On a New Class of Codes for Identifying Vertices in Graphs

Mark G. Karpovsky, *Fellow, IEEE*, Krishnendu Chakrabarty, *Member, IEEE*, and Lev B. Levitin, *Fellow, IEEE*

***Abstract*—We investigate a new class of codes for the optimal covering of vertices in an undirected graph G such that any vertex in G can be uniquely identified by examining the vertices that cover it. We define a *ball* of radius t centered on a vertex v to be the set of vertices in G that are at distance at most t from v. The vertex v is then said to *cover* itself and every other vertex in the ball with center v. Our formal problem statement is as follows: Given an undirected graph G and an integer $t \geq 1$, find a (minimal) set C of vertices such that every vertex in G belongs to a unique set of balls of radius t centered at the vertices in C. The set of vertices thus obtained constitutes a code for vertex identification. We first develop topology-independent bounds on the size of C. We then develop methods for constructing C for several specific topologies such as binary cubes, nonbinary cubes, and trees. We also describe the identification of sets of vertices using covering codes that uniquely identify single vertices. We develop methods for constructing optimal topologies that yield identifying codes with a minimum number of codewords. Finally, we describe an application of the theory developed in this paper to fault diagnosis of multiprocessor systems.**

***Index Terms*— Code construction, coding theory, covering radius, fault diagnosis, graph theory, multiprocessor systems.**

I. Introduction

GRAPHS find a wide range of applications in several fields of engineering and information sciences. A graph can be used to represent almost any physical situation and the relationship between various entities. Graph models are therefore often employed in solving a number of practical problems [7].

In this paper, we investigate the problem of covering the vertices of a graph G such that we can uniquely identify any vertex in G by examining the vertices that cover it. We define a *ball* of radius t centered on a vertex v to be the set of vertices of G that are at distance at most t from v. (The distance between vertices v_i and v_j is the number of edges in a shortest path between v_i and v_j.) The vertex v is then said to *cover* itself and every other vertex in its ball. We are interested in identifying the vertices of G using a minimum number of balls of radius t. This is formally stated as follows: Given an undirected graph G and an integer $t \geq 1$, find a (minimal) set C of vertices such that every vertex of G belongs to a unique set of balls of radius t centered at the vertices in C. We view C as an identifying code such that all vertices in it are codewords.

An application of the theory developed in this paper lies in fault diagnosis of multiprocessor systems. The purpose of fault diagnosis is to test the system and locate faulty processors. A multiprocessor system can be modeled as an undirected graph $G = (V, E)$, where V is the set of processors and E is the set of links in the system. Specific software routines are executed on certain selected processors to carry out diagnosis. The selection of these processors is done by generating the code C that allows for unique identification of faulty processors. Every processor corresponding to a codeword vertex tests itself and all its neigbhoring processors. This corresponds to the use of balls of radius one centered at the codewords, i.e., $t = 1$. Hence an optimal code (minimum number of codewords) minimizes the amount of overhead required to implement fault diagnosis.

The organization of the paper is as follows. In Section II, we develop topology-independent bounds on the size of C, and present methods for constructing C for practical topologies such as meshes, binary and nonbinary cubes, and trees. Section III addresses the problem of constructing codes that identify not just single vertices as in the previous sections, but sets of vertices of up to a given size. Finally, in Section IV, we discuss the construction of optimal graphs that yield identifying codes with a minimum number of codewords.

II. Code Construction

Let $M(t)$ be the minimum number of codewords required to identify every vertex uniquely when a ball of radius t is used. We first obtain some lower bounds on $M(t)$. Let $V_i(t)$ be the volume of a ball of radius t centered at vertex v_i, i.e., the number of vertices that are at distance at most t from v_i.

Theorem 1: For a graph with N vertices

1) $M(t) \geq \lceil \log_2(N+1) \rceil$.
2) Let $N/2 \geq V_1(t) \geq V_2(t) \geq \cdots \geq V_N(t)$. Then $M(t) \geq K$, where K is the smallest integer such that
$$\sum_{i=1}^{K} h\left(\frac{V_i(t)}{N+1}\right) \geq \log_2(N+1)$$
where $h(x) = -x \log_2 x - (1-x)\log_2(1-x)$ is the binary entropy function.
3) Let K be the smallest integer such that for a certain l $(1 \leq l \leq \min(K, V_1(t))$, the following conditions are

Manuscript received November 5, 1996; revised August 7, 1997. This work was supported in part by the National Science Foundation under Grant MIP 9630096, by NATO under Grant 910411, and by a start-up grant from Boston University's College of Engineering.

The authors are with Reliable Computing Laboratory, Department of Electrical and Computer Engineering, Boston University, Boston, MA 02215 USA.

Publisher Item Identifier S 0018-9448(98)00841-4.

satisfied:

$$N \le \sum_{j=1}^{l-1} \binom{K}{j} + \left\lfloor \frac{1}{l} \left(\sum_{i=1}^{K} V_i(t) - \sum_{j=1}^{l-1} j \binom{K}{j} \right) \right\rfloor. \tag{1}$$

$$\sum_{j=1}^{l-1} j \binom{K}{j} < \sum_{i=1}^{K} V_i(t) \le \sum_{j=1}^{l} j \binom{K}{j}. \tag{2}$$

Then $M(t) \ge K$.

Proof: The first lower bound follows from the fact that there are $N+1$ cases (N different vertices and the selection of no vertex—no vertex is identified) to be distinguished. Therefore, the information can be encoded in a minimum of $\lceil \log_2(N+1) \rceil$ bits.

We prove the second bound as follows. We denote by X_i ($i = 1, 2, \cdots, K$) the result of the (identification) test performed by the ith codeword. Each X_i is a binary random variable: $X_i = 0, 1$. Denote by Y the random variable which is equal to 0 when no vertex is to be identified and j ($j = 1, 2, \cdots, N$) if the jth vertex is to be identified. In the absence of any *a priori* knowledge, we assume that all $N+1$ cases are equiprobable. Thus the entropy $H(Y) = \log_2(N+1)$. Now, denote by $I(X_i; Y/X_1, \cdots, X_{i-1})$ the information in X_i about Y given the outcomes of $X_1, X_2, \cdots, X_{i-1}$. Then

$$\sum_{i=1}^{K} I(X_i; Y/X_1, \cdots, X_{i-1}) = H(Y) = \log_2(N+1). \tag{3}$$

On the other hand,

$$I(X_i; Y/X_1, \cdots, X_{i-1}) = H(X_i/X_1, \cdots, X_{i-1}) - H(X_i/Y, X_1, \cdots, X_{i-1}). \tag{4}$$

Since $H(X_i/Y, X_1, \cdots, X_{i-1}) = 0$ (the value of X_i is uniquely determined by Y), we obtain

$$I(X_i; Y/X_1, \cdots, X_{i-1}) = H(X_i/X_1, \cdots, X_{i-1}) \le H(X_i). \tag{5}$$

However, the probability $\Pr\{X_i = 1\} = V_i(t)/(N+1)$, hence

$$H(X_i) = h\left(\frac{V_i(t)}{N+1}\right). \tag{6}$$

It follows from (4)–(6) that the number of codewords is not smaller than the smallest K such that $\Sigma_{i=1}^{K} h(V_i(t)/N+1) \ge \log_2(N+1)$.

To prove the third bound, we consider a $K \times N$ binary matrix $A = \|a_{kn}\|, k = 1, 2, \cdots, K; n = 1, 2, \cdots, N$, where $a_{kn} = 1$ if the kth codeword covers the nth vertex, and $a_{kn} = 0$ otherwise. Denote by $w_k^{(r)}$ the weight (number of nonzero components) of the kth row and by $w_n^{(c)}$ the weight of the nth column. Obviously,

$$\sum_{n=1}^{N} w_n^{(c)} = \sum_{k=1}^{K} w_k^{(r)}. \tag{7}$$

Our goal is to find the minimum K for a given N and $\{V_i(t)\}, i = 1, \cdots, N$, provided that all columns of A are nonzero and distinct.

Consider the dual problem: for a given K, find the maximum number N of distinct and nonzero columns. Since, obviously,

$$\sum_{k=1}^{K} w_k^{(r)} \le \sum_{i=1}^{K} V_i(t),$$

it follows that

$$\sum_{n=1}^{N} w_n^{(c)} \le \sum_{i=1}^{K} V_i(t). \tag{8}$$

To maximize the number of columns N under the constraint (8), we have to choose the weights of the columns $w_n^{(c)}$ as small as possible, starting with columns of weight 1, 2, etc., up to the point where the right-hand side of (8) is exceeded. Let l be an integer such that

$$\sum_{j=1}^{l-1} j \binom{K}{j} < \sum_{i=1}^{K} V_i(t) \le \sum_{j=1}^{l} j \binom{K}{j}. \tag{9}$$

For the maximum possible number of columns N, (8) should turn into an equality. Taking into account (9), let

$$\sum_{i=1}^{K} V_i(t) = \sum_{j=1}^{l-1} j \binom{K}{j} + ml + g \tag{10}$$

where $0 \le g \le l-1, 0 < ml + g \le l \binom{K}{l}$. Obviously,

$$m = \left\lfloor \frac{1}{l} \left(\sum_{i=1}^{K} V_i(t) - \sum_{j=1}^{l-1} j \binom{K}{j} \right) \right\rfloor. \tag{11}$$

We need to consider the following three cases.

1) $m < \binom{K}{l}, g + m \le \binom{K}{l}$.

Then the largest number of columns N and the equality in (8) are achieved if we use all possible distinct columns of weights $1, 2, \cdots, l-2$, $\binom{K}{l-1} - g$ columns of weight $l-1$, and $m+g$ columns of weight l.

2) $m < \binom{K}{l}, g + m > \binom{K}{l}$.

To maximize N and to achieve the equality in (8), we should use all columns of weights $1, 2, \cdots, l-2$, $\binom{K}{l-1} - g$ columns of weight $l-1$, all $\binom{K}{l}$ columns of weight l, and $m + g - \binom{K}{l}$ columns of weight $l+1$.

3) $m = \binom{K}{l-1}, g = 0$.

Then we should use all columns of weights $1, 2, \cdots, l$.

In all cases, the total number of columns in A is

$$N = \sum_{j=1}^{l-1} \binom{K}{j} + m = \sum_{j=1}^{l-1} \binom{K}{j} + \left\lfloor \frac{1}{l} \left(\sum_{i=1}^{K} V_i(t) - \sum_{j=1}^{l-1} j \binom{K}{j} \right) \right\rfloor. \tag{12}$$

In fact, because of (8), (12) gives an upper bound on the number N of columns for a given number K of rows (where l is defined by (9)). It follows that for a given N, the minimum number of rows K should satisfy (1). Thus (1) and (2) together determine a lower bound on the number of codewords for a given number N of vertices. □

In the special case of a regular graph where $V_i(t) = V(t)$ for all i, (9) and (12) take simpler forms

$$\sum_{j=0}^{l-2} \binom{K-1}{j} < V(t) \leq \sum_{j=0}^{l-1} \binom{K-1}{j}. \tag{13}$$

$$N \leq \sum_{j=1}^{l-1} \binom{K}{j} + \left\lfloor \frac{K}{l} \left(V(t) - \sum_{j=0}^{l-2} \binom{K-1}{j} \right) \right\rfloor. \tag{14}$$

A simpler lower bound in the case of a regular graph is given by Theorem 2.

Theorem 2: The size of an identifying code for a regular graph with N vertices is lower-bounded by

$$M(t) \geq \frac{2N}{V(t)+1}. \tag{15}$$

Proof: As in the proof of Theorem 1, consider the $K \times N$ binary matrix $\|a_{kn}\|$, where $a_{kn} = 1$ if and only if the kth codeword covers the nth vertex, and $a_{kn} = 0$ otherwise. The number of nonzero elements in the matrix is obviously $KV(t)$. On the other hand, since at most K columns can have weight 1 and the remaining $N - K$ columns must be of weight at least 2, the number of nonzero elements should be at least $K + 2(N - K) = 2N - K$. Hence, $KV(t) \geq 2N - K$. Therefore,

$$M(t) \geq K \geq \frac{2N}{V(t)+1}. \quad \square$$

The lower bound (15) is, in general, weaker than (13) and (14). However, both bounds coincide if $l \leq 2$. It can be shown in Theorem 1 (part 3), that $l \leq 2$ if and only if $V(t) \leq \sqrt{2N}$. The latter condition is satisfied for a broad class of graphs if t does not grow too fast with N.

We next examine some specific graph topologies.

A. Binary Cubes

A binary n-cube computer is a multiprocessor system with $N = 2^n$ processors interconnected as an n-dimensional binary cube. Each processor P_i constitutes a node of the cube and is a self-contained computer with its own CPU and local memory. Each P_i also has direct communication paths to n other neighbor processors through the edges of the cube. An example of a commercial binary-cube computer is the NCUBE/ten, which is a ten-dimensional system developed by NCUBE Corporation [8], [14].

Let $M_n(t)$ be the minimum number of codewords required for identifying the vertices in an n-dimensional binary cube using balls of radius t. We first consider the case $t = 1$. The specific topology of the n-dimensional cube imposes additional constraints which makes the lower bounds of Theorems 1 and 2 unattainable. A tighter lower bound is given by the following theorem, a proof of which is given in the Appendix.

Theorem 3: For an n-dimensional binary cube, $n \geq 3$,

$$M_n(1) \geq \frac{n \cdot 2^n}{V(2)} = \frac{n \cdot 2^{n+1}}{n(n+1)+2} \tag{16}$$

where $V(2) = 1 + n + \binom{n}{2}$ is the volume of the ball of radius two in the Hamming space $\mathcal{Z}_2^n$.

The lower bound (16) is achieved if there exists a perfect covering of the n-dimensional cube by balls of radius two, i.e., a perfect code[1] with distance five. The only such case is for $n = 5$. Then all vertices of weight one and four can be chosen as codewords, and the total number of codewords is ten, which is given by (16).

Let $K(n, q)$ be the size of an optimal code[2] C^* of length n with covering radius q, i.e., every vertex is at Hamming distance at most q from a codeword of C^* [4], [5], [9], [12], [17], [21]. An upper bound on $M_n(1)$ follows from the theorem below.

Theorem 4: Let C^* be an optimal binary code of length n and covering radius 2, i.e., C^* has $K(n, 2)$ codewords. Then, for $t = 1$, a code C identifying vertices in the n-dimensional binary cube can be selected as $C = \{w | \exists v \in C^*, d(v, w) = 1\}$ ($d(v, w)$ is the Hamming distance between v and w).

Proof: We show that every vertex in the cube G is covered by a unique set of codewords.

Case 1:

Let $v \in C^*$. Every neighbor of v belongs to C and therefore covers v. We need to prove that there exists no vertex v' that is covered by the same set of vertices. Let $v = (v_1, v_2, \cdots, v_n)$. The codewords covering v are

$$(\overline{v}_1, v_2 \cdots, v_n), (v_1, \overline{v}_2 \cdots, v_n), \cdots, (v_1, v_2, \cdots, \overline{v}_n)$$

where $v_i \in \{0, 1\}$ and $\overline{v}_i = 1 - v_i$. Clearly, v is the only vertex that is a neighbor of all these codewords.

Case 2:

Let $v \in C$ and $v \notin C^*$. Now, v is covered by itself and every neighbor $v' \in C$. We show that there exists $v'' \in C$ such that $d(v, v'') = 1$ but $d(v', v'') > 1$. Note that because there are no triangles in G, there does not exist a vertex u such that $d(u, v) = d(u, v') = 1$. Hence, we have only to prove the existence of v'' for every v. Let $x \in C^*$ be a neighbor of v'. Since $d(v, x) = 2$, there exist exactly two vertices that are at distance one from both x and v. One of these vertices (from above) is v'; the other is v''.

Case 3:

Let $v \notin C$ and $v \notin C^*$. Suppose v is covered by l codewords in C ($l > 2$). Without loss of generality, let $v = (0, 0, \cdots, 0)$, and the codewords covering v be

$$(1, 0, \cdots, 0), (0, 1, 0, \cdots, 0), \cdots, (\underbrace{0, 0, \cdots, 0, 1}_{l}, 0, \cdots, 0).$$

[1] A binary $(n, k, 2l + 1)$ code is perfect if $\Sigma_{i=1}^{l} \binom{K}{i} = 2^{n-k}$.

[2] An optimal covering code is one that has a minimum number of codewords.

Clearly, there is no vertex other than v that is at distance one from all these codewords. Hence v is uniquely identified.

Next, suppose $l = 1$. Without loss of generality, let $v = (0, 0, \cdots, 0)$, and the only codeword v' covering v be $(1, 0, \cdots, 0)$. There must exist at least one vertex w in the covering code C^* such that $d(w, v') = 1$. Thus

$$w \in \{(1,1,0,\cdots,0),(1,0,1,0,\cdots,0),\cdots,(1,0,\cdots,0,1)\}.$$

It can now be easily seen that each vertex in the above set contributes codewords to C that cover v, which contradicts the assumption that $l = 1$.

Finally, suppose $l = 2$. Let the two codewords covering v be $(1,0,\cdots,0)$ and $(0,1,0,\cdots,0)$. Now, $w = (1,1,0,\cdots,0)$ is the only vertex other than v that is at distance one from both these codewords. If $w \in C^*$ then w is uniquely identified. If $w \notin C^*$ then there exists $w^* \in C^*$ that generates $(1,0,\cdots,0)$ as a codeword of C. Once again, without loss of generality, let $w^* = (1,0,1,0,\cdots,0)$. Then $(0,0,1,0,\cdots,0) \in C$ which contradicts our assumption that $l = 2$. Hence $w \in C^*$ and v is uniquely identified. □

Corollary 1: The number of codewords in an optimal identifying code with $t = 1$ for a binary n-cube $(n \geq 3)$ is upper-bounded by

$$M_n(1) \leq nK(n, 2). \tag{17}$$

Exact values for small n as well as bounds on $K(n, 2)$ are available in the literature; see, e.g., [3]. In particular

$$K(n,2) \leq K(\lfloor n/2 \rfloor, 1)K(\lceil n/2 \rceil, 1).$$

Using this and (16), we get

$$\frac{n \cdot 2^n}{V(2)} \leq M_n(1) \leq nK(\lfloor n/2 \rfloor, 1)K(\lceil n/2 \rceil, 1).$$

For example, if $m = 2^s - 1$, then $K(m, 1) = 2^m/m + 1$. Therefore, for $n = 2^s - 2$ we have

$$\frac{n \cdot 2^{n+1}}{n(n+1)+2} \leq M_n(1) \leq \frac{n \cdot 2^{n+2}}{(n+2)^2}.$$

The ratio r_n of the upper bound to the lower bound

$$r_n = \frac{2(n^2+n+2)}{(n+2)^2} \rightarrow 2$$

with $n \rightarrow \infty$.

Another solution to the identifying code construction problem for an n-dimensional binary cube is obtained by selecting codewords separately for its two constituent $(n-1)$-dimensional cubes. This "divide and conquer" approach, which implies that $M_n(1) \leq 2M_{n-1}(1)$, often gives better results for small n than the construction method using $K(n, 2)$ (see Table I). Note that for $n = 3$ and $n = 4$, we achieved the lower bound on $M_n(1)$ using *ad hoc* construction methods.

The construction of Theorem 4 can be extended in a straightforward manner for $t > 1$. We now construct an optimal C^* with covering radius $2t$; the number of codewords in C^* is $K(n, 2t)$. The identifying code C is generated by selecting vertices that are at distance exactly t from the vertices in C^*.

TABLE I
NUMBER OF CODEWORDS REQUIRED FOR IDENTIFYING VERTICES IN BINARY CUBES

n	Lower bound bound on M_1	$M_1(n)$ (using $K(n,2)$)	$M_1(n)$ (divide and conquer)
3	4*	6	—
4	6*	8	—
5	10	10	—
6	16	24	20
7	29	49	40
10	177	300	320
16	7282	14336	20480

* Lower bound attained by construction.

Theorem 5: For any given $t < n/2$, a code C for identifying vertices in the n-dimensional binary cube $(n > 2)$ can be obtained by selecting as codewords all vertices at distance exactly t from the codewords of an optimal code C^* which has covering radius $2t$, i.e., $C = \{x | \exists u \in C^*, d(x, u) = t\}$.

Proof: We first make the following observation: if vertices v_1 and v_2 are such that there is at least one ball centered at a vertex in C^* to which v_1 (v_2) belongs but v_2 (v_1) does not belong, then v_1 and v_2 can be distinguished using codewords from C. Therefore, we only need to prove that any two vertices can be distinguished if they belong to the same ball of radius $2t$ centered at a vertex $u \in C^*$.

Without loss of generality, let

$$u = \underbrace{(0, 0, \cdots, 0)}_{n}.$$

All vertices of weight t now belong to C and serve as codewords for identifying a vertex. Given two vertices v_1 and v_2 that are in the same ball centered at u, we show that we can always find a codeword $x \in C$ such that x covers one of them but not the other. We define $z = x \cdot y$ to be a vector with components $z_i = x_i y_i$. In addition, $y \leq x$ if $x \cdot y = y$, and $\overline{y}$ is the component-wise negation of y. Let $w(v_1) = l_1$ and $w(v_2) = l_2$, where $w(v)$ is the weight of v. Assume, without loss of generality, that $l_1 \leq l_2$. It follows therefore that $l_2 \geq 1$ because otherwise both v_1 and v_2 will be the same vertex $(0, 0, \cdots, 0)$.

Consider now three cases:

1) $w(v_1 \cdot \overline{v_2}) \geq t$. We choose $x \in C$ $(w(x) = t)$ such that $x \leq v_1 \cdot \overline{v}_2$. Then $d(x, v_1) = l_1 - t \leq t$, and $d(x, v_2) = t + l_2 > t$.

2) $w(v_1 \cdot \overline{v_2}) = l_3 < t$. Note that if $l_3 = 0$ then $l_2 > l_1$, otherwise, v_1 and v_2 would be identical. Assume that at least one of two conditions is fulfilled: $l_3 > 0$ or l_1 is even. Choose $x \in C$ such that

$$v_1 \cdot \overline{v}_2 \leq x, \; w(x \cdot v_1 \cdot v_2) = \max\{0, \lceil l_1/2 \rceil - l_3\}$$

and

$$w(x \cdot \overline{v_1} \cdot \overline{v}_1 \cdot \overline{v}_2) = \min\{n - l_2 - l_3, t - \lceil l_1/2 \rceil\} = l_4.$$

Then $w(x \cdot v_1) = \lceil l_1/2 \rceil$ and $d(x, v_1) = t + l_1 - 2\lceil l_1/2 \rceil \leq t$. On the other hand,

$$d(x, v_2) = t + l_2 - 2(t - l_3 - l_4) > t.$$

Thus in both cases, codeword x covers v_1 but not v_2.

TABLE II
BOUNDS ON THE NUMBER OF CODEWORDS IN AN OPTIMAL IDENTIFYING CODE WITH $t = 2$ FOR AN n-DIMENSIONAL BINARY CUBE

n	$V(2)$	Lower bound on $M_n(2)$	Upper bound on $K(n,4)$	Upper bound on $M_n(2)$ using $K(n,4)$	Upper bound on on $M_n(2)$ using (18)
4	11	5†	1	—	9
5	16	6†	2	—	12
6	22	8†	2	—	16
8	37	17†	2	56	36
9	46	26†	2	72	80
12	79	104‡	12	792	400
16	137	950‡	64	7680	6400
20	211	9893‡	512	97220	90000

†Lower bound from Theorem 1 (part 3), ‡Lower bound from (15)

3) $w(v_1 \cdot \overline{v}_2) = 0$ and l_1 is odd. Then $v_1 \leq v_2$ and at least one of two conditions is fulfilled: l_1 is even or $l_2 - l_1 \geq 2$. Choose $x \in C$ such that $w(x \cdot v_2) = \lceil l_2/2 \rceil$ and

$$w(x \cdot v_1) = \max\{0, \lceil l_2/2 \rceil - l_2 + l_1\}.$$

Then

$$d(x, v_2) = t + l_2 - 2\lceil l_2/2 \rceil \leq t$$

and

$$d(x, v_1) = t + l_1 - 2\max\{0, \lceil l_2/2 \rceil - l_2 + l_1\} \geq t.$$

Thus in this case, codeword x covers v_2 but not v_1. □

Corollary 2: For $t < n/2$, the number of codewords required for identifying vertices in a binary cube is upper-bounded by

$$M_n(t) \leq K(n, 2t)\binom{n}{t}.$$

We next estimate the ratio r_n between the upper bound and the lower bound on the number of codewords $M_n(t)$ when $n/2t$ is an integer. We know from (15) and Corollary 2 that

$$\frac{2^{n+1}}{V(t)+1} \leq M_n(t) \leq \binom{n}{t} K(n, 2t).$$

Since $K(n_1+n_2, t_1+t_2) \leq K(n_1, t_1)K(n_2, t_2)$ [3], it follows that

$$\frac{2^{n+1}}{1 + \sum_{i=0}^{t} \binom{n}{i}} \leq M_n(t) \leq \binom{n}{t} (K(n/2t, 1))^{2t}.$$

Using the following well-known upper bound on $K(q, 1)$ (see [3]):

$$K(q, 1) \leq \frac{2^q}{2^{\lfloor \log_2(q+1) \rfloor}}$$

we obtain

$$\frac{2^{n+1}}{1 + \sum_{i=0}^{t} \binom{n}{i}} \leq M_n(t) \leq \binom{n}{t} \frac{2^n}{2^{2t\lfloor \log_2(n/2t+1) \rfloor}}.$$

If $n/2t + 1 = 2^s$, then using

$$V(t) = \sum_{i=0}^{t} \binom{n}{i} \sim n^t/t!$$

for $n \rightarrow \infty$ and constant t, we get[3]

$$2^{n+1} t! n^{-t} \lesssim M_n(t) \lesssim (2t)^{2t} 2^n n^{-t} (t!)^{-1}.$$

The ratio r_∞ of the upper bound to the lower bound ($n \rightarrow \infty$) is therefore given by

$$r_\infty = 2^{2t-1} t^{2t} (t!)^{-2}.$$

For $t = 1$, we have $r_\infty = 2$ as before, while for $t = 2$, $r_\infty = 32$.

For the special case of $n = (4s+1)t, s \geq 1$, we have the following corollary, which follows from the fact that

$$K(r(2s+1), rs) \leq (K(2s+1, s))^r = 2^r.$$

Corollary 3: The number of codewords required for a binary cube with $(4s+1)t$ dimensions using balls of radius st is upper-bounded by

$$M_{(4s+1)t}(st) \leq \binom{(4s+1)t}{st} 2^t.$$

As special cases, for $s = 1$, we have $M_{5t}(t) \leq \binom{5t}{t} 2^t$, and for $s = 2$, we have $M_{9t}(2t) \leq \binom{9t}{2t} 2^t$.

Table II shows the upper and lower bounds on $M_n(2)$. For the lower bounds, we used (15) for $n > 9$ since $V(2) \leq \sqrt{2N}$ for these cases, and (15) coincides with the bound given by Theorem 1 (part 3). For $n \leq 9$, we applied Theorem 1 (part 3) directly and obtained tighter bounds than given by (15). For $n \leq 4$, the covering radius approach cannot be applied with $t = 2$. The last column of the table is based on the following result, which we prove later (see Corollary 7):

$$M_n(2) \leq M_{\lfloor n/2 \rfloor}(1) \cdot M_{\lceil n/2 \rceil}(1). \quad (18)$$

While it may be intuitively expected that the number of codewords required for identification decreases as t increases, this is not necessarily the case. For example, $M_3(2) = 7$ but $M_3(1) = 4$.

[3] $a(n) \sim b(n) \leftrightarrow \lim_{n \rightarrow \infty} a(n)/b(n) = 1.$

B. Nonbinary Cubes

The next topology that we examine is a nonbinary cube, which finds several applications in parallel processing. A p-ary n-dimensional cube has p^n processors and each processor is connected to its $2n$ neighbors. (Every processor has two neighbors in each dimension.) Similar practical architectures include two-dimensional rectangular meshes such as Intel's Paragon architecture [10] and three-dimensional meshes such as the MIT-Intel J-machine [6].

We next consider codeword selection for the identification of vertices in n-dimensional p-ary cubes. Every vertex in this case can be assigned a coordinate vector $(x_1, x_2, \cdots, x_n)$ of length n, where $0 \le x_i \le p-1$. Two vertices

$$x = (x_1, x_2, \cdots, x_n)$$

and

$$x' = \{x'_1, x'_2, \cdots, x'_n\}$$

are neighbors if

$$x - x' = (0, 0, \cdots, \pm 1, 0, \cdots, 0) (\text{mod } p).$$

Let $P(x) = (p_1, p_2, \cdots, p_n)$ be the parity vector corresponding to $(x_1, x_2, \cdots, x_n)$ such that $p_i = 0$ (1) if x_i is even (odd). For a p-ary code $\mathcal{C}$, let $\mathcal{P}(\mathcal{C}) = \{P(x) | x \in \mathcal{C}\}$ be the binary parity code with codewords $(p_1, p_2, \cdots, p_n)$.

We use $M_n^{(p)}(t)$ to denote the number of codewords required to identify vertices in a p-ary n-cube using balls of radius t $(t < n)$. (For the binary case $p = 2$, we had omitted the superscript.) First we examine the construction of the identifying code $\mathcal{C}$ for $t = 1$.

Theorem 6: For an n-dimensional p-ary cube ($n = 2^s - 1$, p even and $p > 4$), vertex identification is achieved with a smallest possible number of codewords, i.e., $M_n^{(p)}(1) = p^n/(n+1)$, if and only if the identifying code $\mathcal{C}$ consists of all codewords such that their parity vectors form the perfect binary $(n, n-s, 3)$ code.

Proof: We first prove that every vertex is covered by a unique combination of codewords. Every codeword is covered only by itself because the Hamming distance between any two parity vectors of codewords is at least three. Next consider a noncodeword vertex with coordinates $(x_1, x_2, \cdots, x_n)$ and corresponding parity vector $(p_1, p_2, \cdots, p_n)$. There are two vertices with coordinates

$$x' = (x'_1, x'_2, \cdots, x'_n)$$

and

$$x'' = (x''_1, x''^2, \cdots, x''^n)$$

such that they have the same parity vector $(q_1, q_2, \cdots, q_n)$, x' and x'' are neighbors of x in the n-dimensional nonbinary cube, $(q_1, q_2, \cdots, q_n)$ belongs to the code $\mathcal{C}$, and the Hamming distance between $(p_1, p_2, \cdots, p_n)$ and $(q_1, q_2, \cdots, q_n)$ is one. We note that for $p > 4$, x is uniquely determined by x' and x''.

To prove necessity, we note that if two vertices in the p-ary n-dimensional cube are neighbors, their parity vectors are at distance 1. Thus for an identifying code, the covering radius of the set of parity vectors must be equal to 1, and the smallest set with this property is a perfect $(n, n-s, 3)$ code. □

For the important case of the three-dimensional p-ary cube, we have the following useful corollary, obtained from the above theorem with $n = 3$.

Corollary 4: For a three-dimensional p-ary cube (p even and $p > 4$), optimal codeword selection ($M_3^{(p)}(1) = p^3/4$) is achieved if and only if the vertices with parity vectors $(0, 0, 0)$ and $(1, 1, 1)$ are chosen as codewords.

Theorem 6 and Corollary 4 show that the density of codewords is only 0.25 for three-dimensional cubes, and tends to zero as n increases. The next theorem is a generalization of Theorem 6 for arbitrary n.

Theorem 7: Let C^* be an optimal binary code of length n and covering radius one. Then $\mathcal{C}$ is an optimal p-ary (p even, $p > 4$) identifying code for a p-ary n-dimensional cube if and only if $\mathcal{C}$ consists of all vectors such that their parity vector code $P(\mathcal{C}) = C^*$.

The proof of the theorem is similar to the proof of Theorem 6; the only difference being that the perfect $(n, n-s, 3)$ code is now replaced by an optimal binary code with covering radius one.

Corollary 5: For an n-dimensional ($n = 2^s$), p-ary (p even, $p > 4$) cube

$$\frac{p^n}{n+1} \le M_n^{(p)}(1) \le \frac{p^n}{n}. \tag{19}$$

Proof: The lower bound follows from (15). The upper bound follows from Theorem 7 since $K(2^s, 1) = 2^{2^s - s} = 2^n/n$. □

Note that the above construction is not the best for all values of n. For example, if we apply this construction to the case $n = 2$, then $C^* = \{00, 11\}$ and we obtain a set of $p^2/2$ codewords in a "checkerboard" pattern, implying a codeword density of 0.5. However, the following theorem gives a better construction for $n = 2$.

Thoerem 8: Let $K^{(p)}(n, 2)$ be a minimal number of codewords in a p-ary n-dimensional code with covering radius 2 in the Lee metric [17]. Then for any $p > 4$

$$M_1^{(p)}(n) \le (2n+1)K^{(p)}(n, 2). \tag{20}$$

Proof: To prove (20), it is sufficient to show that all vertices in a Lee ball B_2 of radius 2 with center v can be identified by balls of radius 1 centered at all vertices that belong to the ball B_1 of radius 1 centered at v. Without loss of generality, we can assume that $v = (0, 0, \cdots, 0)$. Then

$$B_1 = \{(0, 0, \cdots, 0)\} \bigcup \{(0, \cdots, 0, \pm 1, 0, \cdots, 0) \ (\text{mod } p)\}$$

and

$$B_2 = B_1 \bigcup \{(0, \cdots, 0, \pm 2, 0, \cdots, 0) \ (\text{mod } p)\} \bigcup \{(0, \cdots, \pm 1, 0, \cdots, 0, \pm 1, 0, \cdots, 0) \ (\text{mod } p)\}.$$

Let $x \in B_2$. We have to consider the following four cases:

1) $x = (0, \cdots, 0)$. Then x belongs to all balls of radius 1 with centers in B_1.

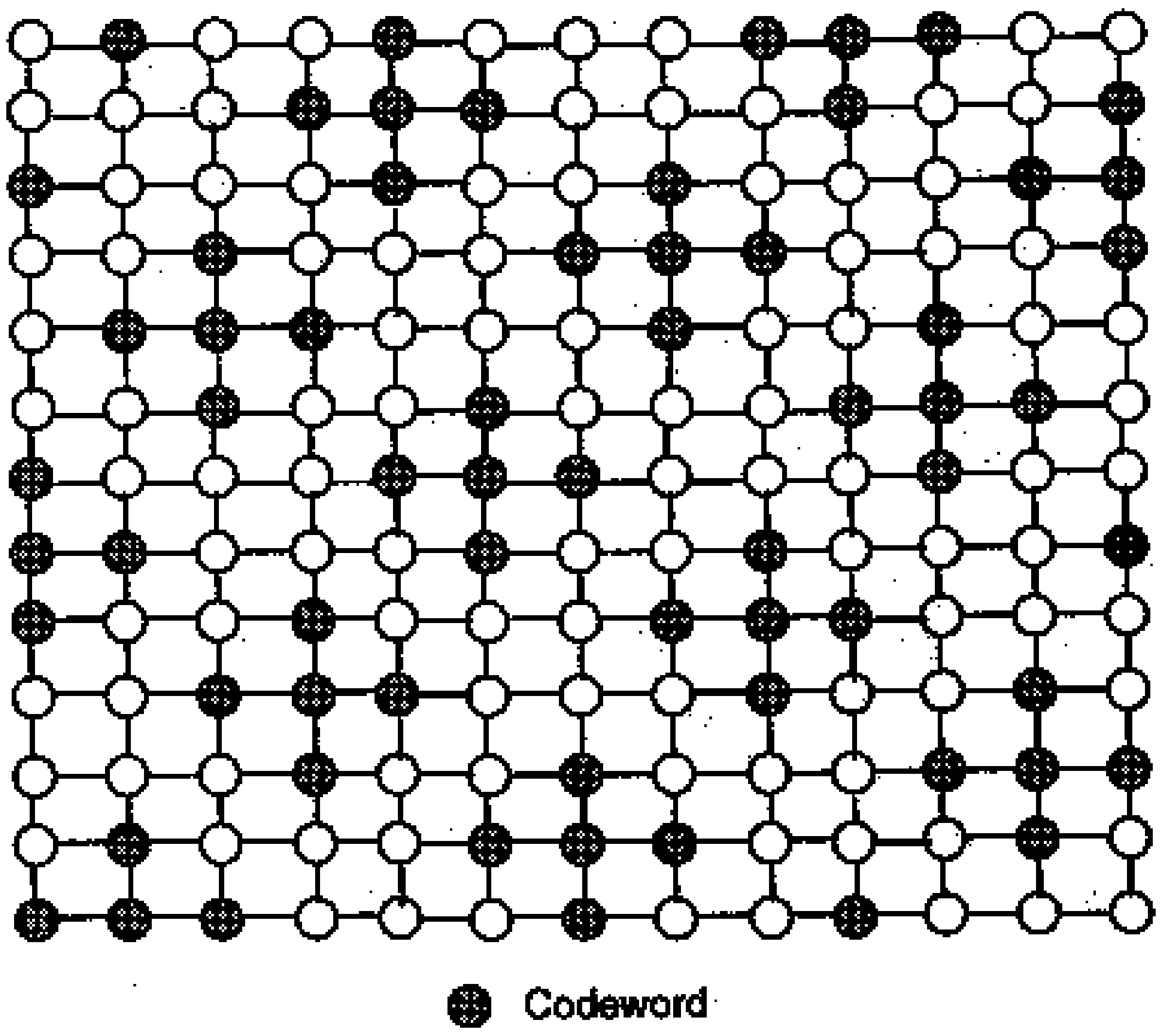


Fig. 1. Identifying code for $n = 2, p = 13$ constructed using Theorem 8. The edges wrap around.

2) $x = (0, \cdots, 0, \pm 1, 0, \cdots, 0)$. Then x belongs to two balls of radius 1 with centers at x and $(0, \cdots, 0)$, respectively.

3) $x = (0, \cdots, 0, \underbrace{\pm 1}_{i}, 0, \cdots, \underbrace{\pm 1}_{j}, 0, \cdots, 0)$.

Then x belongs to two balls with centers

$$(0, \cdots, 0, \underbrace{\pm 1}_{i}, 0, \cdots, 0) \quad \text{and} \quad (0, \cdots, 0, \underbrace{\pm 1}_{j}, 0, \cdots, 0).$$

4) $x = (0, \cdots, 0, \underbrace{\pm 2}_{i}, 0, \cdots, 0)$.

Then x belongs to one ball with center $(0, \cdots, 0, \underbrace{\pm 1}_{i}, 0, \cdots, 0)$.

This completes the proof. □

Corollary 6: Let $n = 2$ and $p = 13s$. Then

$$M_2^{(p)}(1) \leq \frac{5}{13} p^2. \tag{21}$$

Proof: The proof follows from the fact that $|B_1| = 5, |B_2| = 13$, and $K^{(p)}(3, 2) = p^2/13$. □

Fig. 1 shows that construction given by Theorem 8 for $n = 2$ and $p = 13$. ($K^{(13)}(2, 2) = 13, M_2^{(13)}(1) = 65$.) However, the above construction is not optimal for $n = 2$. Fig. 2 shows the best known construction for $n = 2$ and $p = 8s$.

We next turn to the code construction problem when balls of radius greater than one are used. The following theorem provides a powerful "divide-and-conquer" technique for determining $M_n^{(p)}(t)$ for $t > 1$. (Note that $M_t^{(2)}(t)$ is not defined.)

Theorem 9: The number of codewords required to identify vertices in a p-ary n-dimensional cube is given by

$$M_n^{(p)}(t) \leq M_a^{(p)}(s) \cdot M_{n-a}^{(p)}(t - s)$$

where $0 \leq t < n, 0 \leq s \leq t, 0 \leq s \leq a, 0 \leq t - s \leq n - a$, and $1 \leq a \leq n - 1$.

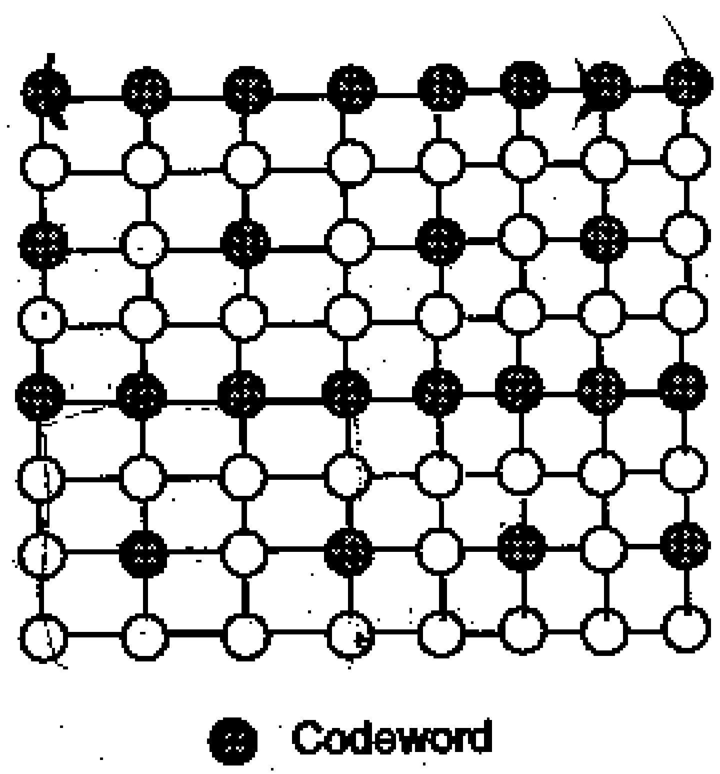


Fig. 2. Identifying code for $n = 2, p = 8s$ with $M_2^{(p)}(1) = \frac{3}{8}p^2$. The construction is repeated with period 8 and wrapped around.

Proof: Let $x = (x_1, x_2)$ and $y = (y_1, y_2)$ be vectors of length n, where $x_1(y_1)$ and $x_2(y_2)$ are of length a and $n - a$, respectively. Let $v = (v_1, v_2)$ be a vector of length n such that $v_1(v_2)$ covers $x_1(x_2)$ but not $y_1(y_2)$ with a ball of radius $s(t - s)$ centered at it. Then $D(v_1, x_1) \leq s$ and $D(v_2, x_2) \leq t - s$, where $D(x, y)$ is the distance between vertices x and y in the p-ary nonbinary cube. This implies that

$$D(v, x) = D(v_1, x_1) + D(v_2, x_2) \leq t.$$

Hence v covers x with a ball of radius t. Now, $D(v_1, y_1) > s$ and $D(v_2, y_2) > t - s$, which implies that $D(v, y) > t$. Thus v does not cover y with a ball of radius t. Therefore, the identifying code $\mathcal{C}(n, t)$ for an n-dimensional p-ary cube can be constructed using the identifying codes for the smaller a and $n - a$ dimensions in the following way:

$$\mathcal{C}(n, t) = \{(x, y) | x \in \mathcal{C}(a, s), y \in \mathcal{C}(n - a, t - s)\}$$

and $M_n^{(p)}(t) \leq |\mathcal{C}(n, t)|$. □

Corollary 7: As a special case of Theorem 9, we have

$$M_n^{(p)}(t) \leq p^a \cdot M_{n-a}^{(p)}(t)$$

where $n - a \geq t$.

Proof: From Theorem 9, we have

$$M_n^{(p)}(t) \leq M_a^{(p)}(0) \cdot M_{n-a}^{(p)}(t) = p^a \cdot M_{n-a}^{(p)}(t).$$

(When $t = 0$, every vertex in the p-ary a-dimensional cube must be selected as a codeword.) □

Corollary 8: For any $p \geq 2$ and $t < n$, we have

$$M_n^{(p)}(t) \leq M_{\lfloor n/2 \rfloor}^{(p)}(\lfloor t/2 \rfloor) \cdot M_{\lceil n/2 \rceil}^{(p)}(\lceil t/2 \rceil).$$

Corollary 9: The following upper bounds exist on the number of codewords in optimal identifying codes for binary and nonbinary cubes.

1) $M_{2t}^{(p)}(t) \leq 0.4^t \cdot p^{2t}; p = 5s$.
2) $M_{2kt}^{(p)}(t) \leq 0.4^t \cdot p^{2kt}$, for any $k > 0, p = 5s$.
3) $M_{(2^s-1)t}^{(p)}(t) \leq (p^{2^s-1}/2^s)^t$, for any $s > 0$, and even $p > 4$.
4) $M_{2t}^{(2)}(t) \leq 0.75^t \cdot 2^{2t}$.
5) $M_{3t}^{(2)}(t) \leq 0.5^t \cdot 2^{3t}$.

6) $M^{(2)}_{5t}(t) = (5/16)^t \cdot 2^{5t}$.
7) $M^{(2)}_{4t}(t) \le (0.125)^t \cdot 2^{4t}$.

Proof: To prove part 1), we note from Theorem 9 that

$$M^{(p)}_{2t}(t) \le (M^{(p)}_2(1))^t = 0.4^t p^{2t}.$$

Hence the density of codewords in a p-ary cube with $2t$ dimensions is at most 0.4^t, and decreases with an increase in t. Part 2) follows directly from Corollary 7 and part 1). To prove 3), we use the result

$$M^{(p)}_{(2^s-1)t}(t) \le (M^{(p)}_{2^s-1}(1))^t = (p^{2^s-1}/2^s)^t.$$

For even $p > 4$ and $s = 2$, we have $M^{(p)}_{3t}(t) \le 0.25^t \cdot p^{3t}$. The proofs of 4) and 5) are similar, but using optimal code constructions with $t = 1$ for binary cubes of dimension 2, 3, 4, and 5 (see Table I). □

We now determine the ratio r_∞ between the upper bound and the lower bound on $M^{(p)}_n(t)$ as $n \to \infty$. It follows from (19) that $r_\infty = 1$ if $t = 1$ and $n = 2^s$. We next examine the case $t = 2$. By applying (15) and Corollary 8, we get (for $n > 2, p > 4$, p is even)

$$\frac{2p^n}{V(2)+1} \le M^{(p)}_n(2) \le (M_{(\lceil n/2\rceil)}(1))^2$$

and

$$V(2) = 1 + 2n + \binom{n}{2}4 + \binom{n}{l}2 = 1 + 2n + 2n^2.$$

If $n/2 = 2^s$, then

$$M^{(p)}_{n/2}(1) \le \frac{p^{n/2}}{n/2}.$$

Therefore,

$$\frac{p^n}{1+2n+n^2} \le M^{(p)}_n(2) \le \frac{4p^n}{n^2}$$

which implies that for large n

$$\frac{p^n}{n^2} \lesssim M^{(p)}_n(2) \le \frac{4p^n}{n^2}.$$

The ratio r_∞ of the upper bound to the lower bound ($n/2 = 2^s$, $n \to \infty$) is equal to 4.

We next extend this analysis to $t > 2$. First we use the approximation

$$V(t) \sim \sum_{i=0}^{t} \binom{n}{i} 2^i \sim n^t 2^t/t!$$

for p-ary n-dimensional cubes if p and t are constant, and $n \to \infty$. Thus for $n/t = 2^s$ and constant t and p

$$\frac{p^n t!}{n^t 2^{t-1}} \lesssim M^{(p)}_n(t) \le \frac{p^n t^t}{n^t}. \tag{22}$$

Therefore, $r_\infty = t^t 2^{t-1}/t!$. For example, for $t = 2$, we have $r_\infty = 4$ as above, while for $t = 3, r_\infty = 18$.

To conclude this section, we note that its main results (Theorems 6–9) can be easily generalized to the case of mixed codes with codewords $(x_1, x_2, \cdots, x_n)$ where $x_i \in \{0, 1, \cdots, p_i - 1\}$. (For Theorems 6 and 7, p_i is even and $p_i > 4$ for all $i = 1, \cdots, n$.)

C. Other Topologies

The next topology that we consider is a balanced p-ary tree. A number of hierarchical computing systems such as dictionaries and search machines can be modeled as a tree [2], [24]. Many parallel algorithms can be mapped on to p-ary tree, and the architecture of a general-purpose multiprocessor can often be modeled by a tree structure [19]. Another application of a tree structure is the data network of the Thinking Machine CM-5 [11], [16].

We can uniquely identify vertices in a p-ary l-level tree with $t = 1$ by selecting as codewords vertices at levels $l, l-2, l-4, \cdots$, where the root is at level one and the leaf vertices are at level l. This yields the following bound on the number of codewords $M(1)$:

$$M(1) \le \frac{p^{l+1}}{p^2-1}\left(1 - p^{-2\lfloor 0.5(l-1)\rfloor+1}\right). \tag{23}$$

Theorem 10: For a p-ary tree with l levels ($l \ge 3$), we have the following bounds on the minimum number of codewords in the identifying code:

$$p^{l-3}(p^2+1) \le M(1) \le \begin{cases} \dfrac{p^{l+1}-1}{p^2-1}, & \text{if } l \text{ is odd} \\ \dfrac{p^{l+1}-p}{p^2-1}, & \text{if } l \text{ is even.} \end{cases}$$

Proof: The upper bounds follow from (23). The lower bound on $M(1)$ is obtained by viewing the p-ary l-level tree as containing p^{l-3} 3-level subtrees, each containing $1+p+p^2$ vertices, of which there are p^2 leaf vertices. We next show that at least p^2+1 vertices from each of these subtrees must be selected as codewords. First we note that at least $p(p-1)$ leaf vertices must be codewords (to cover the noncodeword leaf vertices), and in order to distinguish between the level-two vertices, the root of the subtree must be selected. A similar argument can be used for cases where p sibling vertices are selected as codewords. This yields a minimum of p^2+1 vertices in each subtree, and hence $M(1) \ge p^{l-3}(p^2+1)$. □

Corollary 10: For p-ary trees with $l = 3$ levels, $M(1) = p^2+1$, while for a p-ary tree with $l = 4$ levels, $M(1) = p(p^2+1)$.

The code construction of Theorem 10 is asymptotically optimal if $p \to \infty$ since $M(1) \sim p^{l-1}$ for large p, which coincides with the lower bound. For the binary tree ($p = 2$), we have

$$5 \cdot 2^{l-3} \le M(1) \le (16/3) \cdot 2^{l-3}$$

for large l, hence the codeword selection is very close to optimal. Table III lists the number of codewords for binary and ternary trees.

We next prove that the vertices in a tree are not identifiable if $t > 1$.

Theorem 11: It is not possible to uniquely identify the vertices of a p-ary l-level tree for $t > 1$.

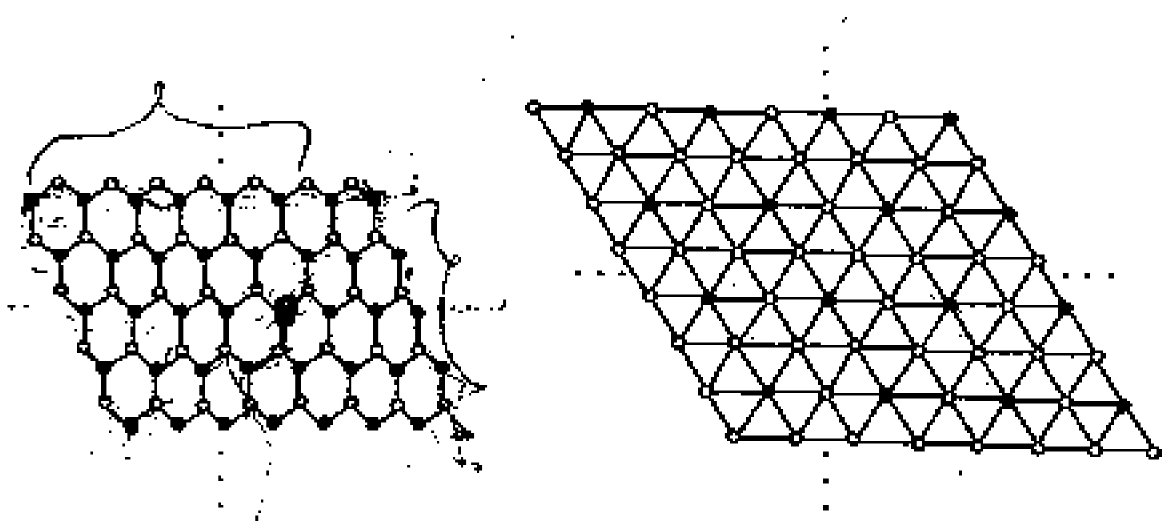
TABLE III
NUMBER OF CODEWORDS $m(1)$ FOR (a) BALANCED BINARY TREE ($P = 2$); (b) BALANCED TERNARY TREE ($p = 3$)

l	N	Lower bound, $\hat{m}(1)$, on $M(1)$	Upper bound $\hat{M}(1)$, on $M(1)$	$\hat{M}(1)/N$	$\hat{M}(1)/\hat{m}(1)$
3	7	5	5	0.714	1
4	15	10	10	0.666	1
5	31	20	21	0.677	1.05
8	255	160	170	0.666	1.0625
10	1023	640	682	0.666	1.0656
11	2047	1280	1365	0.666	1.0664
12	4095	2560	2730	0.666	1.0664
16	65535	40960	43690	0.666	1.0666

(a)

l	N	Lower bound, $\hat{m}(1)$, on $M(1)$	Upper bound $\hat{M}_1$, on $M(1)$	$\hat{M}(1)/N$	$\hat{M}(1)/\hat{m}_1$
3	13	10	10	0.769	1
4	40	30	30	0.75	1
5	121	90	91	0.752	1.0111
8	3280	2430	2460	0.75	1.0123
10	29524	21870	22143	0.75	1.0125
11	88573	65610	66430	0.75	1.0125
12	265720	196830	199290	0.75	1.0125
16	21523360	15943230	16142520	0.75	1.0125

(b)



(a) (b)

Fig. 3. Codewords (shaded) with $t = 1$ for a (a) hexagonal mesh and (b) triangular mesh (the ends wrap around).

Proof: Consider the subtree consisting of the sibling leaf vertices $V_l = \{v_1, v_2, \cdots, v_p\}$ and their parent v_{p+1}. For $t > 1$, v_i and v_j $(1 \leq i, j \leq p)$ cannot be distinguished by any selection of codewords. This is because the vertices in V_l are at distance two from each other and any vertex $v_j \notin V_l$ is at the same distance from all the vertices in V_l. Hence the vertices in V_l are not distinguishable if $t > 1$. □

Finally, we address the problem of code construction for hexagonal and triangular meshes, the former topology having received attention recently [23]. Every hexagonal (triangular) mesh has three (six) neighbors. Fig. 3 shows these topologies with the codewords (shaded) for vertex identification with $t = 1$.

For the hexagonal mesh, the number of codewords $M(1) = N/2$, where N is the total number of vertices in the graph. Every codeword is covered only by itself while every non-codeword is covered by a unique subset of three codewords. The lower bound on $M(1)$ for this topology obtained from (15) is $2N/5$.

The code construction for the triangular mesh is perfect since the number of codewords $M_1 = N/4$, which corresponds to the lower bound of (15). In this case, every codeword is covered only by itself while every noncodeword is covered by exactly two codewords. The above discussion is summarized by the following theorem.

Theorem 12: For a hexagonal mesh with N vertices ($N \to \infty$), the number of codewords $M(1)$ is given by

$$0.4N \lesssim M(1) \lesssim 0.5N,$$

while for a triangular mesh with N vertices,

$$M(1) \sim 0.25N.$$

III. IDENTIFYING SETS OF VERTICES

We have assumed thus far that only a single vertex in the graph G has to be uniquely identified. In this section, we show that codeword selection for single vertices provides a near-complete identification of sets of vertices of higher cardinality. Let $C(l)$ be the fraction of sets of vertices of cardinality exactly l that are uniquely identifiable.

Theorem 13: The fraction $C(l)$ of sets of vertices of cardinality exactly l that are uniquely identifiable with $t = 1$ by a code identifying single vertices (see Section II) is lower-bounded by

$$C(l) \geq \prod_{i=0}^{l-1} \frac{N - iV(4)}{N - i}$$

where $V(4)$ is the number of vertices at distance 4 or less from any given vertex in the graph, and N is the number of nodes in the graph G.

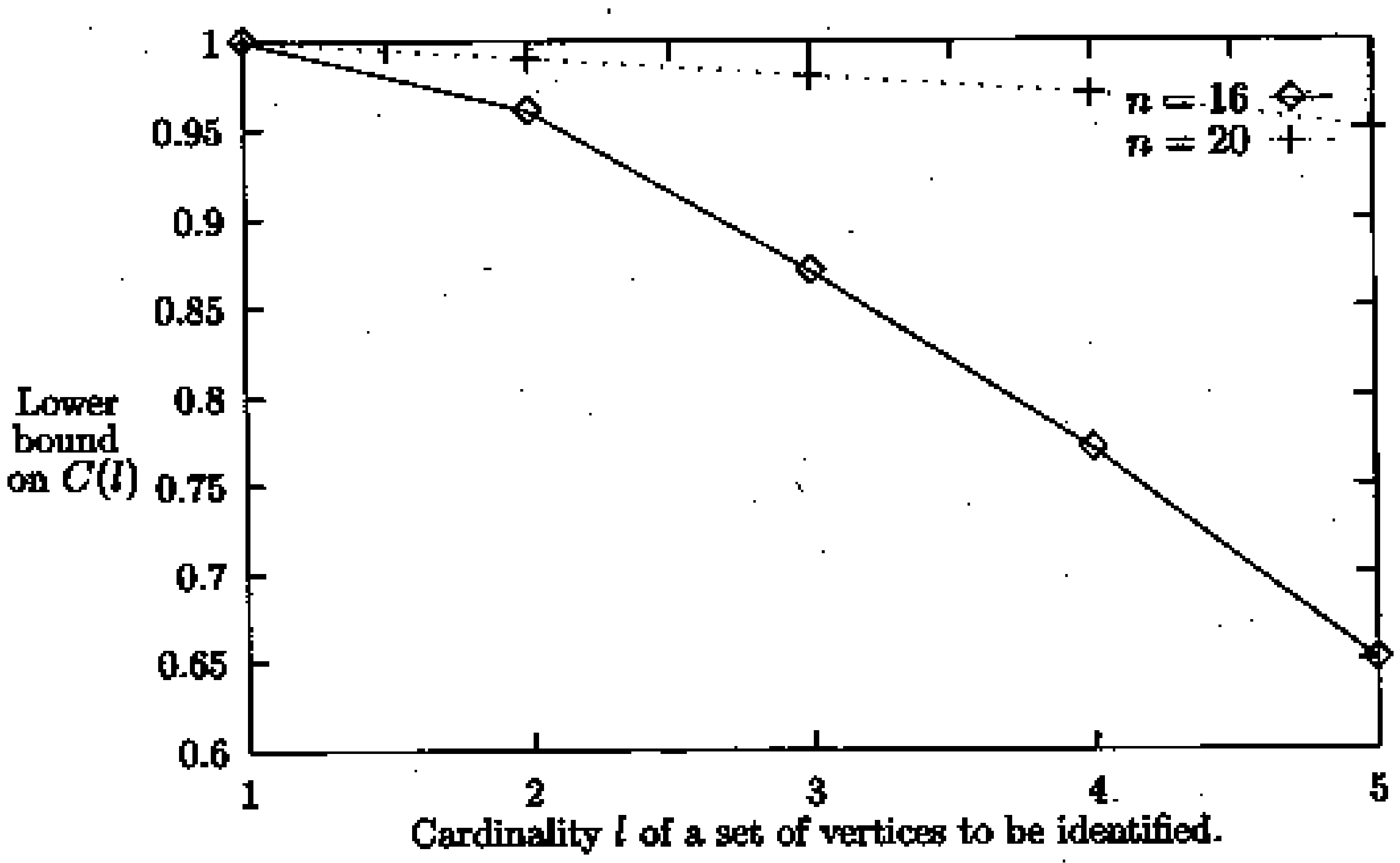
Proof: A set of vertices is uniquely identifiable if the distance between any two vertices in this set is at least five. Note that this condition is sufficient but not necessary. The fraction of identifiable sets of vertices is therefore lower-bounded by

$$C(l) \geq \frac{N(N - V(4))(N - 2V(4)) \cdots (N - (l-1)V(4))}{\binom{N}{l} l!}$$
$$= \prod_{i=0}^{l-1} \frac{N - iV(4)}{N - i}. \qquad \square$$

For example, $V(4) = 40$ for a p-ary two-dimensional cube ($p \geq 9$), and $V(4) = \Sigma_{i=1}^{4} \binom{n}{i}$ for an n-dimensional binary cube. It follows from the theorem that over 96% of sets of two vertices in a 16-dimensional binary cube are identifiable.

Fig. 4 shows the lower bound on the fraction of uniquely identifiable sets of vertices of higher cardinality in binary cubes.

Corollary 11: As the number of vertices in a graph with constant degree tends to infinity, the fraction of sets of vertices of cardinality exactly l that are uniquely identifiable approaches one if $l = o(\sqrt{N})$.



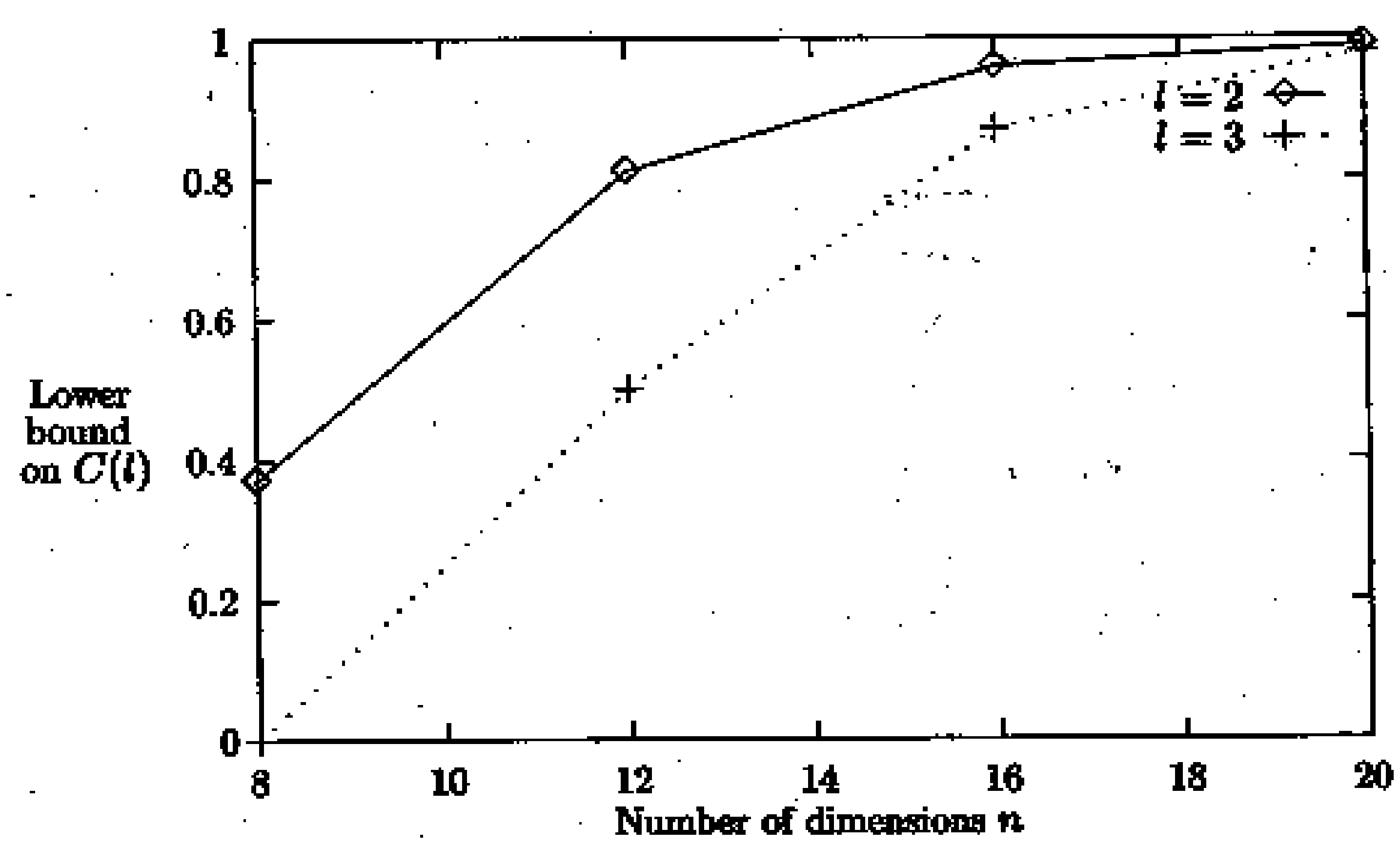


Fig. 4. Lower bound on the fraction of sets of vertices that are uniquely identifiable in binary cubes.

Proof: Let

$$\prod = \prod_{i=0}^{l-1} (N - iV(4)/N - i).$$

It can be easily seen that for $i \lesssim \sqrt{N}$

$$\ln \frac{N - iV(4)}{N - i} = \ln \left(1 - \frac{i(V(4) - 1)}{N - i}\right) \sim -\frac{i(V(4) - 1)}{N - i}$$

and

$$\prod \sim \sum_{i=1}^{l-1} -\frac{i(V(4) - 1)}{N - i}.$$

Now

$$\left|\sum_{i=1}^{l-1} \frac{i(V(4) - 1)}{N - i}\right| \leq \frac{(l - 1)(V(4) - 1)}{N - l + 1} (l - 1)$$

and

$$\lim_{N\to\infty} \frac{(l - 1)(V(4) - 1)}{N - l + 1}(l - 1) = 0$$

if $l^2/N \to 0$ (since $V(4)$ is constant). □

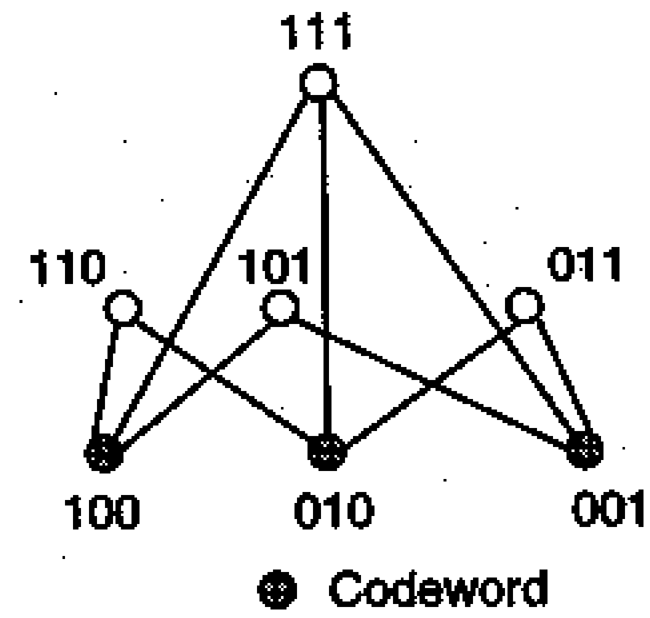


Fig. 5. An optimal graph for uniquely identifying a single vertex.

IV. Optimal Graphs

Finally, we develop a method for the construction of optimal graphs that require a minimal number of codewords for identifying sets of vertices. We are interested in generating a graph with N vertices in which the number of codewords is as close to $\lceil \log_2(N + 1) \rceil$ as possible for the identification of single vertices and to $\lceil \log_2 \sum_{i=0}^{l} \binom{N}{i} \rceil$ for identification of sets of up to l vertices.

We first consider identification of single vertices ($l = 1$). Consider a graph with $N = 2^n - 1$ vertices labeled

$$(1, 0, 0, \cdots, 0), (0, 1, 0, \cdots, 0), \cdots, (1, 1, \cdots, 1)$$

with vectors of length $n = \log_2(N + 1)$. We select all vectors of weight one as codewords. Consider any noncodeword $B = (b_1 b_2 \cdots b_n)$, where $b_j \in \{0, 1\}$. B is connected to codeword

$$(\underbrace{0, \cdots, 0}_{i-1}, 1, 0, \cdots, 0)$$

if and only if $b_i = 1$. (An example of this topology for $N = 7$ and $n = 3$ is given in Fig. 5.) This construction ensures that every vertex is covered by a unique set of codewords, hence identification of single vertices is achieved using a minimal code.

We next extend this construction to a general method for generating optimal graphs (and codes) for identifying sets of vertices.

Consider a matrix A with rows corresponding to codewords and columns corresponding to vertices in the graph. An entry $a_{i,j}$ in this matrix is one if codeword i covers vertex j. An optimal graph is constructed by generating A with a minimum number of rows. For identifying single vertices, A can be any matrix with different nonzero columns. If the logical OR of any $k (k \leq l)$ columns of A yields a unique nonzero vector, then sets of vertices of cardinality up to l are identifiable.

There are $\Sigma_{i=0}^{l} \binom{N}{i}$ sets of cardinality at most l. Hence a lower bound on the minimal number of rows $r(N, l)$ of A is given by

$$r(N, l) \geq \left\lceil \log_2 \sum_{i=0}^{l} \binom{N}{i} \right\rceil.$$

It is difficult to find the exact value of $r(N, l)$. However, near-optimal construction of the matrix A (and therefore the graph) for sets of vertices can be obtained using superimposed codes of length N [13] and techniques for conflict resolution in multiuser channels with N users [18]. For these codes, the