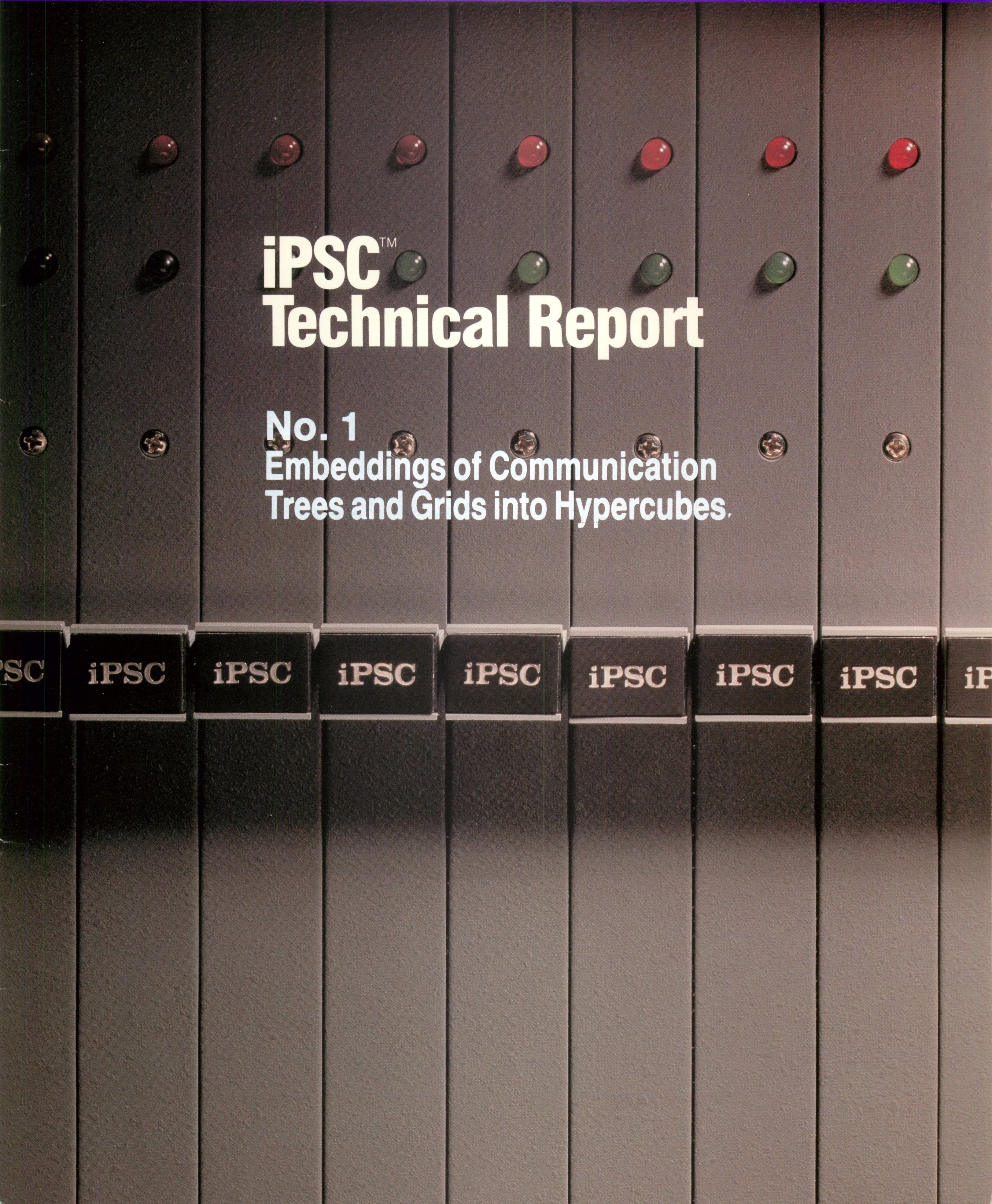
intel

iPSC™ Technical Report

No. 1
Embeddings of Communication Trees and Grids into Hypercubes



Embeddings of Communication Trees and Grids into Hypercubes

Joseph E. Brandenburg and
David S. Scott, both of
Intel Scientific Computers

ABSTRACT

This paper presents proofs of the following three results:

1. A full binary tree with $2^n - 1$ nodes can be embedded in a hypercube of dimension $d = n + 1$.

2. If G is an $l_1 \times l_2 \times \cdots \times l_k$ rectangular mesh, then the smallest hypercube which contains G as a subgraph has dimension $d = m_1 + m_2 + \ldots + m_k$ where $m_i = \lceil \log_2 l_i \rceil$ for all i.

3. If a global broadcast is implemented using a maximally embedded tree, then no node but the root (the source of the broadcast) needs to know the node number of the root. Other nodes only need to know the node from which they received the message.

1. INTRODUCTION

Commercial hypercube structured parallel computers have recently become available. This has sparked interest in the question of whether particular graphs can be embedded in hypercubes preserving nearest neighbor connections.

In this paper we prove minimality results for embedding full binary trees and rectangular meshes in hypercubes. We also prove an independence result that simplifies implementing a global broadcast in the hypercube.

2. EMBEDDING FULL BINARY TREES

If T_n is a full binary tree with n levels, then T_n has 2^n-1 nodes. Since an n dimensional hypercube (C_n) has 2^n nodes, this raises the question of whether T_n can be embedded in this cube. For $n > 2$, the answer is no, as can be seen by examination of the $n = 3$ cube [1].

Theorem 2.1: T_n can be embedded in the $n+1$ cube for all n.

To prove the first theorem, we prove that an even more complicated structure is contained in the cube. We first define U_k for $k \geq 1$.

As a graph, $U_k = (V, E)$ where

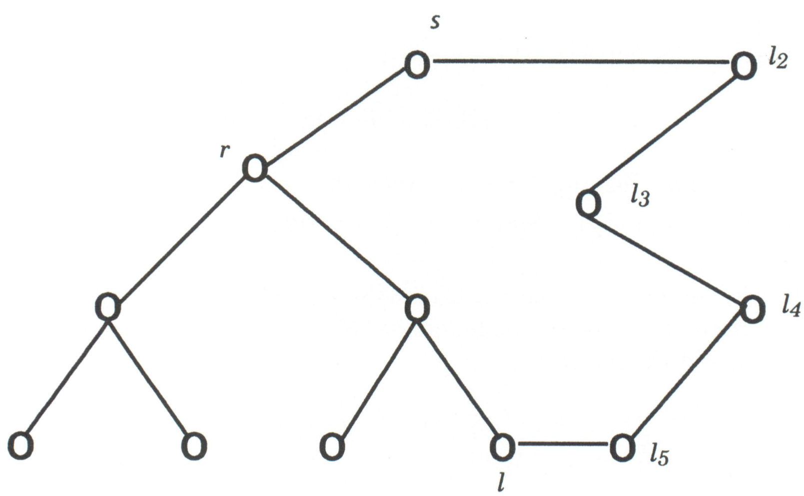
$$V = V' \bigcup \{s\} \bigcup \{l_2, \ldots, l_{k+2}\}$$

(for convenience, set $s = l_1$)

$$E = E' \bigcup \{(r, s)\} \bigcup_{j=1}^{k+1} \{(l_j, l_{j+1})\} \bigcup \{(l_{k+2}, l)\}$$

and $T_k = (V', E')$ is a k-level full binary tree with root r and a leaf l.

Graphically, U_3 is represented by



Lemma 2.2: For every $n \geq 1$, U_n embeds into C_{n+1} (the $n+1$ dim hypercube).

Proof of the Lemma:

Let $n = 1$. Since $T_1 = (\{r\}, \phi)$, r is both the root and a leaf. Thus, U_1 is a cycle of 4 nodes and graph isomorphic to C_2.

Now suppose U_{k-1} embeds into C_k, $k > 1$, for every k-cube C_k.

Given C_{k+1}, choose a binary labeling of the nodes of C_{k+1} such that nearest neighbors differ in only one bit. Let $C_k^{(i)}$ be the subcube of C_{k+1} of all nodes whose leading bit is i ($i = 0, 1$) and all edges connecting such nodes. $C_k^{(0)}$ and $C_k^{(1)}$ are known to be k dimensional hypercubes so, by induction, there exist

$$\tau^{(i)}: U_{k-1}^{(i)} \rightarrow C_k^{(i)} \quad \text{for } i = 0, 1$$

Let the superscripts 0 and 1 denote the T_{k-1}, r, s and l_j of the corresponding $U_k^{(i)}$. Since the embedding $\tau^{(i)}$ is independent of the labeling of the nodes, make $(s^{(0)}, l_2^{(0)})$ and $(r^{(1)}, s^{(1)})$ isomorphic edges.

Now notice that there is a k level full binary tree rooted at $r = s^{(0)}$ with left subtree $T_{k-1}^{(0)}$ and right subtree $T_{k-1}^{(1)}$. Call this tree T_k. Set $s = l_2^{(0)}$. Note $l_2^{(0)}$ is not a node of T_k, but $(r, s) = (s^{(0)}, l_2^{(0)})$ is an edge of C_{k+1}. Set $l_j = l_{j-1}^{(1)}$ for $j > 2$. Then, for $(j > 2)$, $(l_j, l_{j+1}) = (l_{j-1}^{(1)}, l_j^{(1)})$ are edges in C_{k+1}. Finally, $l_{k+2} = l_{k+1}^{(1)} = l_{(k-1)+2}^{(1)}$ so there is a leaf node l of $T_{k-1}^{(1)}$ (which is also a leaf of T_k) such that (l_{k+2}, l) is an edge of C_{k+1}. This is an embedding since all edges of U_k were preserved and the nodes were mapped in a one to one fashion. QED

Theorem 2.1 now follows from the lemma. It should be noted in the proof of the lemma that the l_j's are chosen with each edge (l_j, l_{j+1}) representing a change in a new dimension. Thus, the path $(l_1, l_2, \ldots, l_{n+2})$ is a path between antipodal points in C_{n+1} (l_{n+2} is the $(n+1)$–*bit* one's complement of s in the binary labelling).

3. EMBEDDING RECTANGULAR MESHES

A five by five rectangular mesh has 25 nodes. Can this mesh be embedded in a 5-cube? The answer is no, but it will embed in a 6-cube. A 9x9x9 cubic mesh has 729 nodes but the smallest cube containing this mesh is a 12-cube with 4096 nodes. The general result is as follows:

Theorem 3.1: The smallest hypercube which contains an $l_1 \times l_2 \times \cdots \times l_k$ rectangular mesh has dimension $n = d_1 + d_2 + \ldots + d_k$ where $d_i = \lceil \log_2 l_i \rceil$ for all i.

Proof:

Let M be a $l_1 \times l_2 \times \cdots \times l_k$ rectangular mesh. Identify M with

$M \equiv L_1 \times L_2 \times \cdots \times L_k$ where

$L_i = \{0, 1, 2, \ldots, l_i - 1\}$.

Let $\mathbf{0}$ denote $0 \times 0 \times \cdots \times 0$ in M. Let C_n be the smallest hypercube such that $\tau: M \rightarrow C_n$ is an embedding. Assume the standard binary labelling of the hypercube (so that $\tau(m)$ is an integer in $[0, 2^n - 1]$ and that adjacent nodes of the hypercube differ only in one bit). Assume, for convenience, that $\tau(0) = \mathbf{0}$. For any point $m \epsilon M$ let m_i be the i^{th} coordinate of m.

Using composite Grey codes, it is easy to embed M in a cube with $n = d_1 + d_2 \cdots + d_k$ [1]. We will show that no smaller cube is sufficient.

For each coordinate j, define the j^{th} injection map $\tau^j : L_j \rightarrow C_n$ via $\tau^j(\lambda) = \tau(m)$ where

$$m_i = \begin{cases} 0 & \text{if } i \neq j \\ \lambda & \text{if } i = j \end{cases}$$

Lemma 3.2: $\tau(m) = \bigoplus_j \tau^j(m_j)$ for all $m \epsilon M$, where $x \oplus y$ is the bitwise exclusive-or function. (The exclusive-or function is associative, commutative and satisfies the two identities $x \oplus x = 0$ and $0 \oplus x = x$).

Proof by induction:

Order the points of the lattice lexicographically and let m be the earliest point for which the Lemma fails. Since $0 \oplus x = x$ the lemma is true if m has only one non-zero coordinate. So m must have two non-zero coordinates, say m_i and m_j.

Consider the little square in the $i-j$ plane:

$$c = (\ldots, m_i-1, \ldots, m_j, \ldots) \text{———} m = (\ldots, m_i, \ldots, m_j, \ldots)$$

$$a = (\ldots, m_i-1, \ldots, m_j-1, \ldots) \text{———} b = (\ldots, m_i, \ldots, m_j-1, \ldots)$$

Let
$$z = \bigoplus_{r \neq i,j} \tau^r(m_r)$$

Then by induction:

$$\tau(a) = \tau^i(m_i-1) \oplus \tau^j(m_j-1) \oplus z$$

$$\tau(b) = \tau^i(m_i) \oplus \tau^j(m_j-1) \oplus z$$

$$\tau(c) = \tau^i(m_i-1) \oplus \tau^j(m_j) \oplus z$$

But $\tau(a)$, $\tau(b)$, $\tau(m)$, and $\tau(c)$ form a cycle in the hypercube and hence:

$$0 = \tau(a) \oplus \tau(b) \oplus \tau(c) \oplus \tau(m)$$

Applying $\tau(m) \oplus \_\_$ to both sides yields

$$\tau(m) = \tau(a) \oplus \tau(b) \oplus \tau(c).$$

Substituting the above values and simplifying yields.

$$\tau(m) = \tau^i(m_i) \oplus \tau^j(m_j) \oplus z$$

This contradiction proves the Lemma.

For each coordinate line $p^j = \{m \in M \mid m_i = 0 \text{ for all } i \neq j\}$ define $\alpha_i^j = \tau^j(i) \oplus \tau^j(i+1)$ for $i = 0, 1, \ldots, l_j-2$. Since $\tau^j(i)$ and $\tau^j(i+1)$ are adjacent nodes in the hypercube, α_i^j is a power of 2 which indicates which bit is different.

Lemma 3.3: $\alpha_i^j \neq \alpha_s^r$ for $j \neq r$.

Proof:

Suppose $\alpha_i^j = \alpha_s^r$ for $j \neq r$. Restrict consideration to the j,r coordinate plane and consider the three elements

By Lemma 3.2:

$$\tau(i,s+1) = \tau^j(i) \oplus \tau^r(s+1)$$

$$\tau(i+1,s) = \tau^j(i+1) \oplus \tau^r(s)$$

But by definition of the α's:

$$\tau(i,s+1) = \tau^j(i) \oplus \tau^r(s) \oplus \alpha_s^r$$

$$\tau(i+1,s) = \tau^j(i) \oplus \tau^r(s) \oplus \alpha_i^j$$

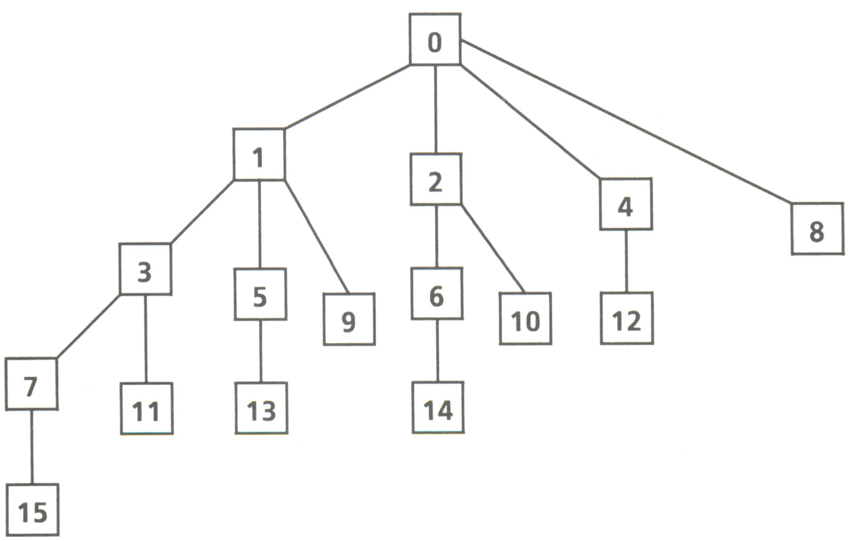
so if $\alpha_s^r = \alpha_i^j$ then $\tau(i,s+1) = \tau(i+1,s)$ which contradicts τ being an embedding.

So the coordinates use disjoint sets of bits in the hypercube and so

$$n \geq d_1 + \cdots + d_k$$

4. BROADCAST TREES

Sometimes it is necessary for one node in a hypercube machine to pass the same message to all of the other nodes. This could be done by using $NP-1$ sequential sends. A better balanced approach would be to treat the cube as a ring and send the message sequentially around the ring. A still better approach is to use a fanout tree in which each node sends the message to all of its neighbors which have not already received it. Assume that each node sends to its neighbors by increasing node number, the tree starting at 0 for C_4 is

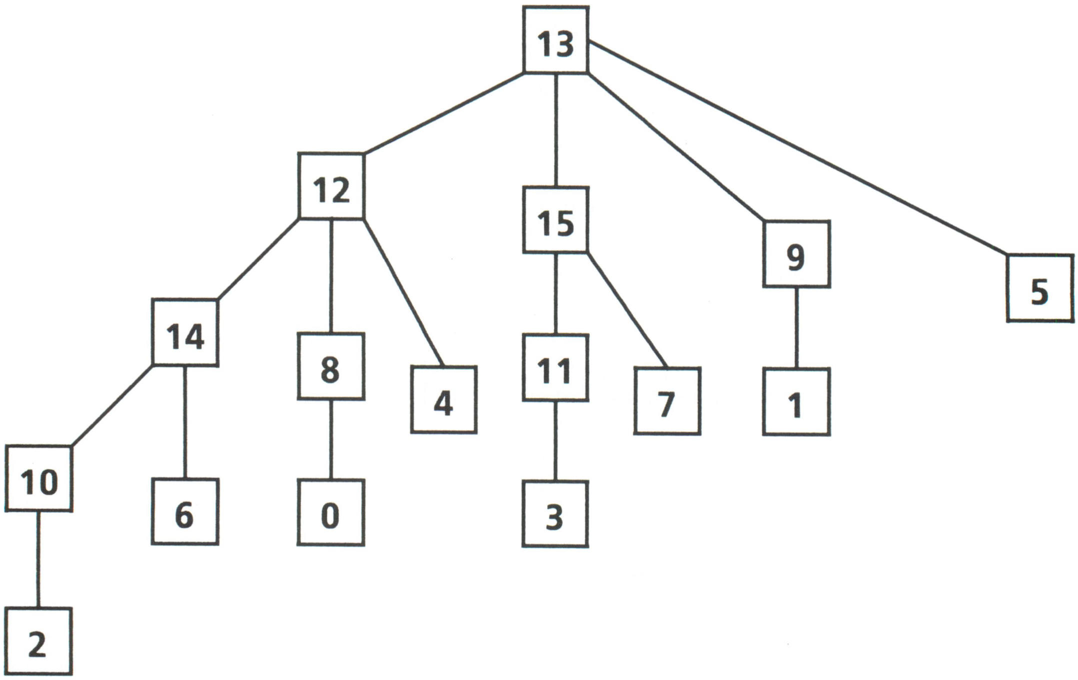


The neighbors of any node x in a hypercube are of the form

$$x \oplus 2^i \text{ for } i = 0, 1, 2, \ldots, n-1$$

Each node in the send algorithm sends only to those neighbors $x \oplus 2^j$ such that $2^j > x$.

The tree rooted at any other node r can be computed by taking the exclusive or of every tree node with r. For example, the C_4 tree rooted at $r = 13$ is



It is easy to compute this tree from the old one, given the root. What is not obvious is that this tree can be implemented without each node knowing the root. They need only know who sent them the message.

Theorem 3.1: If the root r sends to everybody and each other node x, which receives a message from z, sends it on to all neighbors $y = x \oplus 2^j$ such that $2^j > x \oplus z$, then the resulting communication tree is the same as that obtained from exclusive or with r of the tree rooted at zero.

Proof:

If T_r is the tree rooted at r and T_0 is the tree rooted at zero, then $T_0 \cong T_r$ with the map given by $r \oplus \_\_$ in both directions.

If x is in T_r then $x \oplus r$ is its image in T_0. $x \oplus r$ in T_0 sends to all neighbors $x \oplus r \oplus 2^j$ such that $2^j > x \oplus r$. Mapping back to T_r yields:

x sends to all neighbors $x \oplus 2^j$ such that $2^j > x \oplus r$.

If z is the parent of x in T_r, then $r \oplus z$ is the parent of $r \oplus x$ in T_0 and $x \oplus z = (r \oplus z) + (r \oplus x)$. So it is only necessary to verify that a node q in T_0 sends to all neighbors

$q \oplus 2^j$ such that $2^j > q \oplus p$ where p is the parent of q. That is, we need to verify that for the tree rooted at zero, $2^j > q$ and $2^j > q \oplus p$ yield the same set of neighbors.

However, by the way T_0 is constructed, the smallest $2^j > q$ is equal to $2(p \oplus q)$ and hence it is also the smallest power of 2 greater than $p \oplus q$.

5. REFERENCES

[1]

Youcef Saad and Martin Schultz
Topological Properties of Hypercubes
Research Report YALEU/DCS/RR-389 June, 1985
Computer Science Department
Yale University

Printed in USA SC6001/1k/WCP/VJC
Order Number 280182-001

intel®

INTEL SCIENTIFIC COMPUTERS
15201 N.W. Greenbrier Parkway
Beaverton, Oregon 97006
(503) 629-7629