ON THE LOOP SWITCHING ADDRESSING PROBLEM

by

Andrew C. Yao

STAN-CS-77-626
OCTOBER 1977

COMPUTER SCIENCE DEPARTMENT
School of Humanities and Sciences
STANFORD UNIVERSITY

On the Loop Switching Addressing Problem

Andrew Chi-Chih Yao
Computer Science Department
Stanford University
Stanford, California 94305

Abstract.

The following graph addressing problem was studied by Graham and Pollak in devising a routing scheme for Pierce's Loop Switching Network. Let G be a graph with n vertices. It is desired to assign to each vertex v_i an address in $\{0,1,*\}^{\ell}$, such that the Hamming distance between the addresses of any two vertices agrees with their distance in G. Let N(G) be the minimum length ℓ for which an assignment is possible. It was shown by Graham and Pollak that $N(G) \leq m_G(n-1)$, where m_G is the diameter of G. In the present paper, we shall prove that $N(G) < 1.09(\lg m_G)n + 8n$ by an explicit construction. This shows in particular that any graph has an addressing scheme of length $O(n \log n)$.

Keywords: addressing scheme, binary tree, graph, Hamming distance, loop switching network.

This research was supported in part by National Science Foundation grant MCS 72-03752 A03.

1. Introduction.

An interesting routing scheme to Pierce's Loop Switching Network [7] was proposed by Graham and Pollak [3,4] (see also [1]). In this scheme, Pierce's network is represented by a graph where vertices stand for the loops, and edges stand for the contacts between loops in the network. The scheme calls for assigning a sequence of ternary symbols to each vertex such that the distances between vertices in the graph are faithfully represented. The combinatorial problem is described below; for a detailed discussion of the connection between Pierce's network and this combinatorial problem, as well as further information on the subject, see references [1,3,4,7].

Throughout our discussion, $G = (V,E)$ will be a connected graph with a set V of vertices, and a set E of undirected edges. A path of length t in G from a vertex v_i to a vertex v_j is a sequence of vertices $v_{k_0}, v_{k_1}, \ldots, v_{k_t}$ such that $v_{k_0} = v_i$, $v_{k_t} = v_j$, and $\{v_{k_{s-1}}, v_{k_s}\} \in E$ for $s = 1,2,\ldots,t$. The distance $d_G(v_i,v_j)$ between vertices v_i and v_j is the minimum length t for which a path of length t from v_i to v_j exists. The diameter of G, denoted by m_G, is the largest distance between any two vertices in G. That is,

$$m_G = \max\{d_G(v_i,v_j) \mid v_i, v_j \in V\} .$$

Let Σ be the ternary symbol set $\{0,1,*\}$. (The character "$*$" is a "don't-care" symbol.) The Hamming distance H between elements in Σ is defined by $H(1,0) = H(0,1) = 1$, and $H(a,b) = 0$ for all other pairs of a, b in Σ. For two sequences $\alpha = a_1 a_2 \ldots a_\ell$ and $\beta = b_1 b_2 \ldots b_\ell$ in Σ^ℓ, where $\ell > 0$, their Hamming distance is given by

$$H(\alpha,\beta) = \sum_{1 \le i \le \ell} H(a_i,b_i)_1 .$$

An addressing scheme for a graph $G = (V,E)$ with n vertices is an assignment of a sequence $c(v_i) \in \Sigma^{\ell}$ to each vertex v_i such that $H(c(v_i),c(v_j)) = d_G(v_i,v_j)$ for all v_i, v_j in V. The positive integer ℓ is called the length of the addressing scheme, and the sequence $c(v_i)$ the address of vertex v_i. It is desired to find addressing schemes with small length. Let $N(G)$ be the minimum ℓ for which an addressing scheme of length ℓ exists for G. In [3], it was proved that an addressing scheme always exists (i.e., $N(G) < \infty$), and furthermore, $N(G) \leq m_G(n-1)$. We shall improve this bound by explicitly constructing an addressing scheme. The main results are as follows: (We shall use λ to denote the constant $\left(\frac{1}{3} \lg 3 + \frac{2}{3} \lg \frac{3}{2}\right)^{-1} \approx 1.09$.)

Theorem 1. For a graph G with n vertices,

$$N(G) \leq \lambda\, n \lg n + 2n .$$

Theorem 2. For a graph G with n vertices,

$$N(G) \leq \lambda\, n(\lg m_G) + 8n .$$

Note : lg means logarithm to the base 2.

2. Definitions and Preliminaries.

Let G = (V,E) be a (connected, undirected) graph. A path $v_{k_0}, v_{k_1}, \ldots, v_{k_t}$ in G is simple if all the vertices v_{k_s} for $0 \leq s \leq t$ are distinct, except possibly for $v_{k_0} = v_{k_t}$. A graph G' = (V',E') is called a subgraph of G if $V' \subseteq V$ and $E' \subseteq E$. A subgraph G' = (V',E') is said to be a tree in G if G' is connected and there is no simple path of length > 0 in G' from any vertex $v \in V'$ to itself. A tree G' = (V', E') in G is a spanning tree for G if V' = V. For any subset of vertices $V' \subset V$, the diameter of V', written $\mathrm{diam}_G(V')$, is $\max\{d_G(v_i,v_j) \mid v_i,v_j \in V'\}$. In particular, $\mathrm{diam}_G(V) = m_G$. The distance $d_G(v_i, V')$ between a vertex $v_i \in V$ and a subset $V' \subset V$ is defined as $d_G(v_i,V') = \min\{d_G(v_i,v_j) \mid v_j \in V'\}$.

We shall make use of binary trees in our design process. (See for example Knuth [5] for basic definitions regarding binary trees.) Let T be a binary tree with n leaves. Assume the nodes of T are numbered arbitrarily from 1 to 2n-1. The node with number k will be denoted by r_k. We will also use the notation $\boxed{i}$ for a leaf numbered i, and $\textcircled{j}$ for an internal node numbered j. For a node r_k, let R(k) be the subset of leaves in T which are descendants of r_k. The size of R(k) is called the weight of r_k, denoted by w(k). For example, we have $R(1) = \{r_8, r_6, r_9\}$, $R(2) = \{r_2\}$, and W(1) = 3, w(2) = 1 in Figure 1. The external path length P(T) is defined by the following equation

$$P(T) = \sum_{\text{internal node } r_k} w(k) \quad . \tag{1}$$

The quantity P(T) can alternatively be described as the sum of the distances from the leaves to the root [5]. If r_i and r_j are respectively the leftson and the rightson of r_k, we shall write i = leftson(k),

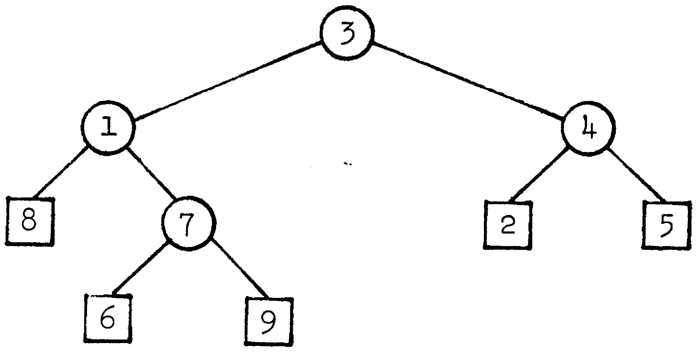


Figure1

j = rightson(k)); k = father(i) = father(j) ; and j = brother(i) , i = brother(j) . As a shorthand, we shall use $\bar{k}$ for father(k) and k' for brother(k) . A binary tree T is said to be <u>weight-balanced</u> if for each internal node r_k ,

$$\frac{1}{3} w(k) \leq w(\text{leftson}(k)) \leq \frac{2}{3} w(k) \ ,$$
$$\frac{1}{3} w(k) \leq w(\text{rightson}(k)) \leq \frac{2}{3} w(k) \ . \qquad (2)$$

The following result is from [6, Theorem 2].

<u>Lemma 1</u> [Nievergelt and Wong]. If T is a weight-balanced binary tree with n leaves, then the external path length of T satisfies

$$P(T) \leq \lambda \, n \lg n \approx 1.09 \, n \lg n \ .$$

In a binary tree T , if a leaf $c^i{}_1$ precedes another leaf $c^j{}_1$ in post-order [5], we shall say that $\underline{\boxed{i} \text{ is to the left of}_a \ j}$ (or $\boxed{j}$ is to the right of $\boxed{}$), and write $\boxed{i} \lessdot \boxed{}$ (or equivalently $\boxed{j} \gtrdot \boxed{i}$). We further extend the relation so that

$$\boxed{i} \lessdot r_k \quad \text{if} \quad \boxed{i} \lessdot \boxed{j} \text{ for all descendants } \boxed{j} \text{ of } r_k ,$$

$$\boxed{i}\ 3 - r_k \quad \text{if} \quad \boxed{i} \gtrdot \boxed{j} \text{ for all descendants } \boxed{j} \text{ of } r_k .$$

Clearly, for any leaf $\boxed{i}$ and node r_k; either $\boxed{i} \lessdot r_k$, $\overset{i}{\text{c}}\text{l} \gtrdot r_k$, or $\boxed{i}$ is a descendant of r_k ; and exactly one of the three relations holds. In Figure 1, we have $\boxed{6} \lessdot \boxed{2}$, $\boxed{6} \lessdot \textcircled{4}$, and $\boxed{2} \gtrdot \textcircled{7}$.

3. The Construction of a Length $O(n \lg n)$ Addressing Scheme.

3.1 The Design Tree.

The key to obtaining an $O(n \lg n)$ scheme is by using a hierarchical design. A design tree M is a pair (T,f) where T is a binary tree with n leaves, and f is a one-to-one mapping from the leaves of T to the vertices of G , For notational convenience, we shall number the nodes of T in such a way that the leaves receive numbers 1 to n and leaf i is associated with vertex v_i under f . The root of T will be labeled with $2n-1$; and the remaining internal nodes with $n+1$ through $2n-2$ (their actual numbering will be unimportant for M).

We now describe an addressing scheme Z(M) corresponding to a given design tree M . Every address $c(v_i)$ in Z(M) will consist of $2n-2$ blocks of code, where the k-th block has length ℓ_k (to be defined later) and is conceptually associated with the node r_k of T . (Note that rk cannot be the root since $k \neq 2n-1$.) Thus we shall write, for $1 \leq i \leq n$,

$$c(v_i) = c_{i1}\, c_{i2} \cdots c_{i,2n-2} \qquad \text{where} \quad c_{ik} \models \ell_k \quad . \tag{3}$$

By definition, the Hamming distance between two addresses $c(v_i)$ and $c(v_j)$ is equal to the sum of the Hamming distances between corresponding blocks. That is,

$$H(c(v_i),c(v_j)) = \sum_{k=1}^{2n-2} H(c_{ik},c_{jk}) \quad . \tag{4}$$

We shall design the code in such a way that in (4), only a few terms will contribute to the sum, other terms being zero. For example, consider the design tree M shown in Figure 2. We shall in fact have

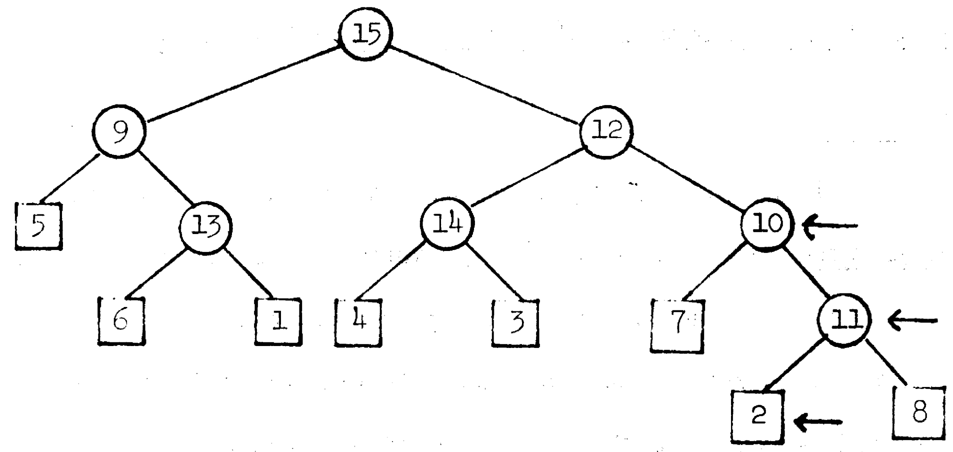


Figure 2. A design tree M with a marked path.

$$H(c(v_3),c(v_2)) = H(c_{3,10},c_{2,10}) + H(c_{3,11},c_{2,11}) + H(c_{3,2},c_{2,2}) \qquad (5)$$

and $H(c_{3,k},c_{2,k}) = 0$ for $k \notin \{10,11,2\}$. The trick to achieve $H(c(v_3),c(v_2)) = d_G(v_3,v_2)$ is as follows. Define $S(k) = \{f(r_i) \mid r_i \in R(k)\}$ i.e., $S(k)$ is the set of vertices associated with the leaf descendants of r_k . We shall require that,

$$H(c_{3,10},c_{2,10}) = d_G(v_3,S(10)) ,$$

$$H(c_{3,10},c_{2,10}) + H(c_{3,11},c_{2,11}) = d_G(v_3,S(11)) ,$$

$$H(c_{3,10},c_{2,10}) + H(c_{3,11},c_{2,11}) + H(c_{3,2},c_{2,2}) = d_G(v_3,S(2)) . \qquad (6)$$

We can view (6) in the following way. Starting at the <u>lowest common ancestor</u> (lca) of $\boxed{3}$ and $\boxed{2}$ (i.e., the common ancestor of $\boxed{3}$ and $\boxed{2}$ farthest from the root), which is r_{12} , we move down the path r_{10} , r_{11} , to the leaf r_2 . Each node r_k encountered along the path, excluding the lca, will add a block of code which creates enough Hamming distance to bring the total up to $d(v_3, S(k))$. An equivalent form of (6) is

$$H(c_{3,k}, c_{2,k}) = d_G(v_3, S(k)) - d_G(v_3, S(\bar{k})) \tag{7}$$

for $k = 10, 11, 2$, and $\bar{k}$ = father(k) .

In general, we want to achieve the following. For $\boxed{i} \lessdot \boxed{j}$, let node h_0 be the lowest common ancestor $\boxed{i}$ i and c^{j}_{1}, and $h_0, h_1, \ldots, h_t = j$ be the path from node h_0 to c^{j}_{1}, then

$$H(c_{i,k}, c_{j,k}) = d(v_i, S(k)) - d(v_i, S(\bar{k}))$$

for $k = h_1, h_2, \ldots, h_t$, and $\bar{k}$ = father(k) ;

$$H(c_{i,k}, c_{j,k}) = 0 \quad \text{for all other } k \text{ .} \tag{8}$$

It is easy to verify that (8), if true for all $\boxed{i} \lessdot \boxed{j}$, will be sufficient to guarantee that $Z(M) = \{c(v_i) \mid 1 \leq i \leq n)$, as given by (3), is an addressing scheme, That is, $d_G(v_i, v_j) = H(c(v_i), c(v_j))$ for all i , j . We now describe a construction of the c_{ik}'s that satisfy (8).

<u>Z(M): The Addressing Scheme Induced by M</u> . For each k, $1 \leq k \leq 2n-2$, let

$$\ell_k = \max_{1 \leq i \leq n} [d_G(v_i, S(k)) - d_G(v_i, S(\bar{k}))] \text{ .} \tag{9}$$

. The block c_{ik} , for $1 \leq i \leq n$, has length ℓ_k and is given by

$$c_{ik} = \begin{cases} 000 - - - \ldots 0 & \text{if } c^{i}_{1} \text{ is a descendant of } r_k \text{ ,} \\ *** - - - \ldots * & \text{if } a^{i} \gtrdot r_k \text{ ,} \\ \underbrace{111 - 1}_{\delta}*** \ldots * & \text{with } \delta = d_G(v_i, S(k)) - d_G(v_i, S(\bar{k})) \text{ ,} \\ & \text{if } c^{i}_{1} \lessdot r_k \text{ .} \end{cases} \tag{10}$$

Finally, form $Z(M) = \{c(v_i) \mid 1 \le i \le n\}$ according to (3). The length of $Z(M)$, denoted by $\tau(M)$, is

$$-c(M) = \sum_{1 \le k \le 2n-2} \ell_k \quad . \tag{11}$$

To see that $Z(M)$ is actually an addressing scheme, we need only show that (8) is satisfied, For $\boxed{\ } < \boxed{j}$, we see from (10) that $H(c_{ik}, c_{jk}) = 0$ unless $\boxed{i} < r_k$ and $\boxed{j}$ is a descendant of r_k ; in the latter case, $H(c_{ik}, c_{jk}) = d_G(v_i, S(k)) - d_G(v_i, S(\bar{k}))$. But this is exactly as required by (8), q.e.d.

3.2 Criteria for a Good Design Tree.

Let us find out what sort of design tree M will generate a short addressing scheme, Notice that for any $1 \le i < n$, $1 \le k \le 2n-2$, we have

$$d_G(v_i, S(k)) - d_G(v_i, S(\bar{k})) \le \mathrm{diam}_G(S(\bar{k})) \quad . \tag{12}$$

Inequality (12) is valid, since we can concatenate a path from v_i to the nearest point in $S(\bar{k})$, with a path of length at most $\mathrm{diam}_G(S(\bar{k}))$, to reach a vertex in $S(k)$. This tells us that

$$\ell_k \le \mathrm{diam}_G(S(\bar{k})) \quad . \tag{13}$$

An upper bound to $\tau(M)$ is therefore

$$\tau(M) \le \sum_{1 \le k \le 2n-2} \mathrm{diam}_G(S(\bar{k})) = 2 \sum_{n+1 \le k \le 2n-1} \mathrm{diam}_G(S(k)) \; , \tag{14}$$

every internal node being the father of two nodes. This upper bound will in general be $O(n^2)$, as the subset $S(k)$ may have diameter $O(n)$ for many k. However, if we insist on two conditions

(i) no two points in $S(k)$ are far apart compared to its size $|S(k)|$, specifically, $\text{diam}_G(S(k)) \leq |S(k)|$; and

(ii) the binary tree is weight-balanced,

then (14) would give

$$\tau(M) \leq 2 \sum_{n+1 \leq k \leq 2n-1} |S(k)| = 2 \cdot P(T) \leq 2\lambda n \lg n \qquad (15)$$

by Lemma 1.

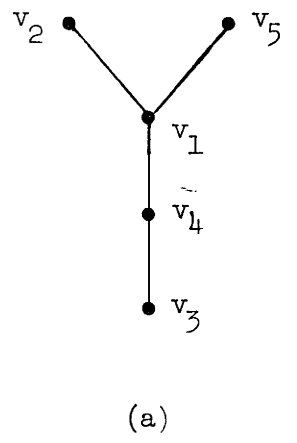
To achieve conditions (i) and (ii), we use the following idea. Let us think of $M = (T,f)$ as a tree built topdown by successively breaking V into smaller parts. From this viewpoint, the tree in Figure 2 is obtained by first dividing (at node 15) $\{v_1, v_2, \ldots, v_8\}$ into $\{v_5, v_6, v_1\}$ and $\{v_4, v_3, v_7, v_2, v_8\}$; each of the two resulting parts are further divided into $\{v_5\}$, $\{v_6, v_1\}$ at node 9, and $\{v_4, v_3\}$, $\{v_7, v_2, v_8\}$ at node 12, respectively. This process is repeated until we have only the singleton sets $\{v_i\}$.

We shall see that in building M in this fashion, it is possible to keep the points in each part close together (condition (i)), and also make the two parts more or less equal in size (condition (ii)) on each decomposition, We shall describe such a method next, and then perform a finer analysis improving the bound given by (15).

3.3 Constructing M from a Spanning Tree.

We shall construct a design tree M with the properties (i) and (ii) given in Section 3.2. Choose any spanning tree with edge set A for the graph G, Let us create a new vertex v_0 and a new edge $\{v_0, v_1\}$.

We now define a one-to-one mapping φ between the edge set of the augmented spanning tree $A' = A \cup \{\{v_0, v_1\}\}$ and the vertex set V (without v_0). The mapping φ is obtained by regarding $(V \cup \{v_0\}, A')$ as a rooted tree with root v_0, and mapping each edge onto its "lower" end point. We shall then number the edges e_i in A' so that $\varphi(e_i) = v_i$. An example of this process is shown in Figure 3.



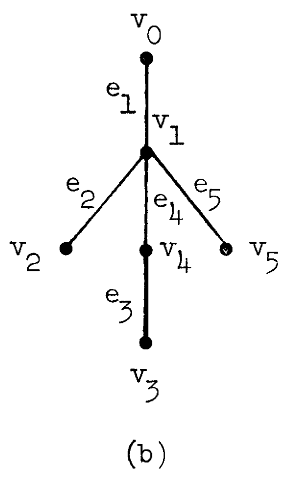
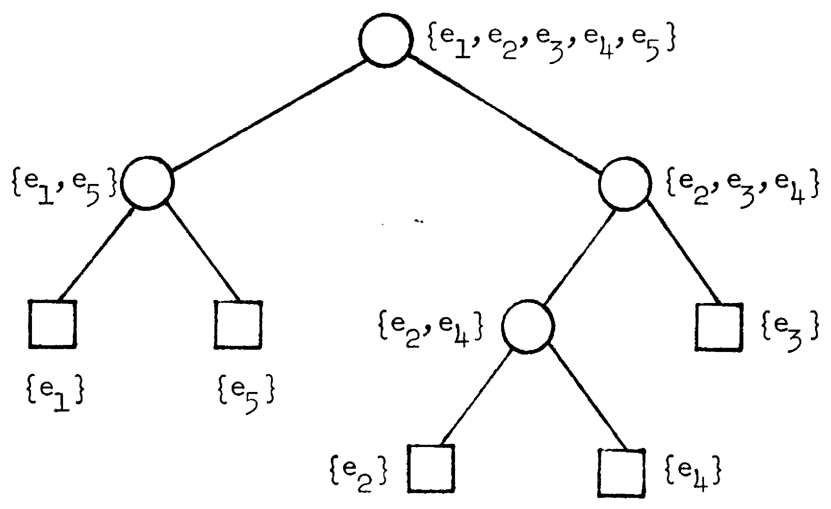


Figure 3. (a) A spanning tree on $V = (v_1, v_2, v_3, v_4, v_5\}$, and (b) the labelling of its edges after augmentation.

Our plan is to construct a binary tree Q by "suitably" splitting the edge set A' into two disjoint subsets, and repeat the process until only one edge remains in each subset. Figure 4(a) shows the binary tree Q that may result from this process when applied to the spanning tree in Figure 3(b). Although the tree Q so constructed is not a design tree on the vertex set, we can easily obtain such a design tree M_Q from

(a)



(b)

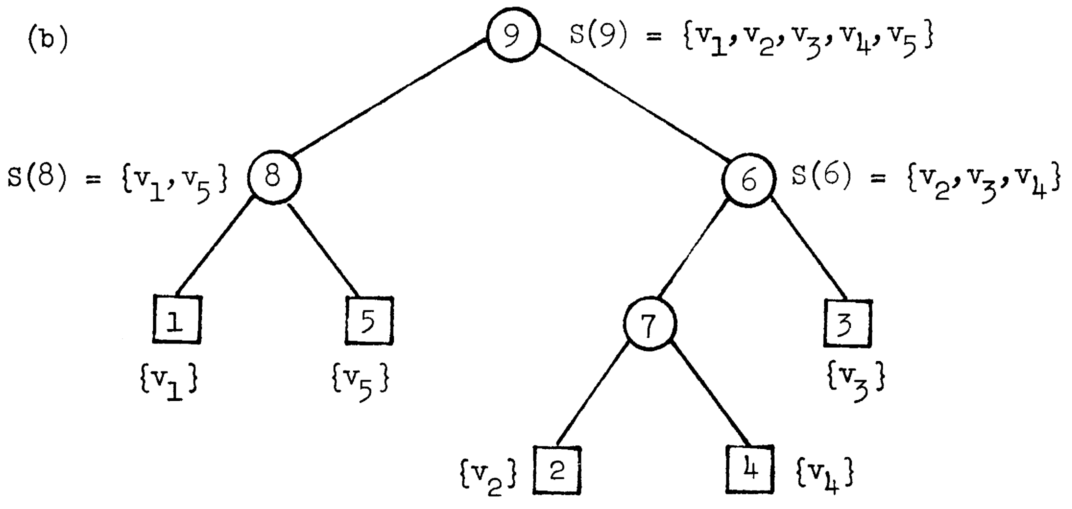


Figure 4. (a) A binary tree Q , and (b) its associated M_Q .

Q in a natural way via the mapping φ , We shall transform Q into M_Q simply by identifying e_i with v_i in the tree Q . The design tree M_Q obtained from the Q in Figure 4(a) is shown in Figure 4(b).

We can now complete our task in two steps, (1) describe the topdown construction of a Q for which M_Q would satisfy conditions (i) and (ii), and (2) analyze the addressing scheme induced by such an M_Q .

(1) Constructing Q . A set of edges B in G is called a tree set if B is the edge set of some tree in G . Two tree sets B_1 and B_2 is said to--form a decomposition of the tree set B if $B_1 \cap B_2 = \emptyset$ and $B_1 \cup B_2 = B$. Note that, in such a decomposition, there is a unique vertex v_s that is incident to both B_1 and B_2 , For example, in Figure 3(b), $B = \{e_2, e_4, e_5\}$ is a tree set. We can decompose B into $\{e_2\}$ and $\{e_4, e_5\}$ with v_1 being the unique vertex v_s .

A decomposition of B into B_1 and B_2 is balanced if $\frac{1}{3} |B| \leq B_i \leq \frac{2}{3} |B|$ for i = 1,2 . The following lemma is implicit in [2].

Lemma 2 [Chung and Graham]. Any tree set B with $|B| \geq 2$ has a balanced decomposition into two tree sets.

Let us now construct Q by breaking the augmented spanning tree A' into parts successively, using a balanced decomposition at each step. For example, the tree Q shown in Figure 4(a) can be obtained this way from A' in Figure 3(b). Once Q is constructed, we transform it into a design tree M_Q for the vertex set as described previously. It remains to analyze the address length obtained from this tree M_Q . To avoid confusion, we use $S(k)$ for the set of vertices associated with node r_k in M_Q , and use $B(k)$ to denote the tree set at the corresponding node in Q . Clearly, if $S(k) = \{v_{i_1}, v_{i_2}, \ldots, v_{i_t}\}$, then $B(k) = \{e_{i_1}, e_{i_2}, \cdot$ [illegible] $_{i_t}$ [illegible]

(2) Analysis. There are two simple properties of the design tree M_Q . Firstly, M_Q is weight-balanced by construction, Secondly, at any node r_k of M_Q , $\text{diam}_G(S(k)) \leq |S(k)|$. This is true since any two vertices in $S(k)$ can be connected through at most $|S(k)|$ edges in the tree set $B(k)$. Thus, the two conditions (i) and (ii) in Section 3.2 are satisfied, which implies $\tau(M) \leq 2\lambda n \lg n$. A stronger bound can be obtained, however, by using the following lemma.

Lemma 3. For each node r_k in M_Q , and $1 \leq i \leq n$,

$$d_G(v_i, S(k)) - d_G(v_i, S(\bar{k})) \leq 1 + |S(k')| , \text{ where } k' = \text{brother}(k) . \quad (16)$$

Proof. Let v_j be a vertex in $S(k)$ closest to v_i , i.e.,

$$d_G(v_i, v_j) = d_G(v_i, S(\bar{k})) . \quad (17)$$

If $v_j \in S(k)$, then $d_G(v_i, S(k)) = d_G(v_i, S(\bar{k}))$, and (16) is true. So we can assume that $v_j \in S(k')$.

Let v_s be the unique vertex that is incident to both an edge in $B(k')$ and an edge in $B(k)$. This implies that

$$d_G(v_j, v_s) \leq |B(k')| = |S(k')| . \quad (18)$$

Now, let $\{v_s, v_t\}$ be an edge in $B(k)$ incident with v_s (see Figure 5).

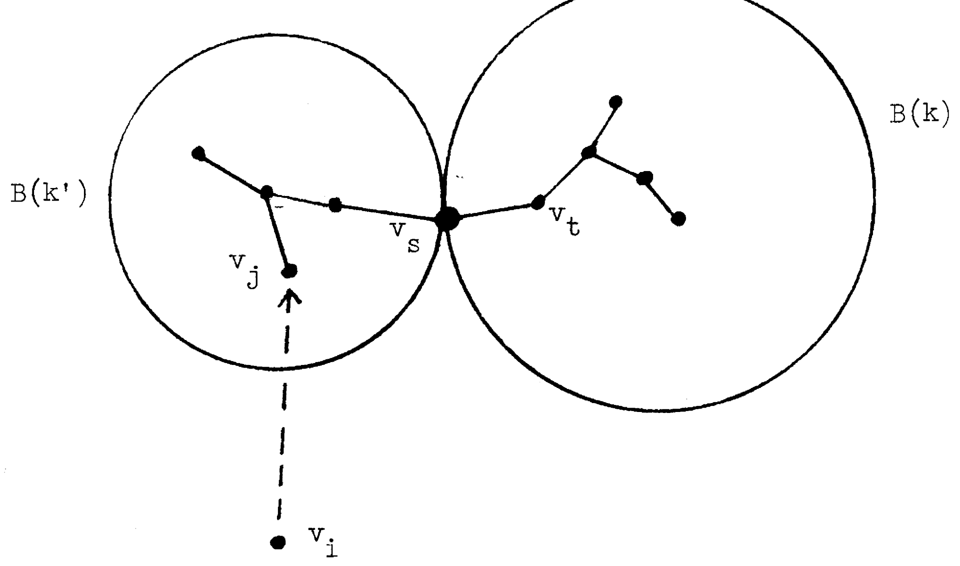


Figure 5

Then,

$$d_G(v_i, v_s) \le d_G(v_i,v_j) + d_G(v_j,v_s) \le d_G(v_i,S(\bar{k})) + |S(k')| \quad , \qquad (19)$$

$$d_G(v_i, v_t) \le d_G(v_i,v_s) + 1 \qquad < d_G(v_i,S(\bar{k})) + 1 + |S(k')| \quad . \qquad (20)$$

Since $\{v_s, v_t\} \in B(k)$, either v_s or v_t must be in $S(k)$. Therefore,

$$d_G(v_i, \text{[illegible]} \le \max\{d_G(v_i,v_s), d_G(v_i,v_t)\} \quad . \qquad (21)$$

Formula (16) follows from (19), (20), and (21). □

Lemma 3 implies that,

$$\ell_k = \max_i \{d_G(v_i,S(k)) - d_G(v_i,S(\bar{k}))\} < 1 + |S(k')| \quad .$$

Therefore

$$\tau(M_Q) = \sum_{1 \leq k \leq 2n-2} \ell_k \leq \sum_{1 \leq k \leq 2n-2} (1 + |S(k')|)$$

$$= 2n - 2 + \sum_{1 \leq k \leq 2n-2} |S(k)| \quad . \qquad (22)$$

Making use of the fact that M_Q is weight-balanced and Lemma 1, we obtain after simplification,

$$\tau(M) \leq \lambda n \log n + 2n \ .$$

This proves Theorem 1.

3.4 Proof of Theorem 2.

When m_G, the diameter of G, is substantially smaller than $n-1$, the addressing scheme we have constructed is better than the bound in Theorem 1 indicates. The key observation is that ℓ_k is always no greater than m_G, because $\ell_k \leq \max_i d_G(v_i, S(k)) \leq m_G$. In the analysis of $\tau(M_Q) = \sum \ell_k$, we can thus use m_G to bound ℓ_k, instead of $1 + |S(k')|$, for some of the nodes r_k.

Let X be the set of nodes r_k in M_Q such that $|S(k)| \leq m_G$, and $|S(\bar{k})| > m_G$. For each $r_k \in X$, let $J_k = \{r_j \mid r_j$ is a descendant of $r_k, r_j \neq r_k\}$. Let $J = \bigcup_{r_k \in X} J_k$. In Figure 6, assume $m_G = 4$, the set X then consists of the nodes marked by arrows, and J is the set of shaded nodes. We shall use inequality $\ell_k < 1 + |S(k')|$ for the nodes $r_k \in J$, and use $\ell_k \leq m_G$ for the remaining nodes in deriving a bound for $\tau(M_Q)$.

The following facts will be used in the calculation.

Fact 1. Let q be the number of nodes not in J, then $q < \frac{6n}{m_G}$,

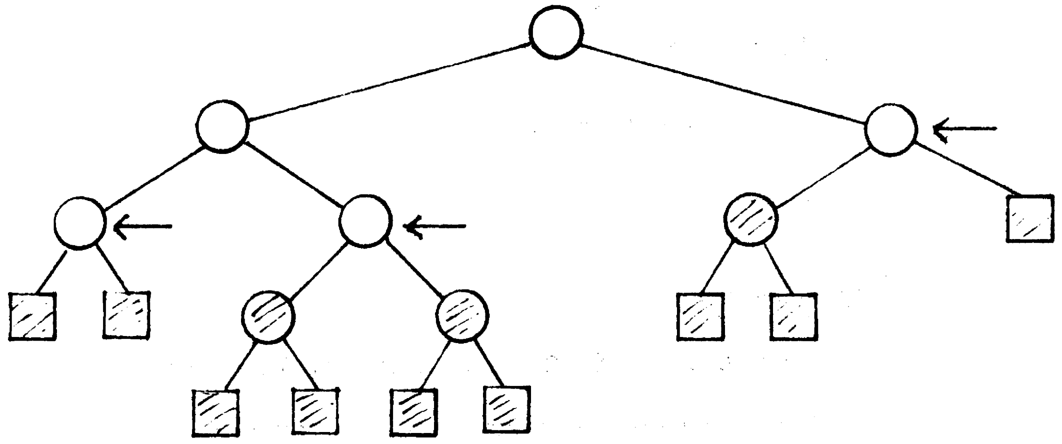


Figure 6. The set of shaded nodes is J .

Proof. $|J| = \sum_{r_k \in X} (2 \cdot |S(k)| - 2) = 2\left(\sum_{r_k \in X} |S(k)|\right) - 2|X| = 2n - 2|X|$.

Hence $q = 2n - 1 - |J| = 2|X| - 1$. Since $|S(k)| \geq \frac{1}{3} |S(\bar{k})| \geq \frac{1}{3} m_G$ for $r_k \in X$, we have $|X| \leq \frac{n}{\frac{1}{3} m_G}$. Thus, $q < 2|X| \leq \frac{6n}{m_G}$. ☐

Fact 2. Let $r_k \in X$, then $\sum_{r_j \in J_k} |S(j')| \leq \lambda |S(k)| \lg |S(k)|$.

Proof. $\sum_{r_j \in J_k} |S(j')| = \sum_{r_j \in J_k} |S(j)|$. Fact 2 then follows from the fact that the subtree of M_Q rooted at r_k is weight-balanced. ☐

We can now prove the desired bound as follows:

$$\tau(M_Q) = \sum_{r_k \notin J} \ell_k + \sum_{r_k \in J} \ell_k \leq \sum_{r_k \notin J} m_G + \sum_{r_k \in J} (1 + |S(k')|) .$$

$$= q\, m_G + |J| + \sum_{r_k \in X} \sum_{r_j \in J_k} |S(j')|$$

$$\leq \frac{6n}{m_G} \cdot m_G + 2n + \lambda \sum_{r_k \in X} |S(k)| \lg |S(k)| \qquad (24)$$

where we have used Facts 1 and 2 in the last step.

Equation (24) leads to, by using $|S(k)| \leq m_G$,

$$\tau(M_Q) \leq 8n + \lambda(\lg m_G) \sum_{r_k \in X} |S(k)|$$

$$= 8n + \lambda(\lg m_G)n .$$

This completes the proof of Theorem 2. ☐

4. Remarks.

In this paper we have given an algorithm which, for a graph with n vertices, constructs an addressing scheme of length $O(n \log n)$. The algorithm can be implemented straightforwardly, and has a $O(n^3)$ running time on a random access machine.

Some slight improvements on our bounds can be obtained by minor modifications of the construction. For example, the 8n term in Theorem 2 can be lowered to 4n , However, we have not found a construction that is guaranteed to give an address of length less than $O(n \log n)$. The very attractive conjecture $N(G) \leq n-1$ of Graham and Pollak [3,4] thus still remains an open problem.

Acknowledgements. I wish to thank Ronald L. Graham for introducing this problem to me in a stimulating conversation on this subject.

References

[1] L. H. Brandenburg, B. Gopinath, and R. P. Kurshan, "On the addressing problem of loop switching," Bell System Technical Journal 51 (1972), 1445-1469.

[2] F. R. K. Chung and R. L. Graham, "On graphs which contain all small trees," to appear in Journal of Combinatorial Theory (B).

[3] R. L. Graham and H. O. Pollak, "On the addressing problem for loop switching," Bell System Technical Journal 50 (1971), 2495-2519.

[4] __________, "On embedding graphs in squashed cubes," in Graph Theory and Applications, Lecture notes in Mathematics, number 303, Springer-Verlag (Proc. of a conference held at Western Michigan University, May 10-13, 1972).

[5] D. E. Knuth, The Art of Computer Programming, Vol. 1, Fundamental Algorithms; Second Edition, Addison-Wesley, Reading, Mass., 1975.

[6] J. Nievergelt and C. K. Wong, "Upper bounds for the total path length of binary trees," Journal ACM 20 (1973), 1-6.

[7] J. R. Pierce, "Network for block switching of data," Bell System Technical Journal 51 (1972), 1133-1145.