le n$ such that $0 < x_i^*, x_j^* < x$. Thus the minimum occurs precisely when $\lfloor \frac{k}{x} \rfloor$ of the $x_i's$ take the value x and the remaining ones are set to zero except possibly one which takes the value $k - x\lfloor \frac{k}{x} \rfloor$.

So for a given value of x, the minimum value of the function in equation 4 is $2(d-1)(\lfloor \frac{k}{x} \rfloor x^{\frac{d-2}{d-1}} + (k - x\lfloor \frac{k}{x} \rfloor)^{\frac{d-2}{d-1}}) + 2x$. Let the function $g(x)$ denote this minimum value for a given x. It is rather easy to see that $(\frac{k}{x} - \lfloor \frac{k}{x} \rfloor) x^{\frac{d-2}{d-1}} \le (k - x\lfloor \frac{k}{x} \rfloor)^{\frac{d-2}{d-1}}$ for $x > 0$. Therefore, if we define a function $h(x)$ by $h(x) = 2(d-1)(\frac{k}{x})x^{\frac{d-2}{d-1}} + 2x$, then $g(x) \ge h(x)$ for $x > 0$. Relaxing the integrality constraint on x, we find that the function $h(x)$ is minimized when $x = k^{\frac{d-1}{d}}$ and its minimum value is $2dk^{\frac{d-1}{d}}$. Thus $g(x)$ is bounded from below by this value.

The last step now simply involves substituting the lower bound on the value of $g(x)$ determined above in equation 3 to get an upper bound on $S(d,k)$. This gives us the desired result.

Finally, the bound determined above is verified to be exact when H is isomorphic to $\mathcal{G}_d(p)$ for $p \ge 2$ by straightforward induction on d. $\square$

Lemma 4 *For every non-negative integer $t \le N$,*

$$\psi_{\mathcal{G}_{d(n)}}(t+1) \ge \frac{N - (tN^{d-1})^{\frac{1}{d}}}{1 + t^{\frac{1}{d}}}$$

where $N = n^d$ denotes the total number of vertices in $\mathcal{G}_d(n)$ and d is a constant.

Proof : The inequality in the statement trivially holds when $t = 0$. We now use Lemma 3 to analyze the case when $t > 0$. Let X_t denote a set of vertices such that the graph induced by $V(\mathcal{G}_d(n)) - X_t$ contains no connected component with more than t vertices. Using the convexity of the function $2d(k - k^{\frac{d-1}{d}})$ for non-negative k, and an analysis similar to one used in Lemma 3, it is not difficult to show that an upper bound on the sum of the degrees of the vertices in the graph induced by $V(\mathcal{G}_d(n)) - X_t$ is given by $\frac{N-|X_t|}{t}(2d(t - t^{\frac{d-1}{d}}))$. Thus $|X_t|$ must satisfy the following inequality :

$$\frac{N - |X_t|}{t}(2d(t - t^{\frac{d-1}{d}})) + 4d|X_t| \geq 2d(N - N^{\frac{d-1}{d}}) \tag{6}$$

The term $4d|X_t|$ gives an upper bound on the possible degree loss due to deletion of $|X_t|$ vertices from $\mathcal{G}_d(n)$.

Rearranging the terms, we get the following lower bound on $|X_t|$:

$$|X_t| \geq \frac{N - (tN^{d-1})^{\frac{1}{d}}}{1 + t^{\frac{1}{d}}} \tag{7}$$

The lemma follows. □

Theorem 2 *For a symmetric d-dimensional grid graph with N vertices, the algorithm* PARTITION *is a*

$$\langle \Omega(N^{\frac{d}{d+1}}), O(N), d(N^{\frac{1}{d}} - 1) \rangle$$

sequential diagnosis algorithm if d is a constant.

Proof : Using Lemmas 2, 4, and Theorem 1, the degree of diagnosability is at least as large as the largest integer t satisfying the following inequality :

$$\frac{N - (tN^{d-1})^{\frac{1}{d}}}{1 + t^{\frac{1}{d}}} - 1 \geq t \tag{8}$$

For sufficiently large N, this inequality is satisfied by $t = \delta N^{\frac{d}{d+1}}$ for any non-negative constant $\delta < 1$. Thus t_F is $\Omega(N^{\frac{d}{d+1}})$.

Since degree of a vertex is bounded by d, total number of edges is $O(N)$. Hence t_T is $O(N)$ by Theorem 1.

Finally, it is readily seen that in a symmetric grid graph, the total number of edges is $O(N)$ and the diameter is $d(N^{\frac{1}{d}} - 1)$. Substituting in Theorem 1, we get the stated result.

□

IV. Hypercube Graphs

A d-dimensional hypercube, denoted by $\mathcal{H}_d$, is simply a symmetric grid graph with each dimension of length 2. Thus if the hypercube has N nodes then it has $\log N$ dimensions. The theorem in the previous section applies only to constant dimension symmteric grid graphs. In this section we use a similar technique to analyze hypercubes.

Lemma 5 *Let H be a subgraph of $\mathcal{H}_d$, $|V(G)| \geq 1$. Then the vertex degree sum of H is bounded from above by*

$$|V(H)| \log(|V(H)|)$$

Furthermore, this bound is exact if H is isomorphic $\mathcal{H}_p$ for some non-negative integer p.

Proof : As before, we use induction on d. The basis case with $d = 1$ is trivial.

Let k denote $|V(H)|$ and let $S(d,k)$ denote the maximum degree sum for any k-vertex subgraph of $\mathcal{H}_d$. Using an argument identical to the one in Lemma 3, we can write the following recurrence for $S(d,k)$.

$$\begin{aligned} S(d,k) \quad &\leq \quad \max_{0\leq x\leq k}\{S(d-1,x)+S(d-1,k-x)+ \\ &\qquad 2\min(x,k-x)\} \end{aligned} \tag{9}$$

If $x=0$, then we are done by induction hypothesis. Otherwise, using the induction hypothesis, we can rewrite equation 9 as follows.

$$S(d,k) \leq \max_{1\leq x\leq \lfloor \frac{k}{2} \rfloor}\{x\log x+(k-x)\log(k-x)+2x\} \tag{10}$$

Let $f(x)=x\log x+(k-x)\log(k-x)+2x$. Since $f''(x)>0$ in the interval $[1,\lfloor\frac{k}{2}\rfloor]$, the function $f(x)$ is convex over this interval. Therefore the maximum occurs at the boundary of this interval and is easily verified to be bounded by $k\log k$.

Finally, it is trivial to verify that the bound is exact when H is isomorphic $\mathcal{H}_p$ for some non-negative integer p. $\square$

Lemma 6 *For every positive integer $t\leq N$,*

$$\psi_{\mathcal{H}_d}(t+1)\geq \frac{N\log(N/t)}{\log(N^2/t)}$$

where $N=2^d$ is the total number of vertices in $\mathcal{H}_d$.

Proof : The proof is similar to that of Lemma 4. If X_t denotes a set of vertices such that the graph induced by $V(\mathcal{H}_d) - X_t$ contains no connected component with more than t vertices, then applying Lemma 5, we get the following inequality :

$$(\frac{N - |X_t|}{t})(t \log t) + 2|X_t| \log N \geq N \log N \tag{11}$$

Solving for $|X_t|$, we get

$$|X_t| \geq \frac{N \log(N/t)}{\log(N^2/t)} \tag{12}$$

Thus we get the desired result. □

Finally, we now derive a lower bound on the degree of diagnosability of hypercubes using the above lemma.

Theorem 3 *For a hypercube graph with N vertices,* PARTITION *is a*

$$\langle \Omega(\frac{N \log \log N}{\log N}), O(N \log N), \log N \rangle$$

sequential diagnosis algorithm.

Proof : Using Lemmas 2 and 6, and Theorem 1, the degree of diagnosability is at least as large as the largest integer t which satisfies the following inequality :

$$\frac{N \log(N/t)}{\log(N^2/t)} - 1 \geq t \tag{13}$$

For sufficiently large N, this inequality is satisfied by $t = \delta \frac{N \log \log N}{\log N}$ for any non-negative

$\delta < 1$. Thus t_F is $\Omega(\frac{N \log\log N}{\log N})$.

The values of t_T and t_I follow in a straightforward manner from Theorem 1. $\square$

V. Cube-Connected Cycles and k-ary Trees

We first derive a general lower bound on the partition numbers of a graph G with a given maximum vertex degree. For sake of clarity, we slightly abuse our notation in the present and the next section; we use V and Δ to denote $V(G)$ and $\Delta(G)$ respectively.

Lemma 7 *For every non-negative integer $t \leq |V|$,*

$$\psi_G(t+1) \geq \frac{|V| - t}{(\Delta - 1)t + 1}$$

Proof : As before, we let X_t denote a set of vertices such that the graph induced by $V - X_t$ contains no connected component with more than t vertices. Since no vertex in G has degree more than Δ, we observe that deletion of a vertex from any subgraph of G cannot create more than $\Delta - 1$ new components. Thus after we delete the vertices in set X_t, the total number of connected components in the remaining subgraph can be no more than $1 + (\Delta - 1)|X_t|$. By the pigeonhole principle, therefore, at least one component must have at least

$$\frac{|V| - |X_t|}{1 + (\Delta - 1)|X_t|}$$

vertices. Thus X_t must satisfy the following inequality :

$$\frac{|V| - |X_t|}{1 + (\Delta - 1)|X_t|} \leq t \tag{14}$$

Simplifying, we get $|X_t| \geq \frac{|V|-t}{(\Delta-1)t+1}$. □

The above lemma gives us the following general theorem.

Theorem 4 *For any given interconnection graph G, the* PARTITION *algorithm is a*

$$\langle \lfloor \frac{\sqrt{4(\Delta - 1)(|V| - 1) + (\Delta + 1)^2} - (\Delta + 1)}{2(\Delta - 1)} \rfloor, O(|E|), \mathcal{D} \rangle$$

sequential diagnosis algorithm.

Proof : It is sufficient to indicate how the value of t_F is determined. Using Lemmas 2 and 7, and Theorem 1, like before, we get a quadratic inequality as below :

$$(\Delta - 1)t^2 + (\Delta + 1)t - (|V| - 1) \leq 0 \tag{15}$$

Solving for largest value of t gives us the result. □

A. Cube-Connected Cycles

The cube-connected cycles topology for a given dimension d is obtained by replacing the vertices of $\mathcal{H}_d$ by cycles of length d each such that if nodes x and y were adjacent to each other along dimension i in $\mathcal{H}_d$, then in the cube-connected cycles, we connect the i^{th} vertex of the cycle at vertex x to the i^{th} vertex of the cycle at vertex y. Thus each vertex in this topology, has two neighbors in the cycle containing it and one neighbor in another cycle. A cube-connected cycles structure in dimension d is denoted by $\mathcal{C}_d$. The total number of vertices in $\mathcal{C}_d$ is $d2^d$. The following lemma bounds the diameter of $\mathcal{C}_d$; the proof follows from [21].

Lemma 8 *The diameter of $\mathcal{C}_d$ is bounded by $\lfloor \frac{5d}{2} \rfloor - 1$.* $\square$

Lemma 8 combined with Theorem 4 gives us the following result.

Theorem 5 *For a cube-connected cycles graph with $N = d2^d$ vertices, the* PARTITION *algorithm is a*

$$\langle \frac{\sqrt{2N+2}-2}{2}, O(N), \lfloor \frac{5d}{2} \rfloor - 1 \rangle$$

sequential diagnosis algorithm. $\square$

B. k-ary Trees

An interesting class of trees is the k-ary trees where each vertex has no more than k children, $k \geq 2$. Clearly, the maximum degree of any vertex in a k-ary tree is $k+1$. Using this observation with Theorem 4 implies that every k-ary tree is $\Omega(\sqrt{\frac{N}{k}})$-diagnosable by the PARTITION algorithm. An important subclass of k-ary trees is the family of complete k-ary trees. Every non-leaf vertex in a complete k-ary tree has exactly k children and depth of any pair of leaves differs by at most one. It is a simple exercise to verify that the diameter of a complete k-ary tree with N vertices is bounded by $\lceil 2 \log_k (Nk + 1 - N) - 2 \rceil$. Thus using Theorem 4, we have the following result for complete k-ary trees.

Theorem 6 *For a complete k-ary tree with N vertices, the* PARTITION *algorithm is a*

$$\langle \frac{\sqrt{4kN + k^2 + 4} - (k+2)}{2k}, O(N), \lceil 2 \log_k (Nk + 1 - N) - 2 \rceil \rangle$$

sequential diagnosis algorithm. □

VI. Arbitrary Topologies

In this section, we derive a lower bound on the degree of diagnosability of any arbitrary topology. The lower bound given by Theorem 4 is effective only when the maximum degree of the interconnection graph G is not very large. To get better bounds for graphs with one or more vertices of sufficiently large degree, we augment the PARTITION algorithm with another algorithm which gives us better bounds on degree of diagnosability when G contains one or more vertices of sufficiently large degree.

As before, let G be a given interconnection graph. Consider the following algorithm which we call the MAX algorithm :

- *Phase 1 : Diagnosing a Single Processor*

  Select a vertex of largest degree in G, say v. Let each of the neighbors of v test the vertex v. Suppose precisely x of the neighbors declare v to be fault-free. If $x+1 \geq \Delta - x$ then v is determined to be fault-free else v is determined to be faulty.

- *Phase 2 : Iterative Diagnosis and Repair*

  If vertex v is identified to be faulty in phase 1 then it is repaired or replaced. Now we proceed in a manner identical to phase 2 of the PARTITION algorithm and hence we omit details. However, we do notice that the total number of iterations of diagnosis and repair are now bounded by $\mathcal{D}(G) + 1$.

The issue now is to determine how many faulty processors can we allow such that any vertex of maximum degree in G is correctly diagnosed in the phase 1 of the MAX algorithm.

Theorem 7 *For any given interconnection graph G, the* MAX *algorithm is a*

$$\langle \lfloor \frac{\Delta}{2} \rfloor, O(|E(G)|), \mathcal{D}(G) + 1 \rangle$$

sequential diagnosis algorithm. $\square$

Proof : We only consider determination of t_F. Let v be a vertex of maximum degree in G. Suppose x neighbors of v declare it to be fault-free and the remaining declare it to be faulty. If we assume v to be fault-free, then we must have at least $\Delta - x$ faulty vertices in G. On the other hand, if we assume v to be faulty, then we must have at least $x + 1$ faulty vertices in G. It is sufficient to choose t_F such that $\Delta - x \leq t_F$ implies $x + 1 > t_F$ and vice versa. Choosing $t_F = \lfloor \frac{\Delta}{2} \rfloor$ achieves this objective. $\square$

The modified PARTITION algorithm can be stated as follows. If $\lfloor \frac{\Delta}{2} \rfloor \geq \lfloor \frac{\sqrt{4(\Delta-1)(|V|-1)+(\Delta+1)^2}-(\Delta+1)}{2(\Delta-1)} \rfloor$ then use the MAX algorithm for diagnosis else use the PARTITION algorithm.

This algorithm is referred to as the MAX_PARTITION algorithm. It is easy to verify that for any G, the maximum of the degree of diagnosabilities of the MAX and the PARTITION algorithm is $\Omega(|V(G)|^{\frac{1}{3}})$. Thus we have the following theorem.

Theorem 8 *For any given interconnection graph G, the* MAX_PARTITION *algorithm is a*

$$\langle \Omega(|V(G)|^{\frac{1}{3}}), O(|E(G)|), \mathcal{D}(G) + 1 \rangle$$

sequential diagnosis algorithm. □

VII. Improving Time Complexity for Dense Graphs

As shown by our analysis in Sections II and VI, the time complexity of both PARTITION and MAX_PARTITION is directly proportional to the number of edges in the interconnection graph. Many topologies of interest lead to sparse interconnection graphs, that is, graphs with $O(N)$ edges, and thus these algorithms are $O(N)$ time algorithms for such topologies. A significant exception to this is the hypercube topology which leads to interconnection graphs with $O(N \log N)$ edges. In such cases when the interconnection graph is dense, however, we may use the structural properties of the underlying topology to embed a suitable spanning subgraph with relatively fewer edges, say $f(N)$, and treat this subgraph as the interconnection graph of interest for the diagnosis algorithm. If this embedding can itself be done in $O(f(N))$ time, then the time complexity of our algorithm is now $O(f(N))$. Alternatively, the embedding may be simply regarded as a one time pre-processing step after which the diagnosis will be performed several times. However, it is worthwhile to note that it is not very likely that the embedded subgraph will have the same diagnosability and diameter as the original graph. This process, therefore, highlights a potential trade-off between the time complexity and the diagnosability and number of iterations of diagnosis and repair needed by the algorithm. We illustrate our point for the hypercube topology.

It is well-known that multi-dimensional grids can be embedded in hypercubes [22]. Consider a d-dimensional hypercube with d even and embed the graph $\mathcal{G}_2(2^{\frac{d}{2}}, 2^{\frac{d}{2}})$ in the hypercube. This embedding is based on a reflected Gray code sequence and can in fact be done in

$O(\log N)$ time. The embedding process simply involves each processor using its d-bit address to compute the addresses of its neighbors in the embedded copy of $\mathcal{G}_2(2^{\frac{d}{2}}, 2^{\frac{d}{2}})$. Now when the PARTITION algorithm is executed, we only consider the hypercube edges which correspond to the embedded copy of $\mathcal{G}_2(2^{\frac{d}{2}}, 2^{\frac{d}{2}})$. Though the algorithm now takes $O(N)$ time, using Theorem 2, we only ensure $\Omega(N^{\frac{2}{3}})$ as the degree of diagnosability. Quite interestingly, even though the diameter of $\mathcal{G}_2(2^{\frac{d}{2}}, 2^{\frac{d}{2}})$ is $2^{\frac{d}{2}+1} - 2$, we can use a simple scheme based on the recursive definition of a hypercube to complete iterative diagnosis and repair in no more than d iterations without altering the $O(N)$ time complexity. The details of this can be found in [20]. When d is odd, we can use a similar approach on its two $(d-1)$-dimensional subcubes and embed a copy of $\mathcal{G}_2(2^{\frac{d-1}{2}}, 2^{\frac{d-1}{2}})$ in each of them. Again, we can now apply Theorem 2 to argue $\Omega(N^{\frac{2}{3}})$ degree of diagnosability.

In the above illustration, we embedded a two-dimensional grid graph to show that $\Omega(N^{\frac{2}{3}})$ diagnosability can be achieved in linear time for hypercubes. This idea can be extended in a straightforward manner to embed c-dimensional symmetric grid graphs in large enough hypercubes to achieve $\Omega(N^{\frac{c}{c+1}})$ diagnosability in linear time for some constant c.

VIII. Conclusions

We presented a generalized sequential diagnosis algorithm and analyzed its degree of diagnosability for several different topologies which included symmetric grid graphs, cube-connected cycles and k-ary trees. Our results show that sequential diagnosis can be used to achieve very high degree of diagnosability even when the vertex degrees are small in the interconnection graph. It was also shown that an augmented version of this algorithm ensures $\Omega(N^{\frac{1}{3}})$

diagnosability for any arbitrary interconnection graph with N vertices. The time complexity of both algorithms was shown to be linear in the number of edges in the interconnection graph.

One important problem not addressed in this work is the determination of good upper bounds on the degree of sequential diagnosability for the specific topologies studied here. We suspect that an analysis in the spirit of the graph partitioning approach of the PARTITION algorithm may lead to determination of good upper bounds. Another interesting area of investigation is development of an approach which is a hybrid of the one-step diagnosis and sequential diagnosis strategies. More precisely, instead of exploring the diagnosability in the extreme situations which either allow no repair or a linear number of iterations of diagnosis and repair, study the diagnosability when the total number of iterations of diagnosis and repair are required to be bounded by some function of the total number of processors.

Acknowledgements

The authors would like to express their sincere thanks to Edgar Ramos for a significant discussion on Lemma 3, Ran Libeskind-Hadas for his helpful comments on an early version of this paper, the anonymous referees for their careful review and suggestions, and Prof. Doug Blough for providing reference [10].

This research was supported in part by the Department of the Navy and managed by the Office of Naval Research under Grant N00014-91-J-1283, by the Semiconductor Research Corporation under Contract 95-DP-109, and by the Joint Services Electronics Program (JSEP) under grant N00014-96-1-0129.

References

[1] F. P. Preparata, G. Metze, and R. T. Chien, "On the connection assignment problem of diagnosable systems," *IEEE Trans. Electron. Comput.*, vol. EC-16, pp. 848–854, December 1967.

[2] A. D. Friedman and L. Simoncini, "System-level fault diagnosis," *IEEE Computer*, vol. 13, pp. 47–53, March 1980.

[3] C. Kime, "System diagnosis," in *Fault-Tolerant Computing: Theory and Techniques* (D. K. Pradhan, ed.), vol. II, ch. 8, Englewood Cliffs, NJ: Prentice Hall, 1986.

[4] A. K. Somani, V. K. Agarwal, and D. Avis, "A generalized theory for system level diagnosis," *IEEE Trans. Comput.*, vol. C-36, pp. 538–546, May 1987.

[5] A. T. Dahbura, "System-level diagnosis: A perspective for the third decade," in *Concurrent Computation: Algorithms, Architectures, Technologies*, New York: Plenum, 1988.

[6] S. L. Hakimi and A. T. Amin, "Characterization of connection assignment of diagnosable systems," *IEEE Trans. Comput.*, vol. C-23, pp. 86–88, January 1974.

[7] J. R. Armstrong and F. G. Gray, "Fault diagnosis in a boolean n-cube array of microprocessors," *IEEE Trans. Comput.*, vol. C-30, pp. 587–590, August 1981.

[8] A. T. Dahbura and G. M. Masson, "An $O(n^{2.5})$ fault identification algorithm for diagnosable systems," *IEEE Trans. Comput.*, vol. C-33, pp. 485–492, June 1984.

[9] S. Huang, J. Xu, and T. Chen, "Characterization and design of sequentially t-diagnosable systems," in *Proc. IEEE Computer Society* 19^{th} *Int'l Symp. Fault-Tolerant Computing*, pp. 554–559, 1989.

[10] A. Kavianpour and K. H. Kim, "A comparative evaluation of four basic system-level diagnosis strategies for hypercubes," *IEEE Trans. Reliability*, vol. 41, pp. 26–37, March 1992.

[11] J. G. Kuhl and S. M. Reddy, "Distributed fault tolerance for large multiprocessor systems," in *Proc. 1980 Computer Architecture Conference*, pp. 222–229, May 1980.

[12] S. H. Hosseini, J. G. Kuhl, and S. M. Reddy, "A diagnosis algorithm for distributed computing systems with dynamic failure and repair," *IEEE Trans. Comput.*, vol. C-33, pp. 223–233, March 1984.

[13] A. Bagchi and S. L. Hakimi, "An optimal algorithm for distributed system level diagnosis," in *Proc. IEEE Computer Society* 21^{st} *Int'l Symp. Fault-Tolerant Computing*, pp. 214–221, 1991.

[14] R. Bianchini Jr. and R. Buskens, "An adaptive distributed system-level diagnosis algorithm and its implementation," in *Proc. IEEE Computer Society* 21^{st} *Int'l Symp. Fault-Tolerant Computing*, pp. 222–229, 1991.

[15] A. Bagchi, "A distributed algorithm for system-level diagnosis in hypercubes," in *Proc. IEEE Workshop on Fault-Tolerant Parallel and Distributed Systems*, pp. 106–113, July 1992.

[16] D. M. Blough, G. F. Sullivan, and G. M. Masson, "Almost certain diagnosis for intermittently faulty systems," in *Proc. IEEE Computer Society* 18^{th} *Int'l Symp. Fault-Tolerant Computing*, pp. 260–265, 1988.

[17] D. M. Blough and A. Pelc, "Reliable diagnosis and repair in constant-degree multiprocessor systems," in *Proc. IEEE Computer Society* 20^{th} *Int'l Symp. Fault-Tolerant Computing*, pp. 316–323, 1990.

[18] S. Rangarajan and D. Fussell, "Probabilistic diagnosis algorithms tailored to system topology," in *Proc. IEEE Computer Society* 21^{st} *Int'l Symp. Fault-Tolerant Computing*, pp. 230–237, 1991.

[19] A. Ghafoor and P. Sole, "Performance of fault-tolerant diagnostics in the hypercube systems," *IEEE Trans. Comput.*, vol. C-38, pp. 1164–1172, August 1989.

[20] S. Khanna and W. K. Fuchs, "New algorithms for sequential diagnosis," Tech. Rep. CRHC-92-13, University of Illinois, Center for Reliable and High-Performance Computing, August 1992.

[21] F. P. Preparata and J. Vuillemin, "The cube-connected cycles: A versatile network for parallel computation," *Comm. ACM*, vol. 24, pp. 300–309, May 1981.

[22] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation*. Englewood Cliffs, NJ: Prentice Hall International, 1989.

Biography

Sanjeev Khanna received the M.Sc.(Tech.) degree in computer science and the M.Sc.(Hons.) degree in economics from Birla Institute of Technology and Science, India in 1990, the M.S. degree in computer science from the University of Illinois at Urbana-Champaign in 1992, and the Ph.D. degree in computer science from Stanford University in 1996. He is currently at the Fundamental Mathematics Research Department, Bell Laboratories.

His research interests include design and analysis of algorithms, structural complexity theory, and application of combinatorial optimization techniques to computer systems, databases and robotics.

W. Kent Fuchs (S'83-M'85-SM'90-F'95) received the B.S.E. degree from Duke University, the M. Div. degree from Trinity Evangelical Divinity School, and the Ph.D. in Electrical Engineering from the University of Illinois.

He is currently Head of the School of Electrical and Computer Engineering, Purdue University. He was formerly Professor in the Department of Electrical and Computer Engineering and the Coordinated Science Laboratory, University of Illinois. His research interests include dependable computing and testing. Research awards include the Senior Xerox Faculty Award for Excellence in Research 1993, selection as a University Scholar, 1991, appointment as Fellow in the Center for Advanced Studies, 1990, and the Xerox Faculty Award for Excellence in Research 1987, all from the University of Illinois. He also received the Digital Equipment Corporation Incentives for Excellence Faculty Award 1986-1988 and the Best Paper Award, IEEE/ACM Design Automation Conference (DAC) 1986, simulation and test category.

Dr. Fuchs was the Guest Editor of the 1992 May special issue of IEEE Transactions on Computers on Fault-Tolerant Computing, and Co-Guest Editor of the 1992 April special issue of IEEE's Computer on Wafer-Scale Integration Architectures and Algorithms. He has been a member of the editorial board for IEEE Transactions on Computers and also IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems. He is a Fellow of the IEEE.

Keywords

Analysis of Algorithms

Degree of Diagnosability

Fault-tolerance

Graph Partitioning

Multiprocessor Systems

Sequential Diagnosis

System-Level Diagnosis

Footnotes

∗ Bell Laboratories, Lucent Technologies, 700 Mountain Avenue, Murray Hill, NJ 07974-0636 (sanjeev@theory.stanford.edu).

† School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907-1285.

[1] Throughout this paper, the word hypercube refers to a binary hypercube.

[2] All logarithms used in this paper have base 2.