# Yale University
# Department of Computer Science


## Spanning Balanced Trees in Boolean Cubes


Ching-Tien Ho and S. Lennart Johnsson



YALEU/DCS/TR-508
January 1987



This work has in part been supported by the Office of Naval Research under contracts N00014-84-K-0043 and N00014-86-K-0564. Approved for public release: distribution is unlimited.

## Table of Contents

# Spanning Balanced Trees in Boolean Cubes

Ching-Tien Ho and S. Lennart Johnsson

Department of Computer Science
Yale University
New Haven, CT 06520

## Abstract

A *Spanning Balanced n-tree* (SBnT) in a Boolean $n$-cube is a spanning tree in which the root has fanout $n$, and all the subtrees of the root have approximately the same number of nodes. The balanced spanning $n$-tree allows for scheduling disciplines that realize lower bound one-to-all personalized communication, all-to-all broadcasting, and all-to-all personalized communication on a Boolean $n$-cube [6, 9]. We give distributed routing algorithms defining the balanced spanning $n$-tree, and state and prove several of its properties.

## 1. Introduction

In recent years several multiprocessor architectures consisting of processors with local storage interconnected as Boolean $n$-cubes have been proposed and built, and are now commercially available. The number of dimensions for such systems currently range from 5 to 12. Multiprocessor architectures with a large number of processors are sometimes referred to as *ensemble architectures*, [12]. Communication, in such architectures, for a large class of computations is critical for overall performance. For some computations nearest-neighbor communication, in some topologies that can be embedded in the Boolean cube preserving proximity, suffices. Examples of the topologies are linear and multidimensional arrays [10], butterfly (FFT) networks, and binary trees [10, 3, 2, 4]. For other operations global communication is required, such as the broadcasting of data from a single node to a subset of other nodes, or all other nodes, or the sending of a unique data set to every other node, or a subset thereof. If the subset is comprised of all the nodes in a subcube, then the communication problem is equivalent to that of communicating to every node in a cube, but the size is reduced. Many linear algebra algorithms can make efficient use of broadcasting [10], and certain matrix transpose algorithms and other permutations [7] effectively use *one-to-all* or *all-to-all personalized communication* [6].

For global communication some form of spanning graph is necessary. In [6, 9] we studied routing and scheduling for *one-to-all broadcasting*, *all-to-all broadcasting*, *one-to-all personalized communication*, and *all-to-all personalized communication* with two different assumptions on the communication bandwidth of each node: *one-port* at a time, and *n-ports* concurrently. In several of these communication operations a *Spanning Balanced n-tree* (SBnT) makes possible scheduling disciplines that realize lower bound communication. With a balanced $n$-tree we mean a tree with fanout $n$ at the root, and approximately the same number of nodes in each subtree of the root.

In this paper we give a definition of the SBnT. This definition can be used as a distributed routing algorithm. Then we analyze some of the properties of the SBnT and give bounds on the deviation from a perfectly balanced $n$-tree. We also discuss some alternative definitions of the SBnT, and provide a solution to minimizing the maximum edge load for *one-to-all personalized communication*, i.e., making the edge load equal to the lower bound. The alternative definitions of the SBnT have the same distribution of nodes among subtrees, but the subtrees have different topologies, and use a different set of edges in the cube. The fact that a different set of edges is used is important with respect to fault-tolerance. Clearly, malfunction of edges that are not part of the SBnT is of no consequence for the communication, so the issue of fault tolerance for communication making use of the SBnT is to find an SBnT that does not include the faulty edges. There exist different SBnTs that only share the edges emanating from the root, i.e., that are edge-disjoint below level 1 with the root at level 0. Similarly, the different definitions of the SBnT have different sets of leaf nodes, giving some flexibility for reducing the consequences of node failure.

In section 2 the notations and definitions used throughout this paper are introduced. Section 3 contains a definition of the SBnT, and an analysis of its properties. Section 4 gives some alternative definitions of the SBnT and compares the characteristics of these alternative SBnTs with those of the SBnT in section 3. A modification of the SBnT into a spanning graph with minimax edge load is given in section 5. In section 6 we prove and give some complexity estimates for personalized communication based on the SBnT.

## 2. Notations and Definitions

For the definition of the balanced spanning $n$-tree we make use of *rotations* and *translations*. In the following, $R$ denotes the right-rotation function defined by $R(i) = (a_0 a_{n-1} a_{n-2} \ldots a_1)$, where $i = (a_{n-1} a_{n-2} \ldots a_0)$ is a node address, and $R^j = R^{j-1} \circ R$ means a right rotation of $j$ steps. The inverse operation $R^{-1} = L$ is a left rotation, i.e., $R^{-1}(i) = L(i) = (a_{n-2} a_{n-3} \ldots a_1 a_0 a_{n-1})$. The *rotation* of a graph with binary node labels is accomplished by applying the same rotation function to all its labels. Similarly, the *translation* of a graph is accomplished by performing a bit-wise exclusive-or operation on all the labels. For alternative definitions of the spanning balanced n-tree we also make use of the *bit-reversal* operation $B(i) = (a_0 a_1 \ldots a_{n-1})$. Clearly, adjacency is preserved under rotation, translation, and bit-reversal. Translation preserves the order of dimensions, rotation the relative order of dimensions, cyclically; but bit-reversal implies a permutation of the dimensions. Moreover, the bit-reversal operation is its own inverse, and the following relationship between bit-reversal and rotation holds: $RBR = B$, $\quad LBL = B$, $\quad R = BLB$, and $L = BRB$.

The period of a binary number $i$, $P_i$, is the least $j > 0$ such that $i = R^j(i)$. For example, the period of (011011) is 3. A binary number is *cyclic* if its period is less than its length and it is *non-cyclic* otherwise. Note that complementation of a binary number preserves the period; hence complementation of a cyclic number is a cyclic number, and complementation of a non-cyclic number is a non-cyclic number. A *relative address* of a node $i$ in a spanning tree rooted at node $s$ is $i \oplus s$. A *cyclic node* is a node with cyclic relative address. Note that a cyclic node is defined only when the source node is given. If there exists a $j$ such that $R^j(i) = k, \quad i \neq k$, then $i$ and $k$ are in the same *generator set* $G$ (or *necklace*[11]). For example, (001001), (010010) and (100100) are in the same generator set. The numbers (110000), (011000), (001100), (000110), (000011) and (100001) are also in the same generator set (but a different set from the preceding ones). The number of elements in the generator set $G_i$ of $i$ is the period of $i$, $P_i$. We use the notation $a|b$ to denote that $a$ divides $b$.

The most familiar spanning tree for a Boolean cube is a Spanning Binomial Tree [5, 1, 6]. The children-nodes of any node in such a tree is defined by complementing leading zeroes in the relative address of the node. For the definition of the Spanning Balanced $n$-tree we also consider a block of consecutive zeroes. However, this block is not necessarily a leading block of zeroes, but rather the leading block of zeroes after a right rotation that minimizes the value of the address. The reason for this choice will be discussed later. Let $\mathcal{M}(i,j)$ be the maximum set of consecutive indices containing all the 0-bit positions immediately to the right of bit $j$, cyclically. Bit $j$ is always a 1-bit in our definition of the SBnT. Hence, $|\mathcal{M}(i,j)|$ is the number of leading zeroes of $R^j(i)$. Formally, let $k$ be the position of the first 1-bit to the right of bit $j$, cyclically. (If $|i| = 1$ then $j = k$). If $i = 0$ then $\mathcal{M}(i,j) = \{n-1, n-2, ..., 0\}$. Hence,

$$\mathcal{M}(i,j) = \begin{cases} \{n-1, n-2, ..., 0\}, & \text{if } i = 0; \\ \{j-1, j-2, ..., k+1\}, & \text{if } j > k; \\ \{j-1, j-2, ..., 0, n-1, n-2, k+1\}, & \text{if } k \geq j. \end{cases}$$

For example, $i = (010110)$, $j = 1$ then $k = 4$ and $\mathcal{M}((010110), 1) = \{0, 5\}$. Similarly, $\mathcal{M}((010110), 4) = \{3\}$.

In a spanning binomial tree the root has $n$ subtrees, with the number of nodes in the subtrees being $2^k$ for $k = \{0, 1, \ldots, n-1\}$. Let $i = (a_{n-1}a_{n-2}\ldots a_0)$ be a node in a subtree and let $j$ be such that $a_j = 1$ and $a_l = 0, \quad l < j$. Then all nodes in the same subtree have the same value of $j$. We let $j$ be the label of the subtree, and $j$ can be considered a *base* for the subtree. Note that after a right rotation of $j$ steps, the least significant bit is 1. For the SBnT we modify the definition of the base such that $j$ is selected as the rotation that minimizes $R^j(i)$. If the value of $j$ minimizing $R^j(i)$ is not unique, then we pick the base as the smallest value of $j$. Subtrees are labeled $\{0, 1, \ldots, n-1\}$.

One of the operations for which the SBnT allows scheduling disciplines, making lower-bound communication possible, is *one-to-all personalized communication* [6]. In this form of communication one node sends a distinct piece of information to every other node. In *all-to-all personalized communication* every node sends a distinct piece of information to every other node.

### 3. A Spanning Balanced n-Tree

We first define a spanning tree rooted at node 0. A spanning tree for an arbitrary location is obtained through translation. We make this translation implicit in the definition of the balanced spanning tree for an arbitrary root node. Our definition can serve as a distributed routing algorithm. For the complexity estimates of various communication operations it is of interest to characterize the distribution of nodes among the subtrees of the root, as well as the fanout for the nodes. We give a lower bound on the number of nodes in a subtree that is low by at most a term of approximately $2\frac{N}{n^2}$, and an upper bound that is twice the average number of nodes in a subtree. We also present a table for the actual number of nodes in the maximum and minimum subtrees generated by the SBnT algorithm for up to 20-dimensional cubes, and show that the relative difference approaches 0 as the number of nodes grow.

Let $J_i = \{j_1, j_2, \ldots, j_m\}$, where $0 \leq j_1 < j_2 < \cdots < j_m < n$, $R^u(i) = R^v(i), \quad u, v \in J_i$, and $R^u(i) < R^l(i), \quad u \in J_i, \quad l \notin J_i$. $|J_i| = \frac{n}{P_i}$ where $P_i$ is the period of $i$. Then $base(i) = j_1$ and

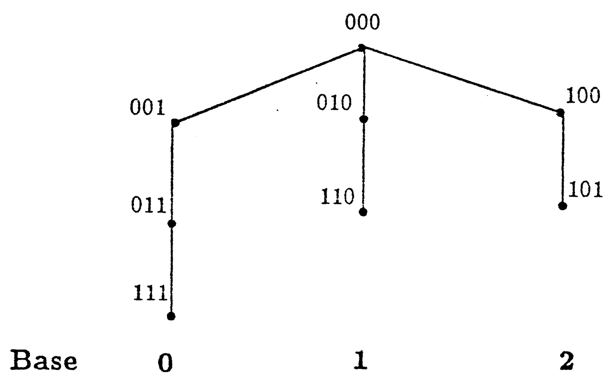



**Figure 1:** A spanning balanced 3-tree in a 3-cube.

node $i$ is assigned to subtree $j_1$, i.e., the value of the base is equal to the minimum number of right rotations such that the rotated number has a minimum value among all the rotated values. The notion of *base* is similar to the notion of "distinguished node" used in [11], in that $base = 0$ distinguishes a node from a generator set (necklace). For non-cyclic nodes the cardinality of the set $J_i$ is 1 and there is only one possible choice of base. For cyclic nodes there are at least two choices. The number of paths from the source node to each cyclic node $i$ is equal to $|J_i|$. By selecting a particular rotation from the set $J_i$ we define the subtree to which the node is assigned. For example, $base((011010)) = 3$ and $base((110110)) = 1$. The period of (011010) is 6 and the period of (011011) is 3. For ease of notation we omit the subscript on $j$ in the following. For the definition of the *parent* and *children* functions we first find the position $k$ of the first bit cyclically to the right of bit $j$ that is equal to 1, i.e., $a_k = 1$, and $a_m = 0, \forall m \in \mathcal{M}(i,j)$. $k = j$ if $i = (0...01_j0...0)$, and $k = -1$ if $i = 0$. Then:

$$children_{SBnT}(i,0) = \begin{cases} \{(a_{n-1}a_{n-2}...\overline{a}_m...a_0)\}, \forall m \in \{0,1,\ldots,n-1\}, & \text{if } i = 0; \\ \{q_m = (a_{n-1}a_{n-2}...\overline{a}_m...a_0)\}, & \\ \quad \forall m \in \mathcal{M}(i,j) \text{ and } base(q_m) = base(i), & \text{if } i \neq 0. \end{cases}$$

$$parent_{SBnT}(i,0) = \begin{cases} \phi, & \text{if } i = 0; \\ (a_{n-1}a_{n-2}...\overline{a}_k...a_0), & \text{otherwise.} \end{cases}$$

The $parent_{SBnT}$ function preserves the base, since for any node $i$ with base $j$, $k$ is the highest-order bit of $R^j(i)$ that is 1. Complementing this bit cannot change the base. It is also readily seen that the $parent_{SBnT}$ and $children_{SBnT}$ functions are consistent.

**Lemma 3.1.** *The $parent_{SBnT}$ ($children_{SBnT}$) function defines a spanning tree rooted at node* 0.

*Proof.* The parent node of a node at distance $d$ from node 0 is at distance $d-1$ from node 0, and each node only has one parent node. Traversing the edges defined by successive applications of the $parent_{SBnT}$ function of any node generates a path to node 0 for any node. Hence, the graph is a spanning tree rooted at node 0.

∎

Figures 1, 2 and 3 shows spanning trees generated by the algorithm above for the root located at node 0 in 3-, 4- and 5-cubes. Figure 4 shows subtree 0 of an SBnT in a 6-cube, in which the nodes in square boxes are cyclic.

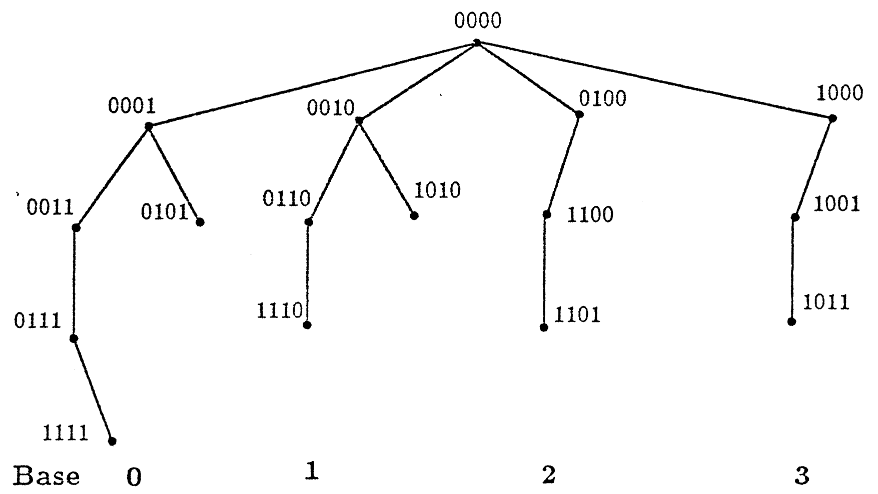



**Figure 2:** A spanning balanced 4-tree in a 4-cube.

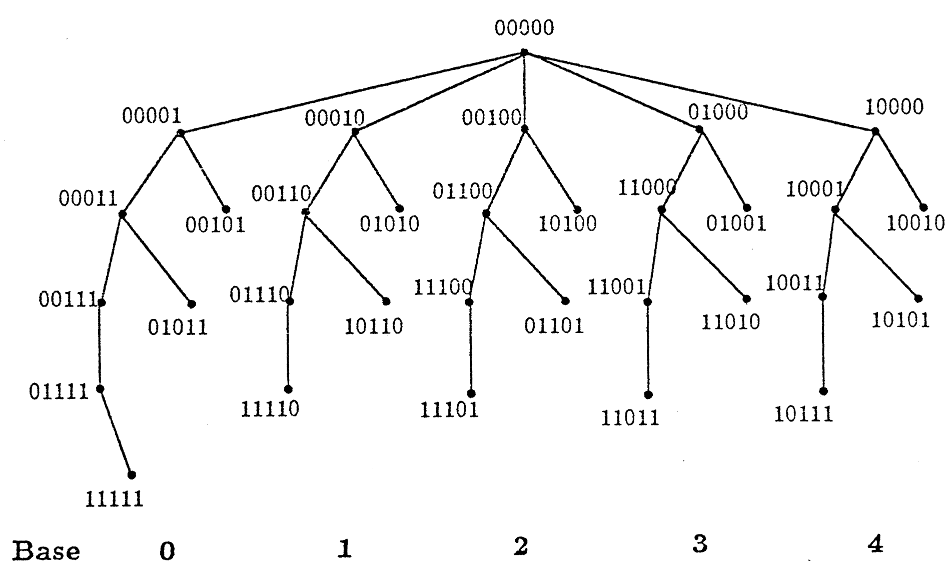



**Figure 3:** A spanning balanced 5-tree in a 5-cube.

For an arbitrary source node $s$ we translate the SBnT rooted at node 0 to node $s$ by performing for each node the bit-wise exclusive-or function of its address and the address of the source node. The base of a node is determined from $c = i \oplus s$, and the children and parent functions are readily modified.

Let $J_{i,s} = \{j_1, j_2, \ldots, j_m\}$, where $0 \leq j_1 < j_2 < \cdots < j_m < n$, $R^u(c) = R^v(c)$, $\quad u, v \in J_{i,s}$, and $R^u(c) < R^l(c)$, $\quad u \in J_{i,s}$, $\quad l \notin J_{i,s}$. Then $base(c) = j_1 = j$ and $k$ is defined by $c_k = 1$ and $c_m = 0, \forall m \in \mathcal{M}(c, j)$, with $k = -1$ if $c = 0$.

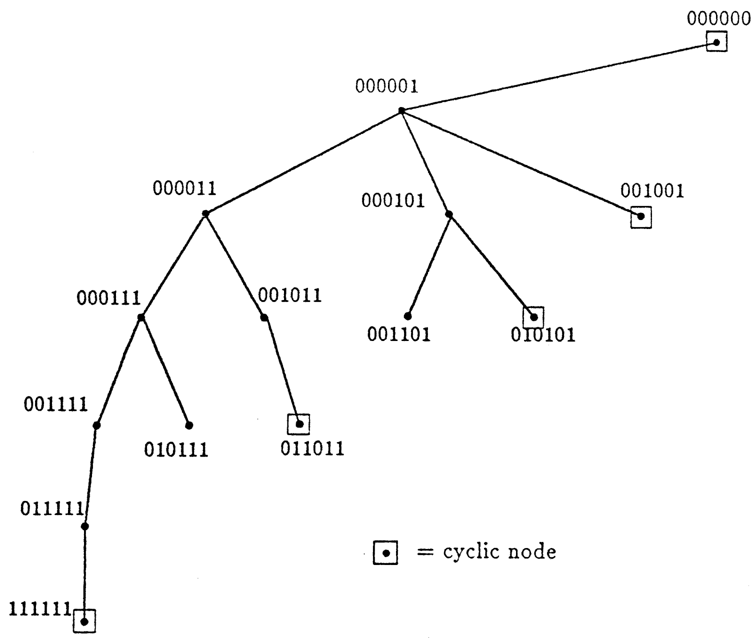



**Figure 4:** Subtree 0 of a spanning balanced 6-tree in a 6-cube.

$$children_{SBnT}(i,s) = \begin{cases} \{(a_{n-1}a_{n-2}...\overline{a}_m...a_0)\}, \forall m \in \{0,1,\ldots,n-1\}, & \text{if } c = 0; \\ \{q_m = (a_{n-1}a_{n-2}...\overline{a}_m...a_0)\}, & \\ \quad \forall m \in \mathcal{M}(i \oplus s, j) \text{ and } base(q_m \oplus s) = base(i \oplus s), & \text{if } c \neq 0. \end{cases}$$

$$parent_{SBnT}(i,s) = \begin{cases} \phi, & \text{if } c = 0; \\ (a_{n-1}a_{n-2}...\overline{a}_k...a_0), & \text{otherwise.} \end{cases}$$

We now state and prove some of the properties of the SBnT graph.

**Lemma 3.2.** *The number of nodes at level $l$ is $\binom{n}{l}$.*

*Proof.* From the $parent_{SBnT}$ function it follows that node $i$ with $|i| = l$ is at level $l$. Furthermore, there exist $\binom{n}{l}$ distinct permutations of $l$ 1−bits out of a string of $n$ bits.

∎

The SBnT is a ***greedy*** **tree[9] in the sense that the number of nodes at distance** $l$ **from the root is the same as for the** $n$**-cube, i.e., the distance for any node in the SBnT to the root is minimal.**

**Corollary 3.1.** *The height of one subtree is $n$, and the height of all other subtrees is $n-1$.*

*Proof.* There is only one node at distance $n$ from the root, and there are $n$ nodes at distance $n-1$ from the root, each of which has a different base.

∎

**Lemma 3.3.** *The maximum fanout of a node at level $l$ is $\lceil \frac{n-l}{2} \rceil$, for $1 \leq l \leq n$.*

*Proof.* Let the relative address $i = 2^l - 1$. Then $|i| = l$ and complementing bits $\{l, l+1, \ldots l + \lceil \frac{n-l}{2} \rceil - 1\}$ does not change the base, but complementing the higher-order bits does. Hence, the maximum fanout is at least $\lceil \frac{n-l}{2} \rceil$. But, for any other $i$ such that $|i| = l$, the maximum size of any block of consecutive zeroes, cyclically, is also $n - l$; hence $|\mathcal{M}| \leq n - l$, and the argument can be applied to the leading block of $R^j(i)$, where $j$ is the base, and the proof is complete.

∎

**Lemma 3.4.** *Let $\phi(i,j)$ be the number of nodes at distance $j$ from node $i$ in the subtree rooted at node $i$. Then, $\phi(i,j) \geq \phi(k,j)$ where node $k$ is a child of node $i$.*

*Proof.* Let $k$ be any non-root node and node $i$ its parent node. Furthermore, we let *address* mean *relative address.* We prove the lemma first for subtree 0 by showing that for any node at distance $j$ below node $k$, there is a unique corresponding node at distance $j$ below node $i$. Let $\alpha$ be the number of leading 0's of $k$. Clearly, the number of leading 0's of $i$ must be $\alpha + \beta + 1$ where $\beta \geq 0$ is the number of consecutive 0's in $k$ between the two leftmost 1-bits. Any node at distance $j$ below $k$, say $k_j$, has an address that can be derived by complementing $j$ out of the $\alpha$ leading 0's of $k$, and with the base unchanged. There exists a corresponding node $i_j$ at distance $j$ below node $i$. The address of node $i_j$ can be constructed by leaving $\beta + 1$ leading 0's of the address of $i$ unchanged, and making the following $\alpha$ bits equal to the first $\alpha$ bits of node $k_j$. Hence, the address of node $i_j$ is derived by changing bit $n - \alpha - 1$ of $k_j$ from 1 to 0, and by moving this 0-bit and the following $\beta$ 0-bits together to the leftmost positions, and shifting the leading $\alpha$ bits $\beta + 1$ steps to the right. This process preserves the base for any given node $k_j$. The same argument applies to any other subtree $j$ by considering $R^j(i)$ and $R^j(k)$.

∎

The property in lemma 3.4 guarantees that the root is the bottleneck for every routing step in personalized communication.

**Lemma 3.5.** *Excluding node $i \oplus s = (11\ldots1)$, all the subtrees of the SBnT are isomorphic if $n$ is a prime number.*

*Proof.* Since $n$ is a prime number, there are no cyclic nodes except nodes $(00\ldots0)$ and $(11\ldots1)$. Excluding the node with relative address (11...1), all other nodes in the subtrees are non-cyclic. Since different subtrees are obtained through rotations of the addresses, they are isomorphic. The proof is completed by noticing that a translation does not alter the topology.

∎

**Corollary 3.2.** *The subtrees of the root of the SBnT are isomorphic if cyclic nodes are excluded.*

**Corollary 3.3.** *If $n$ is prime, the number of edges in dimension $d$ between levels $l$ and $l+1$ is equal to $\frac{1}{n}\binom{n}{l+1}$ for $l = \{0, 1, \ldots, n-2\}$. For $l = n-1$ there is only one edge, and it is in dimension $n-1$. The total number of edges in dimension $d$ is equal to $\frac{N-2}{n} + 1$ for dimension $n-1$ and $\frac{N-2}{n}$ for the other dimensions.*

**Definition 3.1.** A *treetop* of a tree T is a tree which is a connected subgraph of T containing the root of T.

**Lemma 3.6.** *Subtree $i$ of the root of an SBnT is isomorphic to a treetop of subtree $j$ of the root of the SBnT if $i > j$.*

*Proof.* Subtree $i$ is derived from subtree $j$ by pruning away cyclic nodes and incident edges such that the period $P$ of the nodes satisfies the relation, $i \geq \frac{n}{P} > j$, and performing a right cyclic shift of $i - j$ bits for all nodes in the pruned subtree.

∎

**Corollary 3.4.** *Subtrees $P$ to $n - 1$, where $P$ is the length of the period of a cyclic node, contains no cyclic nodes with period $P$.*

**Corollary 3.5.** *There are $\frac{1}{2}n$ subtrees with no cyclic nodes if $n$ is even, and at least $\frac{2}{3}n$ subtrees with no cyclic nodes if $n$ is odd.*

**Lemma 3.7.** *Subtree 0 of an SBnT is isomorphic to a treetop of a Spanning Binomial Tree of an $(n-1)$-cube. Subtrees 1 through $n - 1$ of an SBnT are isomorphic to a treetop of a Spanning Binomial Tree of an $(n-2)$-cube.*

*Proof.* Consider the relative addresses of nodes in subtree 0 of an SBnT of an $n$-cube. They all have the form $(a_{n-1}a_{n-2}\cdots a_1 1)$, where $a_i = 0$ or 1. From the definition of the children function of an SBnT, if node $j$ is a child of node $i$ then node $j$ can be derived by complementing one of the leading 0-bits of node $i$. Recall that the children function of the spanning binomial tree(SBT) is defined by complementing any one of the leading 0-bits. Hence, subtree 0 of the SBnT is a treetop of an SBT of an $(n-1)$-cube. For the nodes in subtree 1 of the SBnT, they all have a relative address of the form $(a_{n-1}a_{n-2}\cdots a_2 10)$. Again, the addresses of the children can be derived by complementing one of the leading 0-bits. So, subtree 1 is a treetop of an SBT of an $(n-2)$-cube. By lemma 3.6, it follows that subtree 2 to subtree $n - 1$ are treetops of an SBT of an $(n-2)$-cube.

∎

**Lemma 3.8.** *Any cyclic node is a leaf node of the SBnT.*

*Proof.* From the definition of the base and children functions for a node $i$, it follows that connections to the children nodes are defined by a subset of the connections obtained by complementing any of the leading zeroes of $R^j(c)$, where $j$ is the base and $c = i \oplus s$. The subset is defined as those bit complementations that preserve the base. But, if $c$ is periodic, then the base is changed since the leading repetitive pattern of $c$ has a larger value than the following patterns.

∎

The imbalance between the subtrees of the root are caused by the cyclic nodes. We will now study the distribution of cyclic nodes in some detail. However, first we give a bound on the imbalance (the total number of cyclic nodes in a subtree).

**Theorem 3.1.** *The number of nodes in a subtree is of order $O(\frac{N}{n})$.*

*Proof.* With $A$ cyclic nodes there are at least $\frac{N-A}{n}$ nodes in a subtree. Denoting the number of generator sets for cyclic nodes by $B$, it follows that the maximum number of nodes in a subtree is $\frac{N-A}{n} + B - 1$. To derive bounds on $A$ we use the complex-plane diagram used by Hoey and Leiserson [8] in studying the shuffle-exchange network. Leighton[11] shows that $B = O(\sqrt{N})$.

Full necklaces, i.e., non-cyclic nodes, are mapped to circles. Degenerate necklaces, i.e., cyclic nodes, are mapped to the origin. In the context of the shuffle-exchange graph each node that is mapped to the origin of the complex plane is adjacent (via an exchange edge) to a node at position

| $n$ | $A$ | $B$ | SBT(max) | SBnT(max) | SBnT(min) | $(N-1)/n$ | factor |
|---|---|---|---|---|---|---|---|
| 2 | 2 | 2 | 2 | 2 | 1 | 1.50 | 1.33 |
| 3 | 2 | 2 | 4 | 3 | 2 | 2.33 | 1.29 |
| 4 | 4 | 3 | 8 | 5 | 3 | 3.75 | 1.33 |
| 5 | 2 | 2 | 16 | 7 | 6 | 6.20 | 1.13 |
| 6 | 10 | 5 | 32 | 13 | 9 | 10.50 | 1.24 |
| 7 | 2 | 2 | 64 | 19 | 18 | 18.14 | 1.05 |
| 8 | 16 | 6 | 128 | 35 | 30 | 31.88 | 1.10 |
| 9 | 8 | 4 | 256 | 59 | 56 | 56.78 | 1.04 |
| 10 | 34 | 9 | 512 | 107 | 99 | 102.30 | 1.05 |
| 11 | 2 | 2 | 1024 | 187 | 186 | 186.09 | 1.00 |
| 12 | 76 | 17 | 2048 | 351 | 335 | 341.25 | 1.03 |
| 13 | 2 | 2 | 4096 | 631 | 630 | 630.08 | 1.00 |
| 14 | 130 | 21 | 8192 | 1181 | 1161 | 1170.21 | 1.01 |
| 15 | 38 | 10 | 16384 | 2191 | 2182 | 2184.47 | 1.00 |
| 16 | 256 | 36 | 32768 | 4115 | 4080 | 4095.94 | 1.00 |
| 17 | 2 | 2 | 65536 | 7711 | 7710 | 7710.06 | 1.00 |
| 18 | 568 | 70 | 131072 | 14601 | 14532 | 14563.50 | 1.00 |
| 19 | 2 | 2 | 262144 | 27595 | 27594 | 27594.05 | 1.00 |
| 20 | 1036 | 111 | 524288 | 52487 | 52377 | 52428.75 | 1.00 |

**Table 1:** A comparison of subtree sizes of spanning sinomial trees and spanning balanced $n$-trees.

$(1,0)$ or $(-1,0)$. Hence, for every full necklace of $n$ nodes there are at most 2 cyclic nodes. Node 0 is adjacent to a node of a full necklace, and so is node $N-1$ (for $n>2$). It follows that an upper bound on $A$ is $2\frac{N-n}{n+2}$ and the number of nodes in a subtree is at least $\frac{N+2}{n+2}$. The relative difference in the number of nodes in the maximum and minimum subtrees approaches 0 for $N \to \infty$.

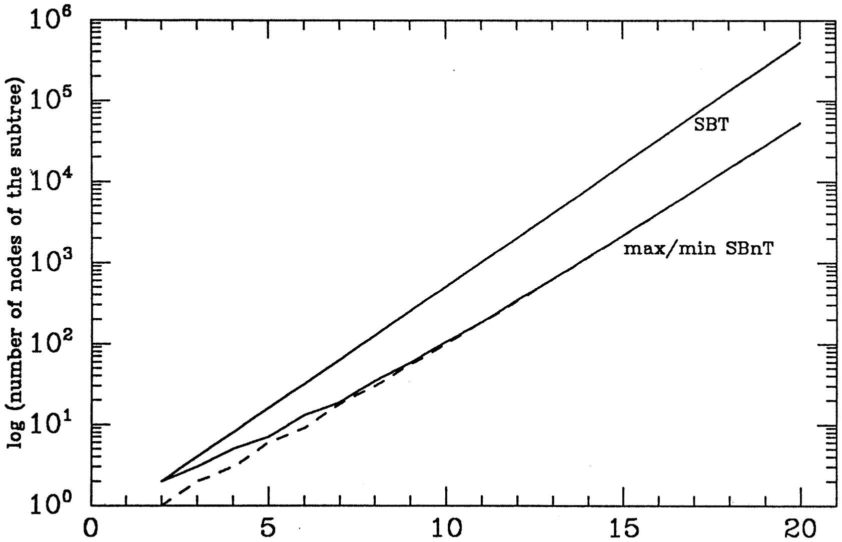
∎

Table 1 gives the sizes of the minimum and maximum subtrees generated according to the definition of the SBnT for up to 20-dimensional cubes. The relative difference approaches 0 rapidly. For comparison we have included the number of nodes in the largest subtree of the corresponding Spanning Binomial Tree. The last column contains the ratio of SBnT(max) to $\frac{N-1}{n}$. Figure 5 contains the same information as the table. The curves for the maximum and minimum SBnT become indistinguishable as the cube dimension increases.

In theorem 3.1 we showed that the total number of cyclic nodes are at most $2\frac{N-n}{2+n}$. We now first show that the ratio of the number of cyclic nodes at level $l$ to the total number of nodes at level $l$ is at most $\frac{2}{n}$, then that for any level of any subtree of the root the number of cyclic nodes is at most equal to the number of non-cyclic nodes at the same level of the same subtree, with the exception of the last level of subtree 0. We do that by defining a function as described in lemma 3.12 and showing that this function is one-to-one from each non-cyclic node to a unique cyclic node at the same level of the same subtree of the SBnT (except the root node, $s$, and the node at the last level of subtree 0, $\bar{s}$). Some properties of the period of a cyclic number are needed.

Note that

1. if $a|c$, and $b|c$ then $\mathrm{lcm}(a,b)|c$, where $a,b,c$ are integers,




From top to bottom: max subtree size of SBT, max subtree size of SBnT, min subtree size of SBnT.

**Figure 5:** Comparing subtree sizes of spanning binomial trees and SBnT.

2. and if $c$ is an *n-bit* cyclic number with period $P$, then $\frac{n}{P}||c|$.

**Lemma 3.9.** *Let $c_1$ and $c_2$ be two distinct n-bit cyclic numbers with periods $P_1$ and $P_2$ respectively and $|c_1| = |c_2|$. Then $\gcd(P_1, P_2) > 1$.*

*Proof.* Let $|c_1| = i$. Assume $\gcd(P_1, P_2) = 1$. Then $P_1|n, P_2|n$ and $\gcd(P_1, P_2) = 1$ imply $n = kP_1P_2$ for some positive integer $k$. By property 2, we have $\frac{n}{P_1}|i$, $\frac{n}{P_2}|i$. But $n = kP_1P_2 \Rightarrow kP_2|i$ and $kP_1|i$, which imply $kP_1P_2|i$, i.e., $n|i$, by property 1. But $0 < i < n$ since $c_1 \neq c_2$ and we have contradiction.

∎

**Lemma 3.10.** *Let $c_1$ and $c_2$ be two distinct n-bit cyclic numbers with periods $P_1$ and $P_2$ respectively, $P_1 \leq P_2$, $P_1|P_2$ and $|c_1| = |c_2|$. Then $Hamming(c_1, c_2) \geq \frac{2n}{P_2}$.*

*Proof.* We derive a lower bound for the Hamming distance between $c_1$ and $c_2$, by finding the minimum number of bits of $c_1$ that have to be complemented to yield $c_2$, for all possible $c_1$ and $c_2$. $c_2$ consists of $\frac{n}{P_2}$ blocks of length $P_2$ each. In order to change the period from $P_1$ to $P_2$ (or change from $c_1$ to $c_2$ if $P_1 = P_2$), at least one bit in each block of $P_2$ bits of $c_1$ should be complemented. So, at least $\frac{n}{P_2}$ bits of $c_1$ should be complemented. However, either all bits are changed from 1 to 0 or vice versa to maintain periodicity. Hence, the number of *1-bits* either decreases or increases by $\frac{n}{P_2}$ and thus $\frac{n}{P_2}$ bits should be changed in the opposite direction to satisfy $|c_1| = |c_2|$. So, the Hamming distance between $c_1$ and $c_2$ is at least $\frac{2n}{P_2}$.

∎

**Lemma 3.11.** *Let $c_1$ and $c_2$ be two distinct n-bit cyclic numbers with periods $P_1$ and $P_2$ respectively and $|c_1| = |c_2|$. Then $Hamming(c_1, c_2) \geq \frac{2n}{P_1P_2}(P_1 + P_2 - 2\gcd(P_1, P_2))$.*

*Proof.* Let $g = \gcd(P_1, P_2)$ and $c$ be an *n-bits* cyclic number with period $g$ and $|c| = |c_1|$. By lemma 3.10, the Hamming distance between $c$ and $c_1$ is at least $\frac{2n}{P_1}$. Similarly, the Hamming distance between $c$ and $c_2$ is at least $\frac{2n}{P_2}$. To obtain $c_1$ from $c$ we change 1-bits to 0-bits (and 0-bits to 1-bits) for every $P_1$ bits of $c$ to produce $c_1$, and change 1-bits to 0-bits (and 0-bits to 1-bits) for every $P_2$ of $c$ bits to produce $c_2$. The number of common bit positions of $c$ that has been changed to generate $c_1$ and $c_2$ is $\frac{2n}{\text{lcm}(P_1,P_2)}$, if we changed $\frac{2n}{P_1}$ and $\frac{2n}{P_2}$ bits of $c$ to convert it to $c_1$ and $c_2$ respectively. In general, $\frac{g}{P_2}$ of the bits we changed to generate $c_1$ and $\frac{g}{P_1}$ of the bits we changed to generate $c_2$ correspond to the common bit positions. So, the Hamming distance between $c_1$ and $c_2$ is at least $\frac{2n}{P_1} + \frac{2n}{P_2} - \frac{4n}{\text{lcm}(P_1,P_2)}$, i.e., $\frac{2n}{P_1P_2}(P_1 + P_2 - 2g)$.

∎

**Corollary 3.6.** *Let $c_1$ and $c_2$ be two distinct n-bit cyclic numbers with periods $P_1$ and $P_2$ respectively, $|c_1| = |c_2|$ and $\gcd(P_1, P_2) \neq P_1$ or $P_2$. Then $Hamming(c_1, c_2) \geq 6$.*

*Proof.* By lemma 3.11, $Hamming(c_1, c_2) \geq \frac{2n}{P_1P_2}(P_1 + P_2 - 2\gcd(P_1, P_2))$, i.e., $Hamming(c_1, c_2) \geq \frac{2n}{\text{lcm}(P_1,P_2)}(\frac{P_1+P_2}{\gcd(P_1,P_2)} - 2)$. The maximum value of $\text{lcm}(P_1, P_2)$ is $n$ and the minimum value of $\frac{P_1+P_2}{\gcd(P_1,P_2)}$ is 5 (for $\gcd(P_1, P_2) \neq P_1$ or $P_2$). So, $Hamming(c_1, c_2) \geq 6$ follows.

∎

**Corollary 3.7.** *Let $c_1$ and $c_2$ be two distinct n-bit cyclic numbers with periods $P_1$ and $P_2$ respectively, $|c_1| = |c_2| = i$. Then $Hamming(c_1, c_2) \geq 4$.*

*Proof.* If $\gcd(P_1, P_2) = P_1$ or $P_2$, then by lemma 3.10, $Hamming(c_1, c_2) \geq \frac{2n}{\max(P_1,P_2)}$. Since $\max(P_1, P_2) \leq \frac{n}{2}$, $Hamming(c_1, c_2) \geq 4$.

If $\gcd(P_1, P_2) \neq P_1$ or $P_2$, then by corollary 3.6, $Hamming(c_1, c_2) \geq 6$.

∎

**Corollary 3.8.** *Any node has at most one cyclic node as a child.*

*Proof.* It follows from corollary 3.7.

∎

The following theorem gives a bound on the ratio of the number of cyclic nodes at each level of the whole SBnT.

**Theorem 3.2.** *In an SBnT, the ratio of the number of cyclic nodes at level $l$, $0 < l < n$, to the number of nodes at the same level is at most $\frac{2}{n}$.*

*Proof.* To prove the theorem we show that for each cyclic number $i$ such that $|i| = l$, we can find a set $NC_i$ of non-cyclic numbers such that $|NC_i| = \frac{n}{2} - 1$ and for $j \in NC_i$, $|j| = l$. Moreover, the sets for different cyclic numbers are disjoint. A binary number consists of a *string* of bits. Let $f$ be a function that maps a string $s$ of length $q$ to a set of strings of the same length $S_f$. We define $f$ to be the function that exchanges any 0-bit with the rightmost 1-bit, or any 1-bit with the rightmost 0-bit. The number of strings in the set $S_f$ is 0 if $s$ contains all 0-bits or all 1-bits, and $q - 1$ otherwise. That $|S_f| = q - 1$ if $|s| \neq 0$ follows from the fact that each 1-bit and 0-bit determine a unique string, except that the rightmost 1-bit and the rightmost 0-bit determine the same string. For each cyclic string $s$ of length $n$ and period $P$, we first find the largest $P'$ satisfying $P|P'$, $P'|n$ and $P' < n$. Note that $P' \leq \frac{n}{2}$. We now want to generate $n - P' - 1$ non-cyclic strings from the given string $s$. The first $P'$ bits of these strings are the same as the first $P'$ bits of the string $s$. The last $n-P'$ bits of the strings are derived by applying the function $f$ to the last $n-P'$ bits of the string $s$. Each generated string is non-cyclic because the Hamming distance between string $s$ and each generated string is 2, and any two cyclic strings of same length and containing the same number of 1-bits have a Hamming distance of at least 4, corollary 3.7. Since $P' \leq \frac{n}{2}$, the number of non-cyclic strings generated is at least $\frac{n}{2}-1$. We now show that the sets $S_f$ generated by two distinct cyclic strings are disjoint. Let $c_1$ and $c_2$ be two distinct cyclic strings with periods $P_1$ and $P_2$ respectively, and $|c_1| = |c_2|$. Consider the following three cases:

- $P_1 = P_2$: Clearly, the two generated sets are disjoint because $c_1 \neq c_2$.
- $\gcd(P_1, P_2) \neq P_1$ or $P_2$: Since the Hamming distance between the generated strings and the given string is 2, and by corollary 3.6 the Hamming distance between $c_1$ and $c_2$ is at least 6, the two sets are disjoint.
- $\gcd(P_1, P_2) = P_1$ or $P_2$: Let $\gcd(P_1, P_2) = P_2$ without loss of generality, i.e., $P_1 | P_2$. Since the first $P_2$ bits of the generated strings of $c_1$ and $c_2$ are distinct, the two generated sets are disjoint.

∎

**Lemma 3.12.** *Let $c$ be a cyclic node with period $P$ and base $j$ at level $l$, $1 \leq l < n$, of the $j^{th}$ subtree of the SBnT. Let $R^j(c) = (r_1 r_2 \ldots r_{\frac{n}{P}})$ where $r_i, i = \{1, 2, \ldots, \frac{n}{P}\}$ are identical bit strings of length $P$. Complementing the first bit of $r_2$ (which is zero by definition of the base) to 1 and complementing the leftmost bit of $r_{\frac{n}{P}}$, which is 1 defines a one-to-one function that maps each cyclic node $c$ to a non-cyclic node at the same level of the same subtree.*

*Proof.* By corollary 3.7, the above mapping function maps from a cyclic node to a non-cyclic node. To prove that the mapping from cyclic nodes to non-cyclic nodes is one-to-one, we assume first that the base is 0, i.e., $c = (r_1 r_2 \ldots r_{\frac{n}{P}})$. Let $c_1$ and $c_2$ be two distinct cyclic nodes at level $l$, $1 \leq l < n$, of subtree 0 (i.e., base 0) with period $P_1$ and $P_2$ respectively. Let $c_1 = (r_1 r_2 \ldots r_{\frac{n}{P_1}})$ and $c_2 = (s_1 s_2 \ldots s_{\frac{n}{P_2}})$. Let $f$ be the mapping function stated in the lemma. In the following we assume the SBnT to be rooted at node 0 without loss of generality.

- $P_1 = P_2$: Then $r_i \neq s_i, \forall i$. Since $\frac{n}{P_1} \geq 2$, the last $P_1$ bits of $f(c_1)$ and $f(c_2)$ are distinct, i.e., at least one bit position has different values in $f(c_1)$ and $f(c_2)$.
- $P_1 \neq P_2$ and $\gcd(P_1, P_2) = P_1$ or $P_2$: Assume $P_2 | P_1$ without loss of generality. The first $P_1$ bits of $f(c_1)$ contain $\frac{lP_1}{n} - 1$ bits that are equal to 1, while the first $P_1$ bits of $f(c_2)$ contain $\frac{lP_1}{n}$ bits equal to 1. Hence $f(c_1) \neq f(c_2)$.
- $P_1 \neq P_2$ and $\gcd(P_1, P_2) \neq P_1$ or $P_2$: From lemma 3.6, $Hamming(c_1, c_2) \geq 6$. Since the Hamming distance between $c_1$ and $f(c_1)$ is 2 and the Hamming distance between $c_2$ and $f(c_2)$ is 2, $Hamming(f(c_1), f(c_2)) \geq 2$, i.e., $f(c_1) \neq f(c_2)$.

For non-zero bases, say $base(c) = b$, we consider $R^b(c)$ instead, i.e., the right cyclic shift of $c$ by $b$ bits. The arguments are similar.

∎

**Theorem 3.3.** *With the exception of the last level, the number of cyclic nodes at level $l$ of any subtree of an SBnT is at most the same as the number of non-cyclic nodes at the same level of the* *same subtree (or any subtree) of the SBnT.*

*Proof.* It follows from lemma 3.12.

∎

This theorem gives a loose bound on the ratio of the number of cyclic nodes at each level in each subtree. Figure 6 shows the ratio of the actual number of cyclic nodes to the total number of nodes for each level of up to 16-dimensional cubes for subtree 0. The bound given by the theorem is pessimistic, except for level 2 of a 4-cube.

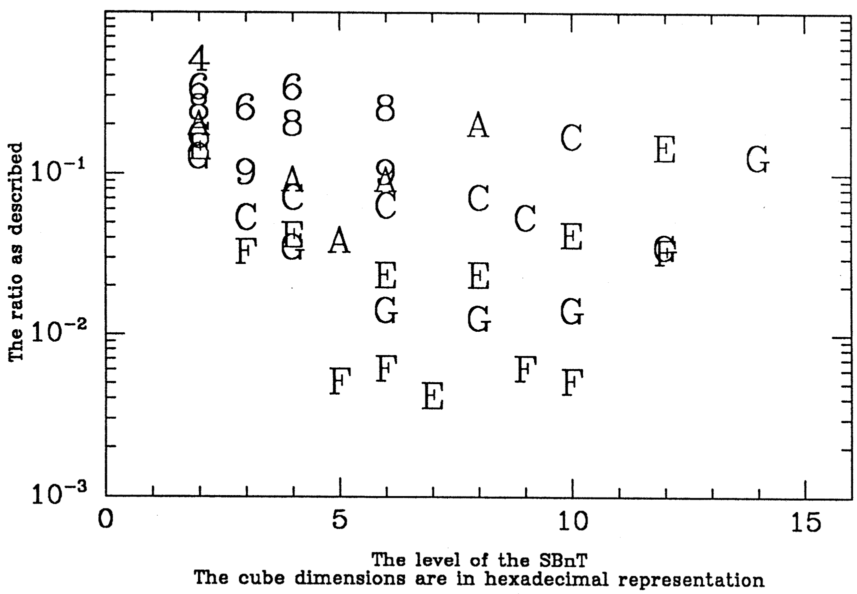



**Figure 6:** The ratio of the number of cyclic nodes to the total number of nodes at level $l$, $0 < l < n$, of subtree 0.

## 4. Other Choices of Spanning Balanced n-Trees

The above choice of *base*, *parent*, and *children* functions is somewhat arbitrary. In this section we discuss some other choices and compare them to the choice above. The different definitions of the SBnT use a different set of edges of the cube, with the exception that they all use all the edges directed away from the root. The reason for using all these edges is to minimize the maximum load on any edge in personalized communication. The fact that different definitions use different sets of edges is of importance with respect to fault tolerance.

An alternative to defining the base for a node $i$ as the minimum number of right rotations $j$ that minimizes $R^j(i)$ is to define the base as the minimum number of left rotations that maximizes $L^j(i)$. To distinguish the tree so defined from the previous spanning balanced n-tree, we refer to this tree as an SBnT-max and the previous tree as SBnT-min, whenever the difference between the two definitions is important. The formal definitions of base, parent and children functions are as follows: Let $k$ be the first 1-bit position to the left of bit $j$, cyclically. Define:

$$\mathcal{M}'(i,j) = \begin{cases} \{n-1, n-2, ..., 0\}, & \text{if } i = 0; \\ \{j+1, j+2, ..., k-1\}, & \text{if } j < k; \\ \{j+1, j+2, ..., n-1, 0, 1, k-1\}, & \text{if } k \leq j. \end{cases}$$

For an arbitrary source node $s$ let $J_{i,s} = \{j_1, j_2, \ldots, j_m\}$, where $0 \leq j_1 < j_2 < \cdots < j_m < n$, $L^u(c) = L^v(c)$, $u, v \in J_{i,s}$, and $L^u(c) > L^l(c)$, $u \in J_{i,s}$, $l \notin J_{i,s}$. Then $base(c) = j_1 = j$ and $k$ is defined by $c_k = 1$ and $c_m = 0, \forall m \in \mathcal{M}'(c,j)$, with $k = -1$ if $c = 0$.

$$children_{SBnT-max}(i,s) = \begin{cases} \{(a_{n-1}a_{n-2}...\overline{a}_m...a_0)\}, \forall m \in \{0, 1, \ldots, n-1\}, & \text{if } c = 0; \\ \{q_m = (a_{n-1}a_{n-2}...\overline{a}_m...a_0)\}, & \\ \quad \forall m \in \mathcal{M}'(c,j) \text{ and } base(q_m \oplus s) = base(i \oplus s), & \text{if } c \neq 0. \end{cases}$$

$$parent_{SBnT-max}(i,s) = \begin{cases} \phi, & \text{if } c = 0; \\ (a_{n-1}a_{n-2}...\overline{a}_k...a_0), & \text{otherwise.} \end{cases}$$

Figure 7 shows the SBnT-max in a 5-cube.

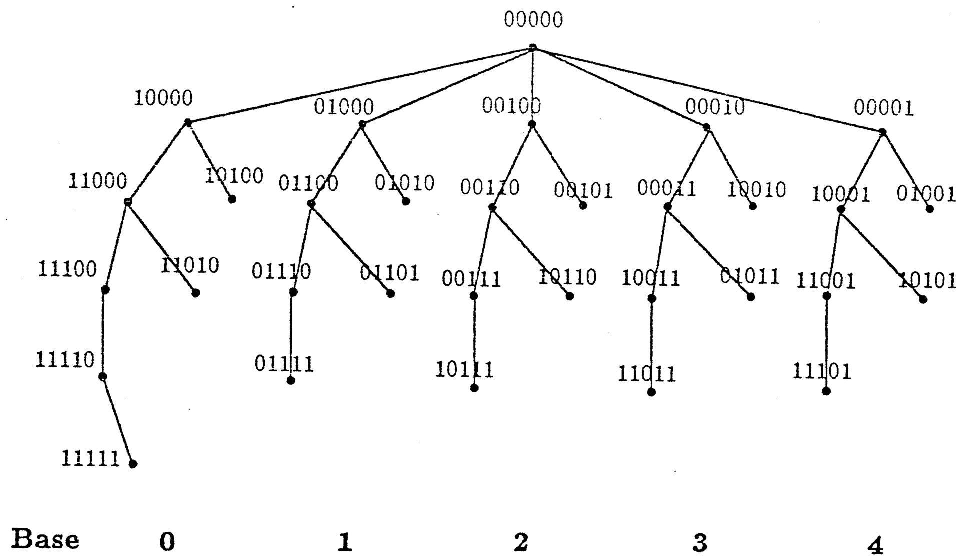



**Figure 7:** An SBnT-max in a 5-cube.

**Lemma 4.1.** *The number of nodes of each subtree of the SBnT-max is equal to the number of nodes in the corresponding subtree of the SBnT-min.*

*Proof.* For a non-cyclic node, all the $n$ nodes in the same generator set belong to $n$ different subtrees. For a cyclic node with period $P$, all the $P$ nodes in the same generator set belong to subtree 0 to subtree $P-1$ respectively.

∎

However, the topologies are not equivalent. To show that, we introduce two dual definitions of the SBnT-min and the SBnT-max. These dual definitions are interesting in their own right, and use mostly a different set of edges than the SBnT-min and SBnT-max definitions. The dual definitions are based on the bit-reversed representation of addresses. The dual of the SBnT-min is denoted SBnT-rmin and the dual of the SBnT-max is denoted SBnT-rmax. Note that the rotation and bit-reversal operations do not commute, i.e., $RB(c) \neq BR(c)$ and $LB(c) \neq BL(c)$. The SBnT-rmin is defined as follows:

Let $J_{i,s} = \{j_1, j_2, \ldots, j_m\}$, where $0 \leq j_1 < j_2 < \ldots < j_m < n$, $L^u(c) = L^v(c)$, $\quad u, v \in J_{i,s}$, and $B \cdot L^u(c) < B \cdot L^l(c)$, $\quad u \in J_{i,s}$, $\quad l \notin J_{i,s}$. Then $base(c) = j_1 = j$ and $k$ is defined by $c_k = 1$ and $c_m = 0, \forall m \in \mathcal{M}'(c,j)$ with $k = -1$ if $c = 0$.

$$children_{SBnT}(i,s) = \begin{cases} \{(a_{n-1}a_{n-2}...\overline{a}_m...a_0)\}, \forall m \in \{0, 1, \ldots, n-1\}, & \text{if } c = 0; \\ \{q_m = (a_{n-1}a_{n-2}...\overline{a}_m...a_0)\}, & \\ \quad \forall m \in \mathcal{M}'(c,j) \text{ and } base(q_m \oplus s) = base(c), & \text{if } c \neq 0. \end{cases}$$

$$parent_{SBnT}(i,s) = \begin{cases} \phi, & \text{if } c = 0; \\ (a_{n-1}a_{n-2}...\overline{a}_k...a_0), & \text{otherwise.} \end{cases}$$

Similarly, the SBnT-rmax is defined as follows:

Let $J_{i,s} = \{j_1, j_2, \ldots, j_m\}$, where $0 \leq j_1 < j_2 < \ldots j_m < n$, $R^u(c) = R^v(c)$, $\quad u, v \in J_{i,s}$, and $B \cdot R^u(c) > B \cdot R^l(c)$, $\quad u \in J_{i,s}$, $\quad l \notin J_{i,s}$. Then $base(c) = j_1 = j$ and $k$ is defined by $c_k = 1$ and $c_m = 0, \forall m \in \mathcal{M}(c,j)$ with $k = -1$ if $c = 0$.

$$children_{SBnT}(i,s) = \begin{cases} \{(a_{n-1}a_{n-2}...\overline{a}_m...a_0)\}, \forall m \in \{0, 1, \ldots, n-1\}, & \text{if } c = 0; \\ \{q_m = (a_{n-1}a_{n-2}...\overline{a}_m...a_0)\}, & \\ \quad \forall m \in \mathcal{M}(c,j) \text{ and } base(q_m \oplus s) = base(c), & \text{if } c \neq 0. \end{cases}$$

$$parent_{SBnT}(i,s) = \begin{cases} \phi, & \text{if } c = 0; \\ (a_{n-1}a_{n-2}...\overline{a}_k...a_0), & \text{otherwise.} \end{cases}$$

Semantically, $base_{min}$ is the number of right rotations yielding the longest block of leading 0-bits, and $base_{max}$ the number of left rotations yielding the longest block of leading 1-bits. Similarly, $base_{rmin}$ is the number of left rotations yielding the longest block of trailing 0-bits, and $base_{rmax}$ the number of right rotations yielding the longest block of trailing 1-bits. For example, $base_{min}$, $base_{max}$, $base_{rmin}$ and $base_{rmax}$ of node (1110100) are 2, 0, 5 and 4 respectively.

**Theorem 4.1.** *The SBnT-min and SBnT-rmin are topologically equivalent, and so are the SBnT-max and SBnT-rmax.*

*Proof.* Consider any node $i$ in subtree 0 of the SBnT-min and the node $B(i)$ in SBnT-rmin. This mapping is one-to-one and onto. We first show that every node of subtree 0 of the SBnT-min has a corresponding node in subtree 0 of the SBnT-rmin. To show this property we need to show that $BL^xB(i)$ is minimized for $x = 0$ since $i$ is in subtree 0 of the SBnT-min. But, $BL^xB(i) = BL^{x-1}LB(i) = BL^{x-1}BR(i)$ and it follows that $BL^xB(i) = R^x(i)$, which is minimized for $x = 0$. Correspondingly, every node of subtree 0 of SBnT-rmin has a counterpart in subtree 0 of the SBnT-min. The same argument applies for any other subtree and it follows that the number of nodes in every subtree of the SBnT-min is the same as the number of nodes in the same subtree of the SBnT-rmin. To complete the proof we notice that the bit-reversal operation preserves adjacency.

For the SBnT-max and SBnT-rmax case we instead use the property that $BR^xB(i) = L^x(i)$. ∎

**Theorem 4.2.** *The SBnT-max and SBnT-min are not topologically equivalent.*

*Proof.* Subtree 0 of an SBnT-max is the uniquely largest subtree. So is subtree 0 of an SBnT-max. If the SBnT-min and the SBnT-max are topologically equivalent, then the two corresponding subtrees 0 must be topologically equivalent. However, in a 6-cube the maximum fanout of any node at level 2 of an SBnT-min is 2, and the maximum fanout of any node at level 2 of an SBnT-max is 3. ∎

To explain the fact that the SBnT-max and SBnT-min are not topologically equivalent we compare the SBnT-min with the SBnT-rmax, which is topologically equivalent to SBnT-max. Due to the following lemma, the straightforward mapping does not preserve the topology.

**Lemma 4.2.** *The bit-reversed value of a bit string with a minimum value among all its rotations is not necessarily the maximum value among the bit-reversals of its rotations, i.e.,* $B \min R^x(i) \neq \max BR^x(i)$ *for some* $i$.

*Proof.* (001101) is the minimum value among all its rotations. But, (101100) is not the maximum bit-reversed value among all rotations of (001101). ∎

Note that for $n \leq 5$, the minimum value among all rotations of an address also yields the maximum bit-reversed value of its rotations. This means that the SBnT-rmax (SBnT-max) and the SBnT-min are topologically equivalent for up to 5-dimensional cubes. Figures 8, 9 and 10 show subtree 0 of an SBnT-max, SBnT-rmin and SBnT-rmax in a 6-cube. The nodes in square boxes are cyclic.

For *one-to-all personalized communication*, the SBnT-min routing has an advantage over the SBnT-max routing in that the maximum fanout is for most levels lower than for the SBnT-max routing. The fanout decreases monotonely for the SBnT-min by lemma 3.4, but this is only true for the SBnT-max for levels $l \geq 2$. Any spanning tree satisfying lemma 3.4 guarantees that the complexity of personalized communication with concurrent communication on all ports is determined by the root. The maximum fanout of nodes at a level $l$ of the SBnT-max is

$$\begin{cases} \lceil \frac{n-l}{2} \rceil, & \text{if } l = 1; \\ n - l - 1, & \text{if } 2 \leq l \leq n-2; \\ 1, & \text{if } l = n-1. \end{cases}$$

For the SBnT-min the fanout at level $l$ is $\lceil \frac{n-l}{2} \rceil$, $1 \leq i \leq n-1$, by lemma 3.3. The preference of the SBnT-min over the SBnT-rmin is due to the simpler computation of the base.

**Lemma 4.3.** *For any node below level 1 of an SBnT, the* $parent_{SBnT-min}$ *and* $parent_{SBnT-rmin}$ *functions define two distinct nodes, if the relative address has a unique longest consecutive block of zeroes, cyclically.*

*Proof.* By definition, $base_{min}$ is the dimension of the 1-bit immediately to the left of the longest block of zeroes. The parent address can be derived by complementing the 1-bit, which is immediately to the right of the longest block of zeroes. Similarly, $base_{rmin}$ is the dimension of 1-bit

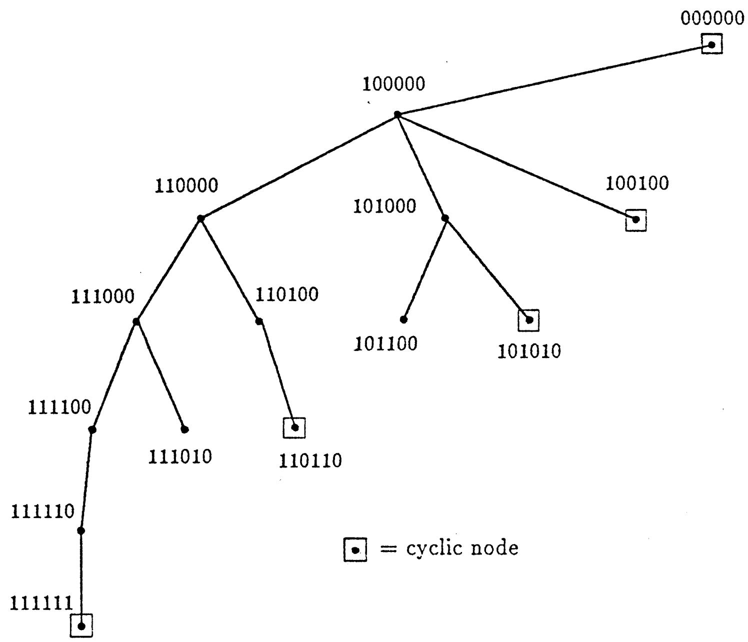



**Figure 8:** Subtree 0 of an SBnT-rmin in a 6-cube.

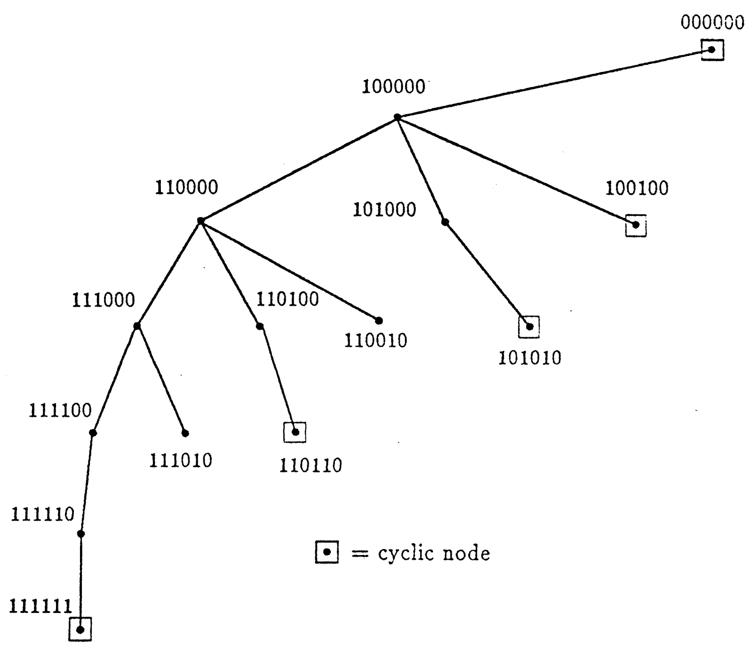



**Figure 9:** Subtree 0 of an SBnT-max in a 6-cube.

immediately to the right of the longest block of zeroes. The parent address can be derived by complementing the 1-bit, which is immediately to the left of the longest block of zeroes.

∎

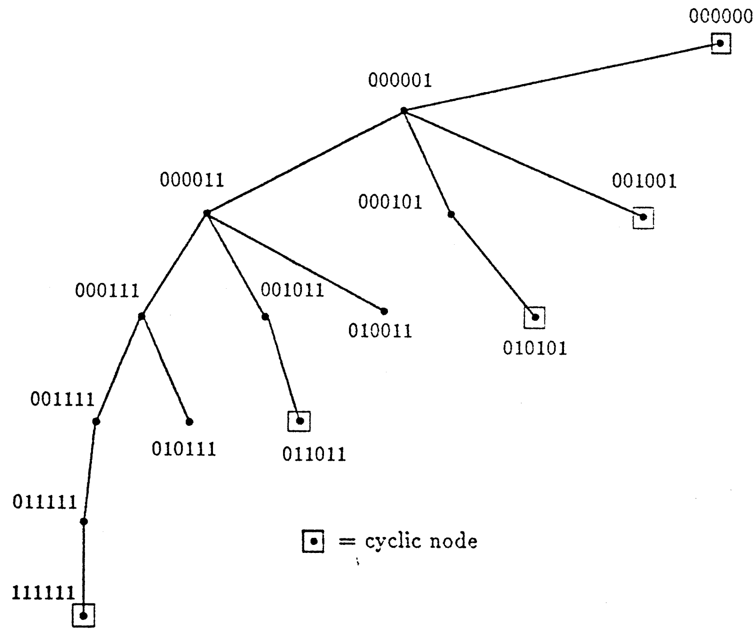



**Figure 10:** Subtree 0 of an SBnT-rmax in a 6-cube.

The $parent_{SBnT-min}$ of a cyclic node is the same as the $parent_{SBnT-rmin}$ of the same node. However, we can modify the definition of the base for SBnT-rmin to be $j_m$ instead of $j_1$, i.e., choose the maximum number of rotations instead of choosing the minimum number of rotations minimizing $BL^j(i \oplus s)$. Denote it as SBnT-rmin'. Then the $parent_{SBnT-min}$ and the $parent_{SBnT-rmin'}$ functions are distinct. Subtree $j$ of the SBnT-rmin' is topologically equivalent to subtree $n - j$ of the SBnT-rmin and the SBnT-min. It can be shown that the SBnT-min and the SBnT-rmin' are edge-disjoint below level 1 for up to 4-dimensional cubes. For 5- and 6-cubes, there are 5 and 6 common edges. For 7- and 8-cubes, there are 14 and 16 common edges. The incoming edges of nodes (01011), (010111) and (0010011) are examples. Modifications to the $parent_{SBnT}$ function, such as permutation of the dimensions, can be made to insure that the modified SBnT, and, for instance, SBnT-min, are edge-disjoint below level 1. The existence of SBnTs that are edge-disjoint below level 1 is important for fault-tolerance and will be discussed elsewhere.

## 5. Minimizing the Maximum Edge Load for Personalized Communication

The imbalance of the spanning balanced n-tree originates from the cyclic nodes. There are only $P$ distinct rotations of an address of a cyclic node with period $P$. But, there are $\frac{n}{P}$ different rotations that yields the same value. By allowing multiple parent nodes for every cyclic node, and by splitting the data set for each cyclic node into $\frac{n}{P}$ parts, the load becomes the same for each subtree of the root in one-to-all personalized communication with the same size data set for each node. The bandwidth requirement for each subtree of the SBnT is $\frac{(N-1)M}{n}$, where $M$ is the size of the data set for each node. To carry out this balancing operation, the definition of base is modified such that the base is a set of integers, called *mbase*. Let $base(i) = b$, then

$$mbase(i) = \begin{cases} \{b\}, & \text{if } i \text{ is non-cyclic;} \\ \{b + jP \mid 0 \leq j \leq \frac{n}{P} - 1\}, & \text{if } i \text{ is cyclic.} \end{cases}$$

We can view the *new* SBnT defined here as a spanning graph[9], which is composed of $n$ rotated *old* SBnT each of which has a weight $\frac{1}{n}$. The parent of a cyclic node in the $i^{th}$ SBnT is derived by choosing the dimension from the set *mbase* that is the lowest greater than $i$, cyclically.

## 6. Personalized Communication Based on an SBnT

As an example of the use of the SBnT we give some complexity results for personalized communication in a Boolean $n$-cube. We first consider the case of *one-to-all personalized communication* with the communication restricted to *one-port* at a time. With this restriction we assume that the entire data set for the subtree rooted at the sending node is communicated in one communication action. For each node we employ a *scheduling discipline* in which data is sent to subtrees in order of decreasing size. For the root there is either no difference in the data volume to the different subtrees, if multiple paths to cyclic nodes are used, or a minor difference. But within the subtrees the difference is significant.

With *one-port* communication the root requires a time of $M(N-1)t_c + n\tau$, where $M$ is the size of the data set for each node, $t_c$ the time to communicate one element of the data set, and $\tau$ the start-up time for each communication action. Then, the last subtree to receive the data has to distribute it to its nodes. With the above scheduling discipline the number of communications required for a node to receive its data, after the root of a subtree of the root receives its data, is equal to $n - 1 - \alpha$, where $\alpha$ is the number of leading zeroes of $R^j(i)$ (for the SBnT-min).

**Lemma 6.1.** *With a scheduling of one subtree at a time and subtrees in order of decreasing size a node $i$ in subtree $j$ receives its data during communication $j + n - 1 - |\mathcal{M}(i,j)|$ with the first communication being numbered* 0.

*Proof.* With the stated scheduling discipline node $i$ in subtree $j$ communicates in the dimensions $\{(n-1-|\mathcal{M}(i,j)|+j) \bmod n, (n-1-|\mathcal{M}(i,j)|+j+1) \bmod n, \ldots, (n-1+j) \bmod n\}$ in that order. Communication in the same dimension takes place concurrently. All nodes with the same value of $|\mathcal{M}(i,j)|$ receives the data during the same communication cycle.

∎

We will now give an upper bound on the time for the data transfer for *one-port* one-to-all personalized communication. The number of edges traversed by the data for node $i$ is $|i|$. We note that the arrival time of the data is an upper bound on the number of edges traversed.

**Corollary 6.1.** *The number of edges traversed by the data for node $i$ satisfies the following bound:* $|i| < n - 1 - |\mathcal{M}(i,j)|$.

The data transfer time for each subtree is bounded from above by $O(M\frac{N \log n}{n} t_c)$. To prove this bound we also need the following lemma. For proof see [11].

**Lemma 6.2.** *The number of $n$-bit binary strings for which the longest block of consecutive zeroes has length $\log n - \log \ln n - 1$ or length greater than $2 \log n$ is at most $O(\frac{N}{n})$.*

**Theorem 6.1.** *The data transfer time of each subtree of an SBnT based one-port one-to-all personalized communication is bounded from above by $O(M\frac{N \log n}{n} t_c)$.*

*Proof.* Consider subtree 0 first. Data for nodes with $|\mathcal{M}(i,j)|$ leading zeroes will traverse at most $n - 1 - |\mathcal{M}(i,j)|$ edges by corollary 6.1 counting from the root of the subtree. The total number of data element transfers in sequence is

$$\begin{aligned}
&\leq M \times \frac{1}{n} \sum_{i=0}^{n-2} (n - i - 1) \times \ \text{(number of nodes with } i \text{ leading 0's)} \\
&\leq M \times \frac{1}{n} (n \times O(\frac{N}{n}) + 2 \log n \times N) \\
&= M \times O(\frac{N \log n}{n}).
\end{aligned}$$

The second equation is derived by lemma 6.2. The first term in the parenthesis is a bound for the nodes below level $2 \log n$ and the second term is for nodes above level $2 \log n$. To complete the proof we notice that since the SBnT satisfies lemma 3.4 simply counting the number of element transfers gives a valid upper bound.

∎

A lower bound for *one-port one-to-all personalized communication* is $M(N-1)t_c + n\tau$, which is realized by routing according to a spanning binomial tree and the above scheduling discipline. For *n-port* communication the lower bound is $\frac{M(N-1)}{n} t_c + n\tau$, which is realized by SBnT routing and a *reverse breadth-first* scheduling [9]. This lower bound is not realized by a routing according to a spanning binomial tree.

The lower bound for *one-port all-to-all personalized communication* is $n(\frac{NM}{2} t_c + \tau)$, which again may be realized by a spanning binomial tree routing and an appropriate scheduling discipline. The minimum number of start-ups for the SBnT routing is approximately twice that of the binomial tree routing, but the bandwidth requirement is approximately the same. For *n-port* communication the SBnT routing again realizes the lower bound $\frac{NM}{2} t_c + n\tau$, but the binomial tree routing does not [9].

## 7. Summary

The Spanning Balanced n-Tree (SBnT) allows for scheduling disciplines that realize minimum time one-to-all personalized communication, all-to-all broadcasting and all-to-all personalized communication on a Boolean $n$-cube with *n-port* communication [9]. The number of nodes in each of the $n$ subtrees is $O(\frac{N}{n})$. The SBnT can be made to be perfectly balanced by allowing multiple parents for cyclic nodes, i.e., splitting the data sets for such nodes. The distribution of cyclic nodes is investigated in some detail. A few different definitions of *Spanning Balanced n-trees* are proposed and compared. They are of particular interest with respect to fault-tolerant communications. Single edge failure, with the exception of the edges from the root, and several forms of multiple edge failures can be routed around, given that the failure is known and the proper SBnT is chosen.


## Acknowledgement

This work has in part been supported by the Office of Naval Research under contracts N00014-84-K-0043 and N00014-86-K-0564.


## References


[1] Aho A.V., Hopcroft J.E., Ullman J.D., The Design and Analysis of Computer Algorithms, Addison-Wesley, 1974, pp. 164–165.

[2] Bhatt S.N.,Chung F.R.K., Leighton F.T., Rosenberg A.L., *Optimal Embeddings of Binary Trees in the Boolean Hypercube,* Technical Report YALEU/CSD/RR-, Yale University, Dept. of Computer Science, December 1985.

[3] Bhatt S.N, Ipsen I.C.F., *How to Embed Trees in Hypercubes,* Technical Report YALEU/CSD/RR-443, Yale University, Dept. of Computer Science, December 1985.

[4] Deshpande S.R., Jenevin R.M., *Scaleability of a Binary Tree on a Hypercube,* Technical Report TR-86-01, University of Texas at Austin, January 1986.

[5] Fischer M.J., Efficiency of Equivalence Algorithms, *Complexity of Computer Computations,* Plenum Press, 1972, pp. 153–167.

[6] Ho C.-T., Johnsson S.L., Distributed Routing Algorithms for Broadcasting and Personalized Communication in Hypercubes., *1986 Int. Conf. Parallel Processing,* IEEE Computer Society, 1986, pp. 640–648. Tech. report YALEU/CSD/RR-483.

[7] ————, *Matrix Transposition on Boolean n-cube Configured Ensemble Architectures,* Technical Report YALEU/CSD/RR-494, Yale University, Dept. of Computer Science, September 1986.

[8] Hoey D., Leiserson C.E., A layout for the shuffle-exchange network, *1980 Int. Conf. Parallel Processing,* IEEE Computer Society, 1980.

[9] Johnsson S.L., Ho C.-T., *Spanning Graphs for Optimum Broadcasting and Personalized Communication in Hypercubes,* Technical Report YALEU/CSD/RR-500, Yale University, Dept. of Computer Science, November 1986.

[10] Johnsson S.L., *Communication Efficient Basic Linear Algebra Computations on Hypercube Architectures,* Journal of Parallel and Distributed Computing, (1986). Report YALEU/CSD/RR-361, January 1985, Dept. of Computer Science, Yale University.

[11] Leighton F.T., *Complexity Issues in VLSI: Optimal Layouts for the Shuffle-Exchange Graph and Other Networks,* MIT Press, 1983.

[12] Seitz C.L., Ensemble Architectures for VLSI – A Survey and Taxonomy, P. Penfield Jr. ed., *1982 Conf on Advanced Research in VLSI,* Artech House, January 1982, pp. 130-135.

# The Yale Journal of Biology and Medicine

## INFORMATION FOR AUTHORS

*The Yale Journal of Biology and Medicine* is an international journal which publishes original contributions, reviews, historical articles, case reports, and symposia related to biomedical matters. It welcomes contributions from all qualified sources. Manuscripts are accepted on the understanding that they have not been, nor will be, submitted for publication elsewhere.

*Submission of Manuscripts:* Manuscripts should be submitted to:

The Managing Editor, The Yale Journal of Biology and Medicine,
333 Cedar Street, New Haven, Connecticut 06510

*Form of Manuscript:* Three copies including figures and title pages are required. Manuscripts must be typewritten and double-spaced throughout, in black ink on one side of 8½ by 11 in. white paper, with minimum margins of 1 inch on all sides. All pages should be numbered consecutively. The title page should contain the article title, author(s)' name(s) and academic degree(s), affiliation(s), a short abbreviated running title of 40 to 50 characters including letters and spaces, and the name and complete mailing address of the person to whom correspondence and reprint requests should be sent. An abstract of less than 200 words must accompany each manuscript. Text should begin on page 1. Manuscripts prepared on word processors should not be justified.

*Editorial style* should conform to *The ACS Style Guide* or the *CBE Style Manual.*

*Abbreviations* used in the manuscript should be listed double-spaced, in alphabetical order, with their definitions, on a separate sheet of paper headed "Abbreviations."

*Footnotes* should be identified by superior Arabic numerals in the text and should be typed double-spaced on a separate sheet headed "Footnotes" at the end of the article. Footnotes to tables should be identified by superior lower case italic letters ($^{a,b}$ etc.) and typed, double-spaced, at the bottom of the table.

*Tables* should be consecutively numbered with Arabic numerals and so cited in the text. Each table, with its number and title, should be typed double-spaced on a separate sheet. Units of measurements should be clearly indicated for each entry of the table, in the column head if possible or else as a part of the entry.

*Figures* should be cited consecutively in the text by Arabic numerals. Figure captions should be typed double-spaced on a separate sheet. All figures must be submitted in duplicate as high-quality glossy prints which will permit photographic reproduction without retouching or redrawing. The lettering should be large enough to be legible after a reduction of 50%. Figures must *not* be mounted on cardboard or other backing. Each figure must be identified by its figure number and the top indicated on the back of the art.

*References* should be cited *numerically,* in order in the text, using an Arabic numeral in brackets. They must be typed *double-spaced,* in numerical order, on a separate sheet headed "References" at the end of the paper. References appearing for the first time in tables and figures must be numbered in sequence with those cited in the text where the table or figure is mentioned. The style of references is that of the *Index Medicus.* When authorship of a reference is multiple, all names will be cited.

Sample references:

1. Pawelek J, Wong G, Sansone M, Morowitz J: Molecular controls in mammalian pigmentation. Yale J Biol Med 46:430–443, 1973
2. Summers WC: Molecular epidemiology of DNA viruses: Applications of restriction endonuclease cleavage site analysis. In Symposium on Recent Advances in Clinical Virology. Edited by GD Hsiung. New York, Praeger Publishers, 1981, pp 55–60
3. Squire J: The Structural Basis of Muscular Contraction. New York, Plenum Press, 1981, 698 pp

*Special symbols* which cannot be typed should be carefully drawn in in black ink and identified, in words, on the margin of the manuscript page. Such symbols might include special characters, accents, or mathematical notation.

*Proofs:* Galley proofs will be sent to the author, with a reprint order form, a note on the preferred marking of proofs, and a request for the key words under which the article should be indexed for the *Journal.*

*Copyright:* All material accepted for publication by *The Yale Journal of Biology and Medicine* becomes the property of the *Journal* which owns its copyright. It is understood that submission includes the right of further reproduction at the *Journal's* discretion.