Yale University
Department of Computer Science

Optimum Broadcasting and Personalized Communication in Hypercubes

S. Lennart Johnsson and Ching-Tien Ho

YALEU/DCS/TR-610
December 1987

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

† A revised edition of TR-500. To appear in *IEEE Transactions on Computers*.

Optimum Broadcasting and Personalized Communication in Hypercubes†

S. Lennart Johnsson and Ching-Tien Ho
Departments of Computer Science
Yale University
New Haven, CT 06520

Abstract. Effective utilization of communication resources is crucial for good overall performance in highly concurrent systems. In this paper we address four different communication problems in Boolean n-cube configured multiprocessors: (1) one-to-all broadcasting: distribution of common data from a single source to all other nodes; (2) one-to-all personalized communication: a single node sending unique data to all other nodes; (3) all-to-all broadcasting: distribution of common data from each node to all other nodes; and (4) all-to-all personalized communication: each node sending a unique piece of information to every other node. Three new communication graphs for the Boolean n-cube are proposed for the routing, and scheduling disciplines provably optimum within a small constant factor proposed. One of the new communication graphs consists of n edge-disjoint spanning binomial trees, and offers optimal communication for case 1; a speed-up with a factor of n over the spanning binomial tree for large data volumes. The other two new communication graphs are a balanced spanning tree, and a graph composed of n rotated spanning binomial trees. With appropriate scheduling and concurrent communication on all ports of every processor, routings based on these two communication graphs offer a speed-up of up to $\frac{n}{2}$, $\frac{n}{2}$ and $O(\sqrt{n})$ over the routings based on the spanning binomial tree for cases 2, 3 and 4 respectively. All three new spanning graphs offer optimal communication times for cases 2, 3 and 4 and concurrent communication on all ports of every processor. The graph consisting of n edge-disjoint spanning trees offers graceful degradation of performance under faulty conditions. Timing models and complexity analysis have been verified by experiments on a Boolean cube configured multiprocessor.

1 Introduction

In this paper we investigate *broadcasting* and *personalized communication* on Boolean n-cube configured ensemble architectures. In *broadcasting,* a data set is copied from one node to all other nodes, or a subset thereof. In *personalized communication*, a node sends a unique data set to all other nodes, or a subset thereof. We consider broadcasting from a single source to all other nodes, *one-to-all broadcasting*, and concurrent broadcasting from all nodes to all other nodes, or *all-to-all broadcasting*. Broadcasting is used in a variety of linear algebra algorithms [10,17,19] such as matrix-vector multiplication, matrix-matrix multiplication, LU-factorization, and Householder transformations. It is also used in data base queries and transitive closure

† A short version of this paper received the "Outstanding Paper Award" at the 1986 International Conference on Parallel Processing.

algorithms [4]. The reverse of the broadcasting operation is *reduction*, in which the data set is reduced by applying operators such as addition/subtraction or max/min.

For personalized communication we consider *one-to-all personalized communication* and *all-to-all personalized communication*. Fundamentally, the difference between broadcasting and personalized communication is that in the latter no replication/reduction of data takes place. The bandwidth requirement is highest at the root and is reduced monotonically towards the leaves. Personalized communication is used, for instance, in transposing a matrix, and the conversion between different data structures [17,22]. Matrix transposition is useful in the solution of tridiagonal systems on Boolean cubes for certain combinations of machine characteristics [20,23], and for matrix-vector and matrix-matrix [21] multiplications.

For single source broadcasting and personalized communication a *one-to-all communication graph* is required. Graphs of minimum height have minimum propagation time which is the overriding concern for small data volumes, or a high overhead for each communication action. For large data volumes it is important to use the bandwidth of a Boolean cube effectively, in particular, if each processor is able to communicate on all its ports concurrently. We propose three new spanning graphs for Boolean n-cubes of $N = 2^n$ nodes: one consisting of n edge-disjoint binomial trees (nESBT); one that consists of n rotated spanning binomial trees (nRSBT); and one *balanced tree* (SBnT), i.e., a tree with fanout n at the root and approximately $\frac{N}{n}$ nodes in each subtree. We prove some of the critical topological properties of the new one-to-all communication graphs, and compare them with Hamiltonian paths and binomial tree embeddings. For each of the communications we consider, we prove the lower bounds in Table 1.

Comm. model	Communication task	Lower bound
one-port	One-to-all broadcasting	$\max((M+n-1)t_c, n\tau)$
	One-to-all personalized comm.	$\max((N-1)Mt_c, n\tau)$
	All-to-all broadcasting	$\max((N-1)Mt_c, n\tau)$
	All-to-all personalized comm.	$\max(\frac{nNM}{2}t_c, n\tau)$
n-port	One-to-all broadcasting	$\max((\frac{M}{n}+n-1)t_c, n\tau)$
	One-to-all personalized comm.	$\max(\frac{(N-1)M}{n}t_c, n\tau)$
	All-to-all broadcasting	$\max(\frac{(N-1)M}{n}t_c, n\tau)$
	All-to-all personalized comm.	$\max(\frac{NM}{2}t_c, n\tau)$

Table 1: Lower bounds for some Boolean cube communications.

We generalize the one-to-all communications to all-to-all communications and study the interleaving of communications from different sources by defining *all-to-all communication graphs* as the union of *one-to-all communication graphs*. We define scheduling disciplines for the four different communications as follows: (1) *one-to-all broadcasting*; (2) *personalized communication*; (3) *all-to-all broadcasting*; and (4) *personalized communication*. We show that for communication restricted to one port at-a-time, our spanning binomial tree scheduling results in communication times within a factor of two of the best known lower bounds for communications 2, 3, and 4. For case 1, the scheduling we define for the n edge-disjoint spanning binomial trees completes within a factor of four of the best known lower bound, also for concurrent communication on all ports. For concurrent communication and *one-to-all personalized communication* the schedules for both the n edge-disjoint and the rotated binomial trees are of optimal order, as are the schedules for the balanced graph. These graphs also yield *all-to-all* communication within a factor of two of the lower bound for arbitrary cube sizes and data volumes, except for the nESBT graph which allows optimum scheduling within a factor of two only asymptotically, Table 16.

Communication in Boolean cubes has recently received significant interest due to the success of the Caltech Cosmic Cube project [30] and commercially available Boolean cube configured concurrent processors (from Intel, NCUBE[12], and Ametek, and cube-like architectures from Floating-Point Systems[11] and Thinking Machines Corp. [13]). Embedding of complete binary trees is treated in [33,17,29,6,3]. Wu [33] also discusses the embedding of k-ary trees. Embedding of arbitrary binary trees is discussed in [3] and improved in [2]. Efficient routing using randomization for arbitrary permutations has been suggested by Valiant et al. [32]. Our algorithms attain a speed-up of up to a factor of n for case 1 with *one-port* and *n-port* communication, and cases 2 and 4 with *n-port* communication over the best previously known algorithms [28]. For case 3 and *n-port* communication, the improvement is by a constant factor. Communication on hypercubes have also been studied independently by Fox et al. [8] and Stout et al. [31]. For case 1, the best algorithm in [8] is about a factor of two slower than the best algorithm presented here and in [14]. The best algorithm in [31] requires one less routing cycle. The nRSBT routing [24] for cases 2, 3 and 4 with *n-port* communication was also discovered independently by Stout. However, the SBnT routing [14] described here performs better, or as well as the nRSBT routing. The nESBT routing degrades gracefully under faulty conditions. The analysis is compared with experimental data.

The outline of the paper is as follows. Notations, definitions, and general graph properties used throughout the paper are introduced in section 2. In section 3, one-to-all communication graphs are defined and characterized. Scheduling disciplines and associated complexity estimates are given in section 4 for *one-to-all broadcasting*; in section 5 for *one-to-all personalized communication*; in section 6 for *all-to-all broadcasting*; and in section 7 for *all-to-all personalized communication*. Section 8 presents implementation results for some of the communication algorithms on the Intel iPSC. Conclusion follows in section 9.

2 Preliminaries

In the following, node i has address $(i_{n-1}i_{n-2}\ldots i_0)$, and node s has address $(s_{n-1}s_{n-2}\ldots s_0)$. The bit-wise *exclusive-or* operation is denoted $\oplus$, and throughout the paper $i \oplus s = c = (c_{n-1}c_{n-2}\ldots c_0)$ where $c_m = i_m \oplus s_m$. c is the *relative address* of node i with respect to node s. Address bits are numbered from 0 through $n-1$ with the lowest order bit being the 0^{th} bit. The m^{th} bit corresponds to the m^{th} dimension in a Boolean space. Caligraphic letters are used for sets. The set of *node* addresses is $\mathcal{N} \equiv \{0, 1, \ldots, N-1\}$, and the set of dimensions is $\mathcal{D} \equiv \{0, 1, \ldots, n-1\}$. $|\mathcal{S}|$ is used to denote the cardinality of a set $\mathcal{S}$. d or m is used to denote an arbitrary dimension, $d, m \in \mathcal{D}$.

Definition 1 The *Hamming* distance between a pair of binary numbers, or nodes, i and j is $Hamming(i,j) = \sum_{m=0}^{n-1}(i_m \oplus j_m)$.

Definition 2 A *Boolean n-cube* is a graph $B = (\mathcal{V}, \mathcal{E}^B)$ such that $\mathcal{V} = \mathcal{N}$ and $\mathcal{E}^B = \{(i,j) | i \oplus j = 2^m, \forall m \in \mathcal{D}, \forall i,j \in \mathcal{N}\}$. An edge (i,j) such that $i \oplus j = 2^m$ is in dimension m, and nodes i and j are connected through the m^{th} *port*.

An *edge* (i,j) is *directed* from node i to node j, and of unit length, i.e., $Hamming(i,j) = 1$. It is a $0 \rightarrow 1$ edge if the bit that differs in i and j is zero in i; otherwise, it is a $1 \rightarrow 0$ edge. In a directed graph all edges are directed. A node with no edges directed to it is a *root* (*source*) node, and a node with no edges directed away from it is a *leaf* (*sink*) node. A node that is neither a leaf node nor a root node is an *internal* node. If (i,j) is a directed edge, then node i is the *parent* of node j, and j is the *child* of i.

Lemma 1 [18,29] A Boolean n-cube has $N = 2^n$ nodes, diameter n, $\binom{n}{x}$ nodes at Hamming distance x from a given node, and n edge-disjoint paths between any pair of nodes, i and j. Of these n paths, $Hamming(i,j)$ are of length $Hamming(i,j)$ and $n - Hamming(i,j)$ paths are of length $Hamming(i,j)+2$. Every node has n edges directed to it, and n edges directed away from it. The total number of (directed) communication links is nN.

Definition 3 With *Rotation of a node* we mean a right cyclic rotation of its address, $Ro(i) = (i_0 i_{n-1} \ldots i_2 i_1)$. With *Rotation of a graph* $G(\mathcal{V},\mathcal{E})$ we mean a graph $Ro(G(\mathcal{V},\mathcal{E})) = G(Ro(\mathcal{V}), Ro(\mathcal{E}))$, where $Ro(\mathcal{V}) = \{Ro(i)|\forall i \in \mathcal{V}\}$ and $Ro(\mathcal{E}) = \{(Ro(i), Ro(j))|\forall (i,j) \in \mathcal{E}\}$. Moreover, Ro^{-1} is a left rotation and $Ro^k = Ro \circ Ro^{k-1}$ for all k.

The right rotation operation is also known as an *unshuffle* operation and the left rotation as a *shuffle* operation.

Definition 4 With *Reflection of a node* we mean a bit-reversal of its address, $Re(i) = (i_0 i_1 \ldots i_{n-2} i_{n-1})$. With *Reflection of a graph* $G(\mathcal{V},\mathcal{E})$ we mean a graph $Re(G(\mathcal{V},\mathcal{E})) = G(Re(\mathcal{V}), Re(\mathcal{E}))$, where $Re(\mathcal{V}) = \{Re(i)|\forall i \in \mathcal{V}\}$ and $Re(\mathcal{E}) = \{(Re(i), Re(j))|\forall (i,j) \in \mathcal{E}\}$.

Definition 5 With *Translation of a node* i by s we mean a bit-wise exclusive-or of the addresses, $Tr(s,i) = c$. With *Translation of a graph* $G(\mathcal{V},\mathcal{E})$ with respect to node s, we mean a graph $Tr(s, G(\mathcal{V},\mathcal{E})) = G(Tr(s,\mathcal{V}), Tr(s,\mathcal{E}))$, where $Tr(s,\mathcal{V}) = \{Tr(s,i)|\forall i \in \mathcal{V}\}$ and $Tr(s,\mathcal{E}) = \{(Tr(s,i), Tr(s,j))|\forall (i,j) \in \mathcal{E}\}$.

Lemma 2 *Rotations, Reflections*, and *Translations* of a graph preserves Hamming distance between nodes. The Rotation operation Ro^k maps every edge in dimension d to dimension $(d-k)$ mod n, the *Reflection* operation maps every edge in dimension d to dimension $n-1-d$, and the *Translation* operation preserves the dimension of every edge. *Rotation* and *Reflection* preserves the direction of every edge. *Translation* reverses the direction of all edges in the dimensions for which $s_m = 1$, $m \in \mathcal{D}$.

Corollary 1 The topology of a graph remains unchanged under *Rotation*, *Reflection*, and *Translation*.

Definition 6 For a binary number i the *Period* of i, $P(i) = \min_{m>0} Ro^m(i) = i$. A binary number i is *cyclic* if $P(i) < n$; and *non-cyclic*, otherwise. A *cyclic node* is a node with cyclic *relative* address.

The period of the number (011011) is 3. A cyclic node is defined only when the source node is given. Node (001000) is cyclic with respect to the source node (000001).

Definition 7 A *Spanning Tree* $T^{id}(s)$ rooted at node s of a Boolean cube is a tree containing all the nodes of the Boolean cube. id is used to identify different spanning trees. $T^{id}(s) = Tr(s, T^{id}(0))$.

Definition 8 For the Boolean n-cube a *one-to-all communication graph*, *o-graph*, with source node s is a connected, directed graph $G^{id}(s) = Tr(s, G^{id}(0))$ where $G^{id}(0)$ is either a spanning tree, $G^{id}(0) = T^{id}(0)$, or a composition of n distinctly rotated spanning trees, $G^{id}(0) = \cup_{d\in\mathcal{D}} Ro^d(T^{id'}(0))$. The weight of every edge in an *o-graph* is 1 if the graph is a tree; and $\frac{1}{n}$, otherwise. id' identifies the generating spanning tree for the *o-graph*.

The weight $\frac{1}{n}$ is introduced to account for the data to be communicated being split into n pieces for an *o-graph* composed of n spanning trees. There exist n paths from the root to any other node. If no two edges of $T^{id_x}(s)$ and $T^{id_y}(s)$ are mapped to the same cube edge $\forall x, y \in \mathcal{D}$, $x \neq y$, then the paths of the *o-graph* are *edge-disjoint*, and there is no contention problem.

The root of a spanning tree has *level* 0. The node i in a spanning tree has a level which is one more than the level of its parent. The height h of a tree is the largest level of all the nodes.

Definition 9 For an *o-graph* G^{id}, the total weight of the edges in dimension d between levels l and $l+1$ is denoted $e^{id}(d,l)$, and the total weight of edges in dimension d is $E^{id}(d)$.

Lemma 3 Given an *o-graph* $G^{id}(s) = Tr(s, G^{id}(0))$ where $G^{id}(0) = \cup_{d \in \mathcal{D}} Ro^d(T^{id'}(0))$, then, $E^{id}(d) = \frac{N-1}{n}$ and $e^{id}(d,l) = \frac{1}{n}\sum_{d \in \mathcal{D}} e^{id'}(d,l)$.

Proof: The lemma follows from definition 9 and the rotation property in lemma 2. ∎

We define three basic spanning trees: a *Hamiltonian* path, T^H, a *Spanning Binomial Tree*, T^{SBT}, and a *Spanning Balanced n-Tree*, T^{SBnT}. These trees are used to form composition graphs that provide multiple paths (not necessarily edge-disjoint) to all other nodes.

For the definition/implementation of *o-graphs* we use distributed algorithms that for any node computes the addresses of its set of children nodes, if any, and the address of the parent node, except for the root node. A node has one parent and a set of children nodes for each spanning tree used for the composition.

Definition 10 The $children^{id}(i,s,k)$ function generates the set of children addresses of node i in the k^{th} spanning tree of $G^{id}(s)$. The $parent^{id}(i,s,k)$ function generates the address of the parent of node i in the k^{th} spanning tree of $G^{id}(s)$. For the G^{id} being a spanning tree we omit the last parameter.

Definition 11 A *greedy spanning tree* rooted at node s of a Boolean n-cube is a spanning tree such that $\forall i \in \mathcal{N}$, the level of node i is $Hamming(i,s)$. A *greedy o-graph* of a Boolean n-cube is a composition of greedy spanning trees.

Lemma 4 A greedy *o-graph* contains only $0 \rightarrow 1$ edges "relative" to the root.

Lemma 5 A spanning tree is greedy iff $|\{i | Hamming(i,s) = l\}| = \binom{n}{l}$. A greedy *o-graph* is acyclic and of minimal height.

Corollary 2 If $G^{id}(0) = \cup_{d \in \mathcal{D}} Ro^d(T^{id}(0))$, and T^{id} is greedy, then $e^{id}(d,l) = \frac{1}{n}\binom{n}{l+1}$.

Lemma 5 follows from definition 11 and lemma 1. Corollary 2 follows from lemmas 3 and 5.

Note that an *o-graph* of minimum height is not necessarily a greedy *o-graph*. In all-to-all personalized communication only greedy *o-graphs* have scheduling disciplines that accomplishes minimal data transfer time.

Definition 12 In *one-port* communication a processor can only send *and* receive on one of its ports at any given time. The port on which a processor sends and receives can be different. In *n-port* communication a processor can communicate on all its ports concurrently.

We also assume that there is an overhead, start-up time τ, associated with each communication of B elements, each of which require a transfer time t_c. B_{opt} denotes an optimal packet size. For the analysis it is convenient to assume that communication takes place during distinct time intervals. The duration of a *Routing Cycle* is $\tau + Bt_c$. Routing cycles are labeled from 0.

For *Broadcasting* the data is replicated $|children^{id}(i,s,k)|$ times in node i for spanning tree k of the *o-graph* $G^{id}(s)$. With *n-port* communication, and negligible time for replication, all ports are scheduled concurrently. With *one-port* communication the order of communications on different ports is important.

In *Personalized Communication* the source node sends a unique message to every other node. An internal node needs to receive and forward all the data for every node of the subtree of which it is a root. The ordering of data for a port is important for the communication time both for *one-port* and *n-port* communication.

The *Scheduling Discipline* defines the communication order for each port, and the order between ports for every non-leaf node. We assume the same data independent scheduling discipline for every node. The scheduling disciplines are completely specified later.

Definition 13 In a *reverse-breadth-first* scheduling discipline for *one-to-all personalized communication* based on an *o-graph* of height h, the root sends out the data for the nodes at level $h-p$ during the p^{th} cycle, $0 \leq p < h$. The data received by an internal node is propagated to the next level during the next cycle, if the data is not destined for the node itself. In a *postorder* [1] scheduling discipline each node sends out the entire data set to each of its children nodes before accepting its own data.

The analysis of the complexity of the communication algorithms is considerably simplified for *n-port* communication, if only the subtree with the maximum number of nodes need to be considered. The following lemma guarantees that this is the case, if the subtree with the maximum number of nodes also has the maximum height.

Lemma 6 Given a spanning tree, let $\phi^{id}(i,x)$ be the number of nodes at distance x from node i in the subtree rooted at node i. If $\phi^{id}(i,x) \geq \phi^{id}(j,x)$ for any child node j of node i, then the data transfer time of *n-port* one-to-all personalized communication based on *reverse-breadth-first* ordering is dominated by the data transfer over the edges from the root.

Proof: The lemma follows from the fact that with *reverse-breadth-first* scheduling, the propagation time for the internal nodes is at most the same as the transmission time for the root during each routing cycle. ∎

Definition 14 An *all-to-all communication graph*, *a-graph*, $G^{id}(*) = \cup_{s \in \mathcal{N}} G^{id}(s)$.

The quantities $v^{id}(d,l)$ and $u^{id}(d,l)$ defined next are useful in deriving the time complexity of *all-to-all personalized communication* for the *a-graph*.

Definition 15 Define $v^{id}(d,l)$ of an *o-graph* as the total weight of all edges within all subtrees rooted at level $l+1$ of all spanning trees with subtree roots connected to a parent node through an edge in dimension d, inclusive of the edge to the parent node. Let $\mathcal{S}_d^{id} = \{j | j \in \mathcal{V}, (i,j) \in \mathcal{E}^{id} \text{ and } i \oplus j = 2^d\}$. Define $u^{id}(d,l)$ to be the total weight of edges terminating on all nodes k such that k is a descendent of j at distance l, $\forall j \in \mathcal{S}_d^{id}$.

Lemma 7 $v^{id}(d,l) = u^{id}(d,l) = \sum_{x=l}^{h-1} e^{id}(d,x) = \frac{1}{n} \sum_{x=l}^{h-1} \sum_{d \in \mathcal{D}} e^{id'}(d,x)$, $\forall l \in [0, h-1]$, $d \in \mathcal{D}$ for an *o-graph* $G^{id}(0) = \cup_{d \in \mathcal{D}} Ro^d(T^{id'}(0))$, where h is the height of $T^{id'}$.

Proof: From the n distinct rotations and lemma 2, it follows that $v^{id}(d,l) = \frac{1}{n}\times$ (the sum of the number of nodes at levels x, $l+1 \le x \le h$) of $T^{id'}$. Similarly, $u^{id}(d,l) = \frac{1}{n}\times$ (the sum of the number of nodes at levels x, $l+1 \le x \le h$) of $T^{id'}$. By lemma 3, $e^{id}(d,l) = \frac{1}{n}\times$ (the number of edges between levels l and $l+1$) of $T^{id'} = \frac{1}{n}\sum_{d\in\mathcal{D}} e^{id'}(d,l)$. ∎

Corollary 3 If an *o-graph* $G^{id}(0) = \cup_{d\in\mathcal{D}} Ro^d(T^{id'}(0))$ and $T^{id'}$ is greedy, then $v^{id}(d,l) = u^{id}(d,l) = \frac{1}{n}\sum_{i=l+1}^{n}\binom{n}{i}$.

Proof: The corollary follows from lemma 7 and corollary 2. ∎

Lemma 8 For an *a-graph* $G^{id}(*)$, the total weight of communication graph edges mapped to every cube edge in dimension d is $E^{id}(d)$. The total weight of communication graph edges between levels l and $l+1$ in dimension d is $e^{id}(d,l)$.

Proof: Since $G^{id}(*) = \cup_{s\in\mathcal{N}} G^{id}(s) = \cup_{s\in\mathcal{N}} Tr(s, G^{id}(0))$, every *o-graph* edge is mapped to a distinct cube edge in the same dimension through N distinct exclusive-or operations. Hence, the total weight of *a-graph* edges mapped to every cube edge in dimension d is $E^{id}(d)$. The bit-wise exclusive-or operation preserves the topology of a spanning tree, and hence the number of edges at a given distance from the source node. ∎

3 Spanning Graphs

3.1 Three Spanning Trees

3.1.1 A Hamiltonian Path

A Hamiltonian path H originating at node 0 is defined by traversing the nodes of the Boolean cube in a *binary-reflected* Gray code [27] order with starting address equal to 0. Let the n-bit code of 2^n integers be $Gray(n)$. Two definitions of the code that are convenient to use are the following.

Definition 16 The *binary-reflected Gray code* on n bits is defined recursively as follows. Let $Gray(n) = (\hat{G}_0, \hat{G}_1, \ldots, \hat{G}_{2^n-2}, \hat{G}_{2^n-1})$.

$$\text{Then} \quad Gray(n+1) = (0\hat{G}_0, 0\hat{G}_1, \ldots, 0\hat{G}_{2^n-2}, 0\hat{G}_{2^n-1}, 1\hat{G}_{2^n-1}, 1\hat{G}_{2^n-2}, \ldots, 1\hat{G}_1, 1\hat{G}_0),$$

$$\text{or alternatively,} \quad Gray(n+1) = (\hat{G}_0 0, \hat{G}_0 1, \hat{G}_1 1, \hat{G}_1 0, \hat{G}_2 0, \hat{G}_2 1, \ldots, \hat{G}_{2^n-1} 1, \hat{G}_{2^n-1} 0).$$

The following alternative definition is also useful for distributed routing algorithms. Let $T(n) = (t_0, t_1, \ldots, t_{2^n-2})$ be the sequence of dimensions on which a transition takes place in proceeding from integer 0 to integer $2^n - 1$ in the n-bit Gray code with the most significant bit labeled $n-1$.

Definition 17 Then the *binary-reflected Gray code* can be defined through the recursion

$$T(n) = (T(n-1), n-1, T(n-1)), \qquad T(1) = 0.$$

Note that t_i is also the lowest order dimension with a 0-bit in the binary encoding of i. Let the binary encoding of $i = (i_{n-1} i_{n-2} \ldots i_0)$ and the Gray code encoding be $\hat{G}_i = (g_{n-1} g_{n-2} \ldots g_0)$. Then the conversions between binary and Gray code encoding are defined by

$$g_m = i_m \oplus i_{m+1}, \quad \text{and conversely} \quad i_m = g_{m+1} \oplus g_{m+2} \oplus \cdots \oplus g_{n-1}.$$

3.1.2 A Spanning Binomial Tree

A 0-level binomial tree has 1 node. An n-level binomial tree is constructed out of two $(n-1)$-level binomial trees by adding one edge between the roots of the two trees, and by making either root the new root, [1,7]. The familiar spanning tree rooted in node 0 of a Boolean n-cube generated by complementing leading zeroes of the binary encoding of a processor address i [9,17,26,28,29] is indeed a *Spanning Binomial Tree* (SBT).

Definition 18 The spanning binomial tree rooted at node s, $T^{SBT}(s)$, is defined as follows. Let p be such that $c_p = 1$ and $c_m = 0, \forall m \in \{p+1, p+2, \ldots, n-1\} \equiv \mathcal{M}^{SBT}(c)$ and let $p = -1$ if $c = 0$. The set $\mathcal{M}^{SBT}(c)$ is the set of leading zeroes of c. Then,

$$children^{SBT}(i,s) = \{(i_{n-1}i_{n-2}\ldots\bar{i}_m\ldots i_0)\}, \forall m \in \mathcal{M}^{SBT}(c),$$

$$parent^{SBT}(i,s) = \begin{cases} \phi, & i = s; \\ (i_{n-1}i_{n-2}\ldots\bar{i}_p\ldots i_0), & i \neq s. \end{cases}$$

It is easy to verify that the parent and children functions are consistent, i.e., that node j is a child of node i iff node i is the parent of node j. Figure 1 shows the $T^{SBT}(0)$ for a 4-cube.

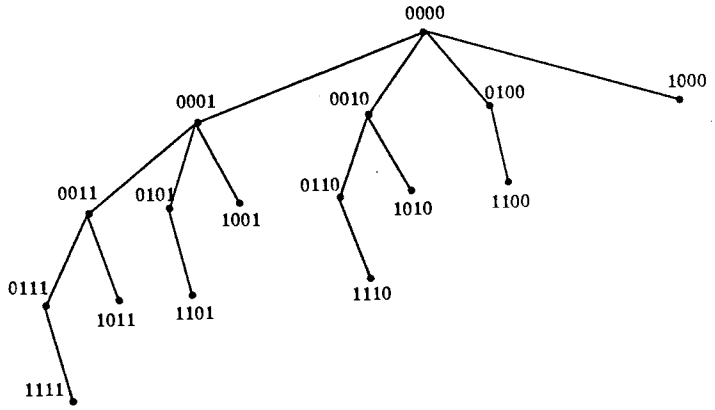

Figure 1: A spanning binomial tree in a 4-cube.

Definition 19 Subtree k of the $T^{SBT}(s)$ consists of all nodes such that $c_k = 1$ and $c_m = 0, \forall m \in \{0, 1, \ldots, k-1\}$.

Lemma 9 There are 2^{n-k-1} nodes in subtree k, and the maximum degree of any node at level l in subtree k is $n-k-l$, $0 < l \leq n-k$.

Lemma 10 Let $\phi^{SBT}(i,s,x)$ be the number of nodes at distance x from node i in the subtree rooted at node i of the SBT rooted at node s. Then, $\phi^{SBT}(i,s,x) \geq \phi^{SBT}(j,s,x)$, if $j \in children^{SBT}(i,s)$.

Proof: From the definition of the SBT, the subtree rooted at node j is a connected subgraph of the subtree rooted at node i. ∎

3.1.3 A Spanning Balanced n-Tree, and a Spanning Balanced Graph

In the *Spanning Balanced n-Tree* [15] the node set is divided into n sets of nodes with approximately an equal number of nodes. Each such set forms a subtree of the source node. The maximum number of elements that need to traverse any edge directed away from the source node is minimized for personalized communication.

Definition 20 Let $K(i,s) = \{k_1, k_2, \ldots, k_m | 0 \leq k_1 < k_2 < \cdots < k_m < n\}$, be such that $Ro^{\alpha}(c) = Ro^{\beta}(c)$, $\forall \alpha, \beta \in K(i,s)$, and $Ro^{\alpha}(c) < Ro^{\gamma}(c)$, $\forall \alpha \in K(i,s)$, $\gamma \notin K(i,s)$. Then, $base^{SBnT}(i,s) = k_1$.

For example, $base^{SBnT}((011100),0) = 2$ and $base^{SBnT}((110100),(000010)) = 1$. Note that $|K(i,s)| = n/P(c)$ where $P(c)$ is the period of c. The value of the base equals the minimum number of right rotations that minimizes the value of c. For non-cyclic nodes $|K(i,s)| = 1$, but for a cyclic node c, $P(c) < n$, and $|K(i,s)| > 1$. The notion of $base^{SBnT}$ is similar to the notion of distinguished node used in [25] in that $base^{SBnT} = 0$ distinguishes a node from a generator set (necklace). To simplify the notation we omit the subscript on k in the following.

Definition 21 Subtree k of the *Spanning Balanced n-Tree* rooted at node s consists of all nodes $i \neq s$ such that $base^{SBnT}(i,s) = k$.

Note that all nodes in subtree k have $c_k = 1$, but not all nodes with $c_k = 1$ are in the k^{th} subtree.

Definition 22 Let $base^{SBnT}(i,s) = k$. For $c = 0$ let $p = -1$, else if $c_k = 1$ then $p = k$, else let p be the first bit cyclically to the right of bit k that is equal to 1 in c, i.e., $c_p = 1$, and $c_m = 0, \forall m \in \{(p+1) \bmod n, (p+2) \bmod n, \ldots, (k-1) \bmod n\} \equiv M^{SBnT}(i,s)$ with $k = n$ if $c = 0$. The spanning tree $T^{SBnT}(s)$ is defined through

$$children^{SBnT}(i,s) = \begin{cases} \{(i_{n-1}i_{n-2}\ldots\bar{i}_m\ldots i_0)\}, \forall m \in M^{SBnT}(i,s), & \text{if } c = 0; \\ \{q_m = (i_{n-1}i_{n-2}\ldots\bar{i}_m\ldots i_0)\}, \\ \quad \forall m \in M^{SBnT}(i,s) \text{ and } base^{SBnT}(q_m,s) = base^{SBnT}(i,s), & \text{if } c \neq 0. \end{cases}$$

$$parent^{SBnT}(i,s) = \begin{cases} \phi, & \text{if } c = 0; \\ (i_{n-1}i_{n-2}\ldots\bar{i}_p\ldots a_0), & \text{otherwise.} \end{cases}$$

The $parent^{SBnT}$ function preserves the base, since for any node i with base k, c_p is the highest order bit of $Ro^k(c)$. Complementing this bit cannot change the base. It is also readily seen that the $parent^{SBnT}$ and $children^{SBnT}$ functions are consistent.

Theorem 1 *The $parent^{SBnT}(i,s)$ function defines a spanning tree rooted at node s.*

Proof: For every node i the $parent^{SBnT}(i,s)$ function generates a path to node s. Hence, the graph is connected. Moreover, the parent node of a node at distance l from node s is at distance $l-1$ from node s, and each node only has one parent node. Hence, the graph is a spanning tree. ∎

Figure 2 shows a spanning balanced 5-tree in a 5-cube.

Lemma 11 The $SBnT$ is a greedy spanning tree.

Proof: From the definition of the $parent^{SBnT}(i,s)$ function it follows that the distance from node i to node s is $Hamming(i,s)$. ∎

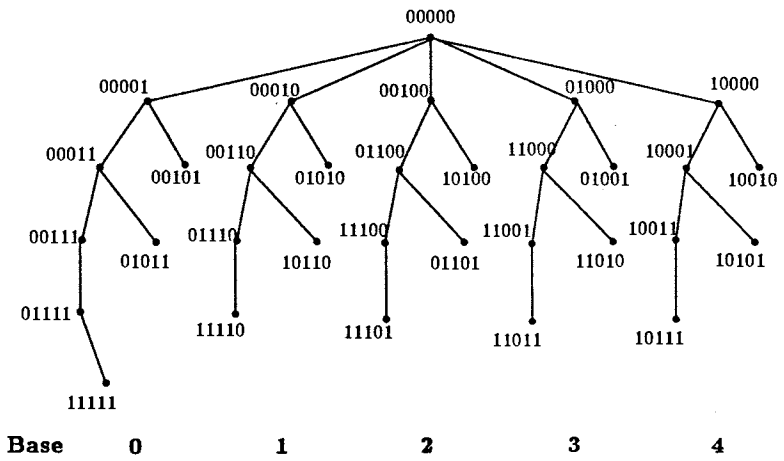

Figure 2: A spanning balanced 5-tree in a 5-cube.

Lemma 12 [15] Let $\phi^{SBnT}(i,s,x)$ be the number of nodes at distance x from node i in the subtree rooted at node i of the SBnT rooted at node s. Then, $\phi^{SBnT}(i,s,x) \geq \phi^{SBnT}(j,s,x)$, if $j \in children^{SBnT}(i,s)$.

Theorem 2 *Excluding node* $(\overline{s}_{n-1}\overline{s}_{n-2}\ldots\overline{s}_0)$, *all the subtrees of the root of the SBnT are isomorphic, if* n *is a prime number. Furthermore, the* k^{th} *subtree can be derived by* $(k-j) \bmod n$ *left rotation steps of each node of the* j^{th} *subtree.*

Proof: For n a prime number there are no cyclic nodes, except nodes with relative addresses $(00\ldots0)$ and $(11\ldots1)$. For all other nodes $|\mathcal{K}(i,s)| = 1$, and each generator set has n members. Any subtree has the same number of nodes as any other subtree at every level. It follows from the definition of $base^{SBnT}$ that if $j \in children^{SBnT}(i,s)$, then $Ro^k(j) \in children^{SBnT}(Ro^k(i),s), \forall k \in \mathcal{D}$. ∎

If n is not a prime number then some subtrees of the root of the SBnT will contain more nodes than others. It can be shown that the number of nodes in a subtree is of $O(\frac{N}{n})$ [15]. The imbalance is illustrated in Table 2. This imbalance is important for personalized communication. The number of elements transferred over the edges can be perfectly balanced in the sense that the maximum at any level is minimized by allowing multiple paths to cyclic nodes. With multiple paths to cyclic nodes the graph is no longer a tree. We call the SBnT so modified a *Spanning Balanced Graph* (SBG). The SBG can be defined as a composition of n distinctly rotated SBnT's as follows.

Definition 23 Define $G^{SBG}(0) = \cup_{d\in\mathcal{D}} Ro^d(T^{SBnT}(0))$ and $G^{SBG}(s) = Tr(s, G^{SBG}(0))$.

The number of different paths to a node i in an SBG rooted at node s is $\frac{n}{P(i\oplus s)}$ where $P(i)$ is the period of i.

3.2 Spanning Graphs Composed of n Spanning Trees

3.2.1 n Rotated Hamiltonian Paths

Definition 24 The graph $G^{nRH}(s) = Tr(s, G^{nRH}(0))$, and $G^{nRH}(0) = \cup_{d\in\mathcal{D}} Ro^d(T^H(0))$.

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

Table 2: A comparison of subtree sizes of SBT and SBnT. The last column contains the ratio of SBnT(max) to $\frac{N-1}{n}$.

The paths generated through n distinct rotations of the $G^H(s)$ path are not edge-disjoint, for $n > 2$. For instance, the edge $(2,6)$ is used in two paths of the graph $G^{3RH}(0)$, and these paths are part of every graph $G^{nRH}(0)$ for $n > 3$.

Lemma 13 Path $Ro^k(T^H(s))$ and path $Ro^m(T^H(s))$ share $2^{n-\alpha-1} + 2^{n-\beta-1} - 2$ edges, where $\alpha = (k-m) \bmod n$ and $\beta = (m-k) \bmod n$.

Each of the n cube edges $((00\ldots01_d0\ldots0),(00\ldots01_{(d+1) \bmod n}1_d0\ldots0))$, $\forall d \in \mathcal{D}$ are shared by $n-1$ paths for $s = 0$. Figure 3 shows the graph $G^{3H}(0)$. The fact that the paths are not edge-disjoint limits the potential for pipelining.

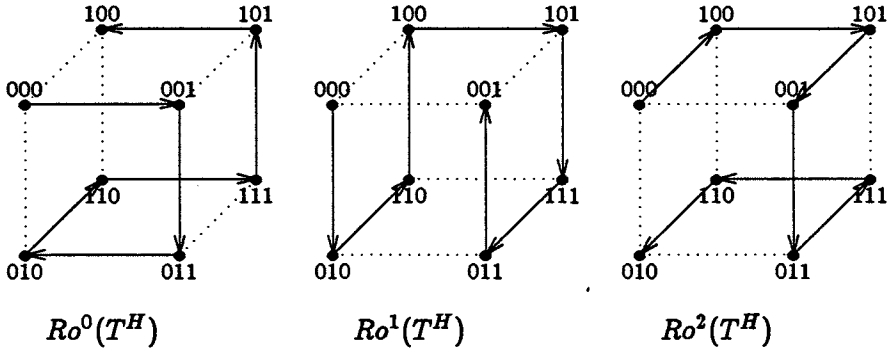

Figure 3: Three rotated binary-reflected Gray code paths in a 3-cube.

Lemma 14 In the graph $G^{nRH}(s)$, the edges between nodes at distance l and distance $l+1$ from the source node are edge-disjoint for $l \in \mathcal{N}$.

Proof: By construction the edges are in different dimensions. ∎

Note that even though the n rotated Hamiltonian paths are not edge-disjoint it is possible to have edge-disjoint embeddings of several Hamiltonian paths generated in other ways. But, to our knowledge, there does not exist 3 edge-disjoint paths on a 3-cube. However, there exist 4 edge-disjoint paths on an n-cube, $n \geq 4$, Figure 4.

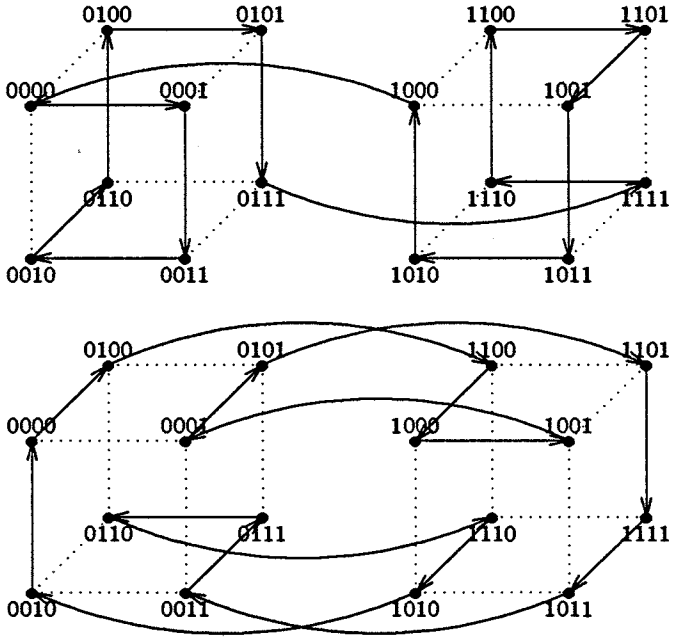

Figure 4: Four directed edge-disjoint Hamiltonian paths in a 4-cube. Only two paths are shown. The other two paths can be derived from the reversed paths.

3.2.2 n Rotated Spanning Binomial Trees

Definition 25 The graph $G^{nRSBT}(s) = Tr(s, G^{nRSBT}(0))$, and $G^{nRSBT}(0) = \cup_{d \in \mathcal{D}} Ro^d(T^{SBT}(0))$.

Figure 5 shows the nRSBT graph of a 3-cube. In general, a cube edge (i,j) is part of several rotated SBT's. The number on each cube edge in the Figure shows the sum of the weights of the graph edges mapped onto it.

Lemma 15 For any node in the graph $G^{nRSBT}(0)$, except the root, the weight of the incoming cube edge in dimension d is $\frac{p+1}{n}$, where p is the number of consecutive 0-bits immediately to the left of bit d.

Proof: A node in the SBT graph has an incoming edge in dimension d if the bit in dimension d is the highest order bit that is one. In the nRSBT graph the number of incoming edges (i,j) to node j is equal to the number of graphs $Ro^m(G^{SBT}(0))$, $m \in \mathcal{D}$ such that there exists an edge (i^m, j^m), where $i = Ro^m i^m$, and $j = Ro^m j^m$. Such an edge occurs in $Ro^m(\mathcal{E}^{SBT}(0))$ for $m = \{d+1, d+2, \ldots, d+p+1\}$, i.e., in these $p+1$ SBT's, bit d is the last bit complemented in reaching node j. ∎

For instance, node (011001) has an incoming cube edge in dimension 0 with weight $\frac{1}{2}$, an incoming edge in dimension 3 weighted $\frac{1}{6}$, and an incoming edge in dimension 4 weighted $\frac{1}{3}$.

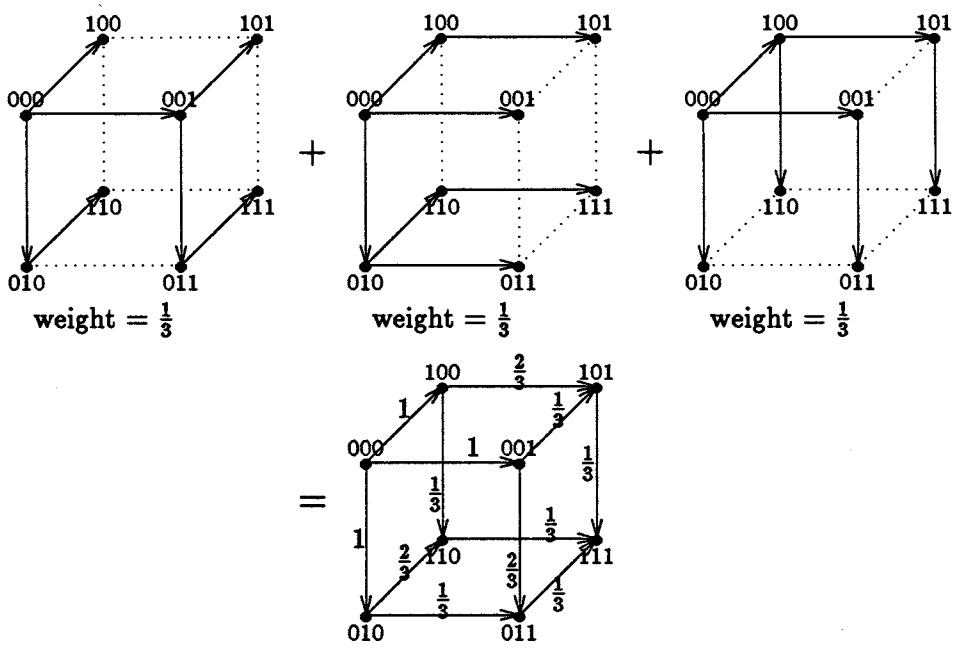

Figure 5: Three rotated spanning binomial trees as an *o-graph* in a 3-cube.

Corollary 4 The sum of the weights of incoming edges is 1 for every node, except the source node.

Proof: From lemma 15 the weight of an incoming edge is equal to the number of dimensions between the dimension considered and the next higher dimension with a 1-bit. ∎

Lemma 16 In any dimension the edges of the nRSBT graph are only mapped to half of the cube edges.

Proof: Rotation does not change the direction of edges. ∎

3.2.3 *n* Edge-Disjoint Spanning Binomial Trees

The nESBT (*n* Edge-disjoint Spanning Binomial Trees) graph is composed of *n* SBT's with one tree rooted at each of the nodes adjacent to the source node. The SBT's are rotated such that the source node of the nESBT graph is in the smallest subtree of each SBT. The nESBT graph is then obtained by reversing the edges from the roots of the SBT's to the source node.

Definition 26 The nESBT graph $G^{nESBT}(s) = Tr(s, G^{nESBT}(0))$, where $G^{nESBT}(0) = \cup_{d \in D} T^{SBT_d}(0)$, and $T^{SBT_d}(0) = Tr(2^d, Ro^{n-d-1}(T^{SBT}(0)))$ (with the root being node 0 instead of node 2^d).

Figure 6 shows an nESBT graph in a 3-cube. The nESBT graph is not a tree, and contains cycles. Every node appears in every subtree of the source node. The height of the nESBT graph is $n+1$, since the source node is adjacent to all the roots of the SBT's used in the definition of the nESBT graph. The number of distinct edges in the n SBT's is $n(N-1)$. An alternative definition of G^{nESBT} is through the functions $children^{nESBT}$ and $parent^{nESBT}$ below.

Definition 27 For a given k, let p be such that $c_p = 1$, and $c_m = 0$, $\forall m \in \{(p+1) \bmod n, (p+2) \bmod n, \ldots, (k-1) \bmod n\}$. For $c = 0$, $p = -1$, $k = n$, and for $c_k = 1$, $p = k$. Then

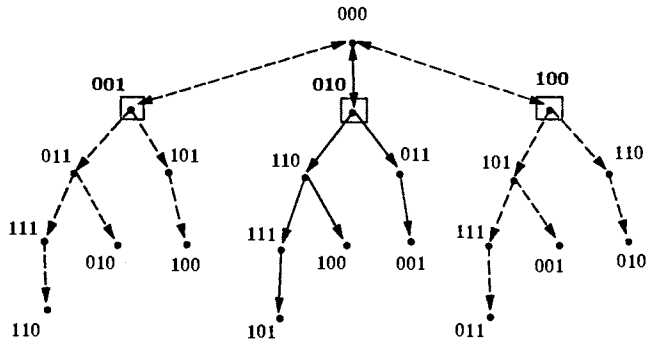

Figure 6: Subtrees of an nESBT viewed as SBT's.

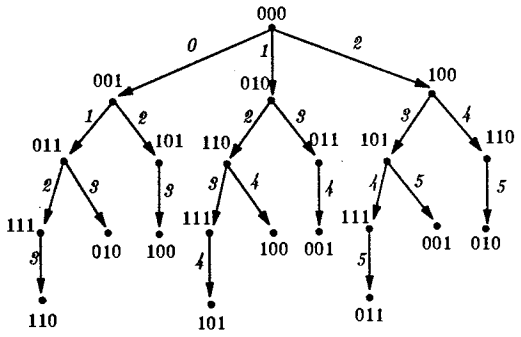

Figure 7: Three edge-disjoint directed spanning trees in a 3-cube.

$\mathcal{M}^{nESBT}(i,s,k) = \{(p+1) \bmod n, (p+2) \bmod n, \ldots, (k-1) \bmod n\}$. The children and parent of node i in the k^{th} spanning tree, $T^{SBT_k}(s)$, are:

$$children^{nESBT}(i,s,k) = \begin{cases} \{(i_{n-1}i_{n-2}\ldots\bar{i}_k\ldots i_0), & \text{if } c=0; \\ \{(i_{n-1}i_{n-2}\ldots\bar{i}_m\ldots i_0)\}, \forall m \in \mathcal{M}^{nESBT}(i,s,k)\bigcup\{k\}, & \text{if } c_k=1, p\neq k; \\ \{(i_{n-1}i_{n-2}\ldots\bar{i}_m\ldots i_0)\}, \forall m \in \mathcal{M}^{nESBT}(i,s,k), & \text{if } c_k=1, p=k; \\ \phi, & \text{if } c_k=0, c\neq 0. \end{cases}$$

$$parent^{nESBT}(i,s,k) = \begin{cases} \phi, & \text{if } c=0; \\ (i_{n-1}i_{n-2}\ldots\bar{i}_k\ldots i_0), & \text{if } c_k=0, c\neq 0; \\ (i_{n-1}i_{n-2}\ldots\bar{i}_p\ldots i_0), & \text{if } c_k=1. \end{cases}$$

Dimension p is the first dimension to the right of dimension k, cyclically, which has a bit equal to one. All nodes with $c_k = 0$, except node s, are leaf nodes of the k^{th} subtree (or k^{th} spanning tree). Conversely, all nodes with $c_k = 1$ are internal nodes of the k^{th} subtree. The exceptional connection to node 0 is handled by the conditions on p. The first case defines the children for the source node, the second the set of children nodes of internal nodes of the k^{th} subtree, except the node at level 1. The third case handles the node at level 1, and the last case handles the leaf nodes. Figure 7 and 12 show that the three subtrees (spanning trees) of a 3ESBT graph are edge-disjoint. The labels on the edges will be used later. Figure 8 shows a 4ESBT graph with source node 0. Every 1-bit in the node address divides the outgoing edges into distinct sets. The set of 0-bits to the right of a 1-bit defines children nodes for a spanning subtree with the same index as the dimension of the 1-bit, Figure 9.

Theorem 3 *The n subtrees of the nESBT graph are edge-disjoint.*

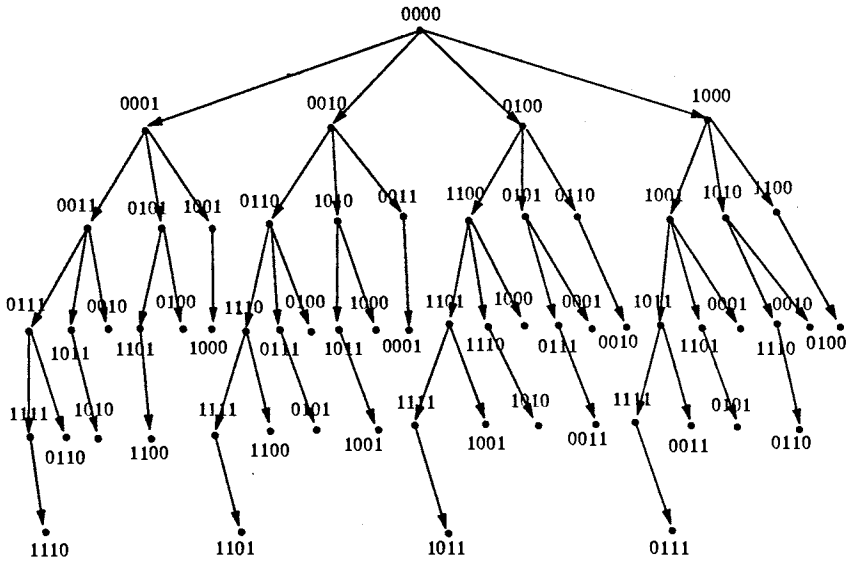

Figure 8: Four edge-disjoint directed spanning trees in a 4-cube.

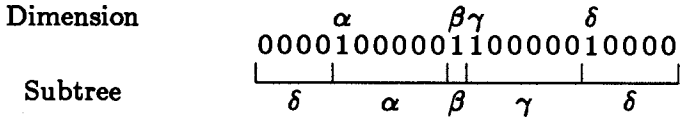

Figure 9: Scheduling of broadcasting operations for a node with nESBT routing.

Proof: We only need to prove that for an arbitrary node the address of its parent node in each of the n subtrees is obtained by complementing a distinct bit.

From the definition of the $children^{nESBT}(i,s,k)$ (or $parent^{nESBT}(i,s,k)$) function it is clear that a node is a leaf node of the k^{th} subtree iff $c_k = 0$, with the exception of node s. If a node is a leaf node in a particular subtree, then its parent address in that subtree is obtained by complementing the corresponding bit in its address (bit k for the k^{th} subtree). If a node is an internal node of the k^{th} subtree, then the corresponding bit is 1, and the parent address is obtained by complementing the first bit cyclically to the right of the k^{th} address bit that is equal to 1. Hence, the addresses of the parent nodes for all the subtrees of which the node is an internal node are also obtained by complementing distinct bits. ∎

Figure 10 shows the parents and children of one node in a 6-cube. The numbers on the edges are the dimensions through which the node connects to its parents or children nodes in different subtrees. The labels on the nodes denotes the subtree to which the parent and children nodes belong.

Corollary 5 There exists an edge-disjoint embedding of n Spanning Binomial Trees in an n-cube.

Corollary 6 The nESBT graph for a Boolean n-cube is a directed graph, such that all directed cube edges, except those incident on the source node, appear precisely once in the nESBT graph.

Proof: Follows from theorem 3. ∎

Corollary 7 The in-degree and the out-degree of any node in an nESBT graph is n, with the exception that the source has in-degree 0 and all the neighbors of the source have out-degree $n-1$.

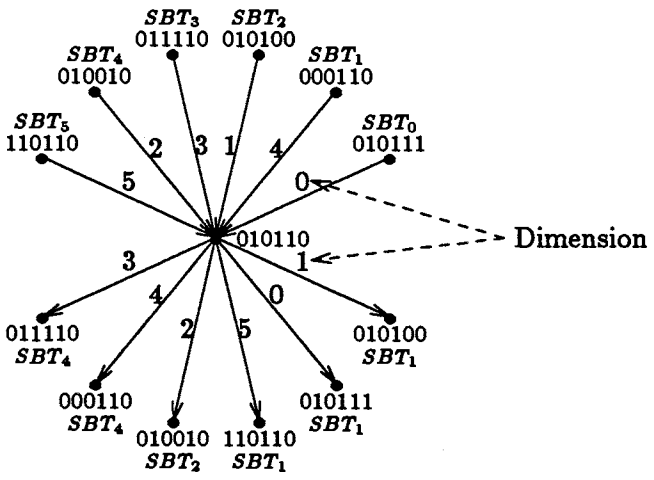

Figure 10: Parents and children of a node in a 6-cube.

Theorem 4 *The height of the nESBT graph is minimal among all possible configurations of n edge-disjoint spanning trees.*

Proof: To prove that with n edge-disjoint spanning trees, the height $n+1$ is minimal, we prove that n disjoint spanning trees with height n is impossible. The total number of directed edges in an n-cube is nN, but only the edges directed out from the source may be used. Each spanning tree has $N-1$ edges. Hence, every eligible edge is used by the n edge-disjoint spanning trees. It follows that the edges directed out from the node with address $(\overline{s}_{n-1}\overline{s}_{n-2}\ldots\overline{s}_0)$ also must be used, and since this node is at distance n from the source node, the theorem follows. ∎

Lemma 17 The number of nodes at level l of a subtree of the nESBT graph is

$$\begin{cases} \binom{n-1}{l-1}+\binom{n-1}{l-2}=\binom{n}{l-1}, & \text{for } n+1\geq l\geq 1, l\neq 2; \\ 1, & \text{for } l=0; \\ n-1, & \text{for } l=2. \end{cases}$$

Proof: Follows directly from the definition. ∎

Lemma 18 Let $\phi^{nESBT}(i,s,x)$ be the number of nodes at distance x from node i in the subtree rooted at node i of the nESBT rooted at node s. Then, $\phi^{nESBT}(i,s,x) \geq \phi^{nESBT}(j,s,x)$, if $j \in children^{nESBT}(i,s)$.

Proof: By definition 26, each subtree of the nESBT is a SBT with the smallest subtree removed. The lemma can be shown by lemmas 10, 17 and 9. ∎

3.3 Summary of Topological Properties of the Communication Graphs

The topological characteristics used for the complexity analysis are summarized in Table 3.

With respect to the entry for the SBT graph in Table 3 note that $\binom{x}{y}=0$ if $x<y$. Moreover,

$$\max_d v^{SBT}(d,l) = v^{SBT}(\min(2l,n-1),l) = \begin{cases} \binom{2l}{l}2^{n-2l-1}, & \text{if } l\in\{0,\ldots,\lfloor\frac{n-1}{2}\rfloor\}; \\ \binom{n-1}{l}, & \text{if } l\in\{\lfloor\frac{n+1}{2}\rfloor,\ldots,n-1\}. \end{cases}$$

Graph	$E(d)$	$e(d,l)$	$v(d,l)$	$u(d,l)$
G^H	2^{n-d-1}	$0,\ d \neq t_l$ or 1, o.w.	$0,\ d \neq t_l$ or $N-l-1$, o.w.	$\frac{N}{2^{d+1}} - \lceil \frac{l-2^d+1}{2^{d+1}} \rceil$
G^{nRH}	$\frac{N-1}{n}$	$\frac{1}{n}$	$\frac{1}{n}(N-l-1)$	$\frac{1}{n}(N-l-1)$
G^{SBT}	2^d	$\binom{d}{l}$	$\binom{d}{l}2^{n-d-1}$	$\binom{n-d-1}{l}2^d$
G^{nRSBT}	$\frac{N-1}{n}$	$\frac{1}{n}\binom{n}{l+1}$	$\frac{1}{n}\sum_{i=l+1}^{n}\binom{n}{i}$	$\frac{1}{n}\sum_{i=l+1}^{n}\binom{n}{i}$
G^{nESBT}	$\frac{N-1}{n}$	$1, l=0;\ n-1, l=1;\ \frac{1}{n}\binom{n}{l}, l \geq 2$	$N-1, l=0;\ N-2, l=1;\ \frac{1}{n}\sum_{i=l}^{n}\binom{n}{i}, l \geq 2$	
G^{SBG}	$\frac{N-1}{n}$	$\frac{1}{n}\binom{n}{l+1}$	$\frac{1}{n}\sum_{i=l+1}^{n}\binom{n}{i}$	$\frac{1}{n}\sum_{i=l+1}^{n}\binom{n}{i}$

Table 3: Some topological characteristics of some *o-graphs*.

The entries in Table 3 for the SBG graph can be proved by lemmas 3 and 11, and corollaries 2 and 3. The characteristics for the nRH graph can be proved from definition 24 and lemmas 3 and 7, and the entries for the nRSBT graph proved using definition 25, corollaries 2, 3, and the fact that T^{SBT} is greedy. The nESBT properties can be proved using definition 26 and lemmas 3, 7 and 17.

4 One-to-All Broadcasting

Lemma 19 A lower bound for *one-to-all broadcasting* with *one-port* communication is $\max((M+n-1)t_c, n\tau)$, and $\max((\lceil \frac{M}{n} \rceil + n - 1)t_c, n\tau)$ for *n-port* communication.

Proof: The height of any *o-graph* is at least n. The root needs a time of Mt_c (or $\lceil \frac{M}{n} \rceil t_c$) to send out the data. An additional delay of at least $(n-1)t_c$ is required to reach the node at maximum distance from the root. ∎

With $M = 1$ a tight lower bound is $n(t_c + \tau)$ both for *one-port* and *n-port* communication. This bound is realized by SBT routing and appropriate scheduling. In fact, it can be shown that with $M = 1$ and *one-port* communication, any routing for broadcasting that yields the lower bound is topologically equivalent to the graph G^{SBT}. With *n-port* communication any *o-graph* of height n can realize the lower bound communication for $M = 1$. The scheduling discipline we propose for the graph G^{nESBT} yields the lowest communication complexity of the broadcasting algorithms we consider, both for *one-port* and *n-port* communication, except $B_{opt} = M$ for *one-port* communication and $B_{opt} = \frac{M}{n}$ for *n-port* communication. For these two cases our scheduling for the graph G^{nESBT} is inferior only by 1 routing cycle. With $1 < M \leq n$ and *n-port* communication , our scheduling for the graph G^{nRSBT} yields the lower bound, $n(t_c + \tau)$.

4.1 The Time Complexity of One-to-All Broadcasting

Table 4 summarizes the routing algorithms and scheduling disciplines we have analyzed. The estimated communication times are summarized in Table 5. For an *o-graph* composed of n spanning trees the data set M is divided into n approximately equal size subsets, and each such subset communicated by one of the spanning subtrees. If the trees are not edge-disjoint we assume no pipelining.

For the graph G^{nRH} the n paths are not edge-disjoint. But, all edges at a given distance from the source are mapped to different cube edges. Sending the entire data set $\frac{M}{n}$ for each path concurrently is contention free. The data transfer time is an order of $\frac{N}{n}$ higher than that of the H

Comm.	Routing	Scheduling discipline
one-port	H	Pipeline.
	SBT	All data for a port at once. Tallest remaining subtree first.
	nRSBT	All data for the tallest remaining subtree of the spanning trees $0, 1, \ldots, p$ at once during routing cycle $p < n$, and all data for the tallest remaining subtrees of all spanning trees during cycles $n \leq p < 2n-1$. For the source $B = \frac{p+1}{n}M$ for $p < n$, and $B = \frac{2n-p-1}{n}M$ for $n \leq p < 2n-1$.
	nESBT	The source node sends to spanning trees $0, 1, \ldots, n-1$ cyclically, i.e., pipeline among n spanning trees. The internal nodes always propagate (replicate) to the tallest subtree of the same spanning tree first.
n-port	SBT	Pipelining for each subtree, subtrees concurrently.
	nRSBT	All data at once for the tallest remaining subtree of every spanning tree. (The subtrees of each spanning tree are treated sequentially, and spanning trees concurrently.)
	nESBT	Pipelining for each subtree; subtrees and spanning trees concurrently.

Table 4: Scheduling disciplines for one-to-all broadcasting.

Comm.	Routing	T	B_{opt}	T_{min}
one-port	H	$(M+(N-2)B)t_c + (\lceil \frac{M}{B} \rceil + N - 2)\tau$	$\sqrt{\frac{M\tau}{(N-2)t_c}}$	$(\sqrt{Mt_c} + \sqrt{(N-2)\tau})^2$
	SBT	$Mnt_c + \lceil \frac{M}{B} \rceil n\tau$	M	$n(Mt_c + \tau)$
	nRSBT	$nMt_c + (2\sum_{i=1}^{n-1} \lceil \frac{Mi}{nB} \rceil + \lceil \frac{M}{B} \rceil)\tau$	M	$nMt_c + (2n-1)\tau$
	nESBT	$(M+nB)t_c + (\lceil \frac{M}{B} \rceil + n)\tau$	$\sqrt{\frac{M\tau}{nt_c}}$	$(\sqrt{Mt_c} + \sqrt{n\tau})^2$
n-port	SBT	$(M+(n-1)B)t_c + (\lceil \frac{M}{B} \rceil + n - 1)\tau$	$\sqrt{\frac{M\tau}{(n-1)t_c}}$	$(\sqrt{Mt_c} + \sqrt{(n-1)\tau})^2$
	nRSBT	$Mt_c + \lceil \frac{M}{nB} \rceil n\tau$	$\frac{M}{n}$	$Mt_c + n\tau$
	nESBT	$(\frac{M}{n} + nB)t_c + (\lceil \frac{M}{Bn} \rceil + n)\tau$	$\frac{1}{n}\sqrt{\frac{M\tau}{t_c}}$	$(\sqrt{\frac{Mt_c}{n}} + \sqrt{n\tau})^2$

Table 5: The complexity of one-to-all broadcasting.

routing with pipelining. For nRSBT routing the number of elements traversing every edge from the root is the same as in the SBT routing, and the transmission time is bounded from below by nMt_c with *one-port* communication. This time is an order of n higher than the lower bound.

Theorem 5 *For a fixed packet size, the number of routing cycles to broadcast n packets by nRSBT routing is bounded from below by* $2n-1$ *for* one-port *and* n *for* n-port *communication.*

Proof: For *one-port* (*n-port*) communication, the root needs n (1) cycles to send out n packets, and the last packet has a latency of $n-1$ cycles for a fixed packet size. ∎

With *one-port* communication, our scheduling discipline for the nRSBT routing completes in $2n-1$ routing cycles for a packet size $B \geq M$. With *n-port* communication the scheduling of the data for each SBT used in the nRSBT graph is made as in the case of *one-port* communication for a single SBT. Since the SBT's are rotated, all ports of the root are used in every routing cycle until the last packet leaves the root. Data for the m^{th} SBT is sent across dimension $(m+p) \bmod n$ during cycle p. There is no edge-conflict for this non-pipelined routing. Figure 11 shows the routings of the three distinctly rotated SBT's in a 3-cube. The labels on the edges represent the routing cycle.

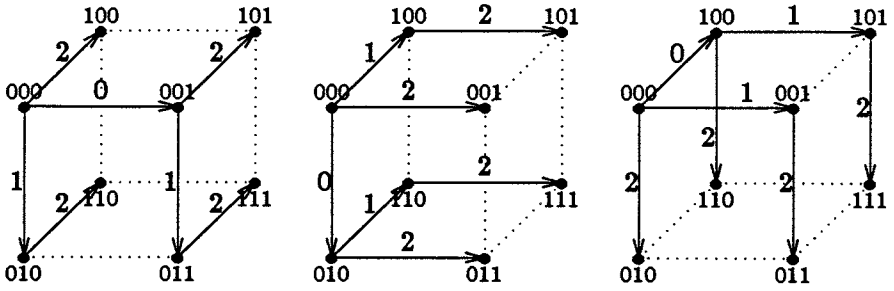

Figure 11: Broadcasting based on 3 rotated SBT's in a 3-cube.

The number of routing cycles to broadcast n packets by nESBT routing is bounded from below by $2n$ for *one-port* and $n+1$ for *n-port* communication, since the height of the graph is $n+1$. The scheduling discipline for *one-port* communication and nESBT routing is defined by labeling the edges of the nESBT graph. The labels define the routing cycles during which the first packet arrives through that graph edge. We first define the edge label in the 0^{th} spanning tree of the nESBT graph. Edges in dimension m is labeled m, except that the labels of the edges to the leaf nodes are labeled n. The labels of the edges in the k^{th} spanning tree are defined by adding k to the label of the corresponding edges in the 0^{th} spanning tree.

Figure 7 shows an nESBT graph for a 3-cube labeled by the algorithm above. Figure 12 shows a different view of the labels of the 3 composed spanning trees.

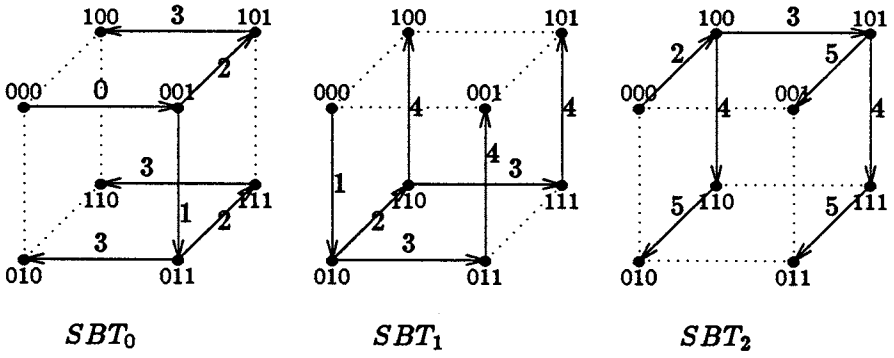

Figure 12: Scheduling in an nESBT graph with *one-port* communication.

Theorem 6 *For the nESBT graph the scheduling discipline defined by the labeling scheme allows conflict free* one-port *communication.*

Proof: It follows from the labeling scheme that edges in dimension m are labeled m (mod n). Also, for each spanning tree, the label of the outgoing edges of any node are greater than the label of the incoming edge. ∎

The largest label of all the input edges is $2n-1$. Broadcasting the first n packets (one packet per subtree) can be done in $2n$ cycles. The complexity for *one-port* nESBT routing follows directly from the definition of the scheduling discipline, and the proof of it being contention free. For *n-port* communication it is easy to determine the time of arrival of messages. The path length between nodes s and i in the k^{th} spanning tree is equal to

$$\begin{cases} |c|-1, & \text{if } c_k = 1; \\ |c|+1, & \text{if } c_k = 0. \end{cases}$$

The input ports of node i that correspond to bits that are equal to 1 in the binary encoding of c receives the first element during the $(|c|-1)^{th}$ routing cycle. The other input ports receive the first element during the $(|c|+1)^{th}$ cycle.

4.2 Summary of One-to-All Broadcasting

In general, the data set to be broadcast is divided into a number of distinct packets. If the *o-graph* is a spanning tree, then every packet needs to be sent on every port of the root. The number of routing cycles required by the root to complete the communication of each distinct packet is summarized on the left of Table 6, and the maximum propagation delay is summarized on the right.

Routing	# of cyc. per pkt.		Propagation delay
	one-port	*n-port*	*one- & n-port*
H	1	1	$N-1$
SBT	n	1	n
nRSBT	n	1	n
nESBT	1	$\frac{1}{n}$	$n+1$

Table 6: Number of cycles per distinct packet (left) and propagation delay (right).

In the nRSBT routing, every packet needs to be routed on every port because of the edge sharing. Note, that broadcasting by Hamiltonian path routing may be faster than by SBT routing depending on the values of M, t_c, τ and N. With *n-port* communication, the source can send out n distinct packets every cycle in the nESBT routing since the spanning trees are edge-disjoint, but only one packet for the nRSBT routing.

For *n-port* communication, the data transmission time is reduced by a factor of approximately n for an arbitrary packet size in the SBT, nRSBT, and nESBT routings and the schedulings defined in Table 4. Optimizing the packet size makes the number of routing cycles proportional to the height of the *o-graph*. The data transmission times for the optimum packet sizes and *n-port* communication are approximately a factor of n less than the transmission times for *one-port* communication with optimized buffers and an *o-graph* for which the root has degree n.

The nESBT routing always offers a reduction in bandwidth requirement for individual communication links by a factor of approximately n over the SBT and nRSBT routings. The nESBT routing offers a speed-up of up to n over SBT and nRSBT routings for sufficiently large values of M, both for *one-port* and *n-port* communication. Communication complexities of broadcasting based on the H, SBT and nRSBT algorithms are compared with that based on the nESBT in Table 7.

5 One-to-All Personalized Communication

In personalized communication no replication of information takes place during distribution, nor is there any reduction during the reverse operation; the root has M elements for every node.

Lemma 20 The data transmission time for *one-to-all personalized communication* is bounded from below by $(N-1)Mt_c$ for *one-port* communication, and by $\frac{(N-1)M}{n}t_c$ for *n-port* commu-

Communication model	Routing	One packet	$B = B_{opt}$, $\tau \gg Mt_c$	$B = B_{opt}$, $n^2\tau \ll Mt_c$
one-port	H/nESBT	$\frac{N-1}{n+1}$	$\frac{N-2}{n}$	1
	SBT/nESBT	$\frac{n}{n+1}$	≈ 1	n
	nRSBT/nESBT	$\frac{n}{n+1}$	≈ 2	n
n-port	H/nESBT	$\frac{N-1}{n+1}$	$\frac{N-2}{n}$	n
	SBT,nRSBT/nESBT	$\frac{n}{n+1}$	≈ 1	n

Table 7: Relative complexities of one-to-all broadcasting.

nication. A lower bound for the total time is $\max((N-1)Mt_c, n\tau)$, and $\max(\frac{(N-1)M}{n}t_c, n\tau)$, respectively.

Proof: The root needs to send out $(N-1)M$ elements. ∎

In SBT routing for personalized communication the maximum number of nodes connected to the root through one of its edges is $\frac{1}{2}NM$, and a data transmission time of the order given in the lemma is not achievable for *n-port* communication. In a SBnT graph all edges of the root connect subtrees of approximately equal size. For the SBG, nRSBT and nESBT graphs, all outgoing (cube) edges of the root transmit the same amount of data, $\frac{(N-1)M}{n}$.

5.1 The Time Complexity of Personalized Communication

The general strategy for personalized communication is to schedule the data for the most remote nodes first. With *n-port* communication the ordering between ports is irrelevant, and a *reverse-breadth-first* ordering for each port of each node is the scheduling discipline. For *one-port* communication the scheduling disciplines depend on the *o-graph*, Table 8.

Comm.	Routing	Scheduling discipline
one-port	H	Order nodes by decreased distance, pipeline.
	SBT	Order nodes by complementing their binary addresses.
	nESBT	All data for the tallest remaining subtree for each spanning tree.
	nRSBT	All data for the tallest remaining subtree of the spanning trees $0, 1, \ldots, p$ at once during routing cycle $p < n$, and all data for the tallest remaining subtrees of all spanning trees during cycles $n \leq p < 2n-1$. For the source $B = N(1 - \frac{1}{2^{p+1}})\frac{M}{n}$ for $p < n$, and $B = N(\frac{1}{2^{p+1-n}} - \frac{1}{N})\frac{M}{n}$ for $n \leq p < 2n-1$.
	SBG	Order nodes by decresing distance for each subtree. Order subtrees cyclically.
n-port	All	Order nodes in *reverse-breadth-first* order. All subtrees (spanning trees) are scheduled concurrently.

Table 8: Scheduling disciplines for one-to-all personalized communication.

Lemma 21 Scheduling nodes by complementing their binary addresses results in port communications in a binary-reflected Gray code order for *one-port* SBT routing.

Note that for *one-port* communication, scheduling the tallest remaining subtree first for each node recursively has the same complexity as the suggested discipline, but if a certain overlap

between communications on different ports is possible, as on the Intel iPSC, then the discipline we adopt yields a lower time complexity.

5.2 Summary of One-to-All Personalized Communication

Tables 9 and 10 summarize the communication complexities of personalized communication. For the SBT routing and *one-port* communication, our scheduling discipline yields a time complexity within a factor of two of the lower bound, providing the packet size is sufficiently large. The data transmission time of the nRSBT scheduling discipline is the same as the lower bound, $M(N-1)t_c$, however, the minimum number of routing cycles is $2n-1$. For the SBG routing and $B \geq \frac{(N-1)M}{n}$ the root needs to perform only one communication per subtree, and completes the communication in time $T = (N-1)Mt_c + n\tau$. But, an additional $n-1$ routing cycles are needed to complete the communication. An upper bound on the time for personalized communication with unbounded packet size is $T = N(1 + \frac{2\log n}{n})Mt_c + (2n-1)\tau$ for SBG routing [15]. The number of routing cycles is almost twice that of the SBT personalized communication, and the total transfer time is higher by a lower order term. With a packet size of $\frac{(N-1)M}{n}$, the number of routing cycles is approximately $2n$ for SBT, nESBT, nRSBT, and SBG communication. With *one-port* communication and a packet size $B \leq M$, the complexities of SBT, nRSBT, SBG and H communications are approximately the same.

With *n-port* communication and *reverse-breadth-first* scheduling, the data transmission time of the SBT, nESBT, nRSBT and SBG routings are all dominated by the root by lemmas 6, 10, 12 and 18. The number of element transfers is approximately the same for all ports of the root except for the SBT communication. The number of routing cycles and the transmission time of the nESBT, SBG and nRSBT communications are lower than that of the SBT by a factor of $\frac{1}{2}n$ for a packet size $B \leq M$. With a sufficiently large packet size all communications yield a minimum of n routing cycles, with the exception of the nESBT, which requires one more cycle. The optimum packet size for the nESBT, nRSBT and SBG communications is $\sqrt{\frac{2}{\pi}}\frac{NM}{n^{3/2}}$, compared to $\frac{NM}{\sqrt{2\pi(n-1)}}$ for the SBT. The nRSBT routing is never of a lower complexity than the SBG routing, and of a higher complexity if n does not divide M. With $(M \bmod n) = k \neq 0$ a combination of *reflections* and *rotations* minimizes the maximum number of elements that need to be transferred over any cube edge. For k even $k/2$ distinct rotations should be used, and for every rotated graph a reflected graph is also used. For k even and optimally rotated SBT's the maximum number of elements transferred over a cube edge is $(N-1)\frac{2^{\frac{n}{k}}-1}{2^{\frac{n}{k}}-1}$ and for optimally reflected and rotated SBT's it is $(N-1)\frac{2^{\frac{2n}{k}-1}+1}{2^{\frac{2n}{k}}-1}$.

6 All-to-All Broadcasting

6.1 Time Bounds and Scheduling Disciplines

Theorem 7 *A lower bound for all-to-all broadcasting is* $\max((N-1)Mt_c, n\tau)$ *for* one-port *communication and* $\max(\frac{(N-1)M}{n}t_c, n\tau)$ *for* n-port *communication.*

Proof: Each node receives M elements from every other node, i.e., each node receives $(N-1)M$ elements. Hence, for *one-port* communication a lower bound for the data transfer time is $(N-1)Mt_c$, and with *n-port* communication the time is bounded by $\frac{(N-1)M}{n}t_c$. ∎

Comm.	Routing	T
one-port	H	$(N-1)Mt_c + \max(\lceil \frac{(N-1)M}{B} \rceil, N-1)\tau$
	SBT	$(N-1)Mt_c + \sum_{i=0}^{n-1} \lceil \frac{2^i M}{B} \rceil \tau$
	nESBT	$\frac{n+1}{n}(N-1)Mt_c + (n\lceil \frac{(N-1)M}{nB} \rceil + \sum_{i=0}^{n-1} \lceil \frac{2^i M}{nB} \rceil)\tau$
	nRSBT	$(N-1)Mt_c + (\sum_{i=1}^{n-1} \lceil \frac{(N-2^i)M}{nB} \rceil + \sum_{i=1}^{n} \lceil \frac{(2^i-1)M}{nB} \rceil)\tau$
n-port	SBT	$\frac{NM}{2}t_c + \sum_{i=0}^{n-1} \lceil \binom{n-1}{i} \frac{M}{B} \rceil \tau$
	nESBT	$\frac{(N-1)M}{n}t_c + (\lceil \frac{M}{nB} \rceil + \lceil \frac{(n-1)M}{nB} \rceil + \sum_{i=2}^{n} \lceil \frac{M}{nB} \binom{n}{i} \rceil)\tau$
	nRSBT	$\frac{(N-1)M}{n}t_c + \sum_{i=0}^{n-1} \lceil \binom{n}{i} \frac{M}{nB} \rceil \tau$
	SBG	$\frac{(N-1)M}{n}t_c + \sum_{i=0}^{n-1} \lceil \binom{n}{i} \frac{M}{nB} \rceil \tau$

Table 9: The complexity of one-to-all personalized communication.

Comm.	Routing	B_{opt}	T_{min}
one-port	H	M	$(N-1)Mt_c + (N-1)\tau$
	SBT	$\frac{NM}{2}$	$(N-1)Mt_c + n\tau$
	nESBT	$\frac{(N-1)M}{n}$	$\frac{n+1}{n}(N-1)Mt_c + 2n\tau$
	nRSBT	$\frac{(N-1)M}{n}$	$(N-1)Mt_c + (2n-1)\tau$
	SBG	$\frac{(N-1)M}{n}$	$\leq N(1+\frac{2\log n}{n})Mt_c + (2n-1)\tau$
n-port	SBT	$\frac{NM}{\sqrt{2\pi(n-1)}}$	$\frac{NM}{2}t_c + n\tau$
	nESBT	$\sqrt{\frac{2}{\pi}\frac{NM}{n^{3/2}}}$	$\frac{(N-1)M}{n}t_c + (n+1)\tau$
	nRSBT	$\sqrt{\frac{2}{\pi}\frac{NM}{n^{3/2}}}$	$\frac{(N-1)M}{n}t_c + n\tau$
	SBG	$\sqrt{\frac{2}{\pi}\frac{NM}{n^{3/2}}}$	$\frac{(N-1)M}{n}t_c + n\tau$

Table 10: The optimum complexity of one-to-all personalized communication.

Lemma 22 The data transfer time for *one-port* communication is minimized if one dimension is routed per cycle, and all nodes use the same scheduling discipline.

Proof: The number of elements transferred on every edge in the dimension subject to routing is the same, since the communication graphs for the different sources are translations of each other. ∎

For *one-port* all-to-all broadcasting it remains to define a scheduling discipline that minimizes the number of start-ups, preserving the minimum data transfer time. This can be accomplished implicitly by labeling the edges of the *o-graph* generating the *a-graph*. The label on the edge corresponds to the cycle during which the data is transferred across that edge. The rules are summarized in Table 11, which also gives the scheduling discipline for the *n-port* case. Rule 1 is obvious. Rule 2 is a sufficient condition satisfying the *one-port* communication constraint. The labeling scheme is the same as the one used in the *one-port one-to-all personalized communication* based on the nESBT graph. The number of start-ups required for the broadcasting is equal to the maximum label plus 1 (the least label being 0). For a spanning tree of height h, the minimax label is at least $h-1$ by rule 1.

For the H path, we label the i^{th} edge in the path i. The maximum label is $N-2$, which is a minimax label. For the SBT graph labeling edges in dimension i by i satisfies both rules. The maximum label is $n-1$, which is also a minimax label. For an *a-graph* based on the nESBT, nRSBT or SBG graphs, we first define the edge label in the 0^{th} spanning tree of each such graph.

Comm.	Scheduling discipline
one-port	1. For each spanning tree of the *a-graph*, the labels of the outgoing edges of any node are greater than the label of the incoming edge. 2. All the edges with the same label in the *o-graph* are in the same dimension.
n-port	All data sent at once, spanning trees concurrently.

Table 11: Scheduling disciplines for all-to-all broadcasting.

Edges in dimension i are labeled i, except for the nESBT graph for which the labels of the edges to the leaf nodes (all in dimension 0) is n. The 0^{th} subtree of the nESBT graph is equal to an n level SBT with the smallest subtree deleted. Hence, the minimax label is n for the 0^{th} spanning tree of the nESBT graph. For the nESBT and nRSBT graphs, the labels of the edges in subtree j are defined by adding j to the label of the corresponding edges in subtree 0. The minimax label for the entire graph is equal to the minimax label of subtree 0 plus $n-1$. For the SBG graph, all the n composed SBnT's are labeled in the same way. By theorem 2, it can be shown that the minimax label of the rotated SBnT $R^d(G^{SBnT})$, $d \in \mathcal{D}$, is $2n-2$ if $d=1$; and $2n-3$, otherwise. The minimax label of the nESBT, nRSBT and SBG are $2n-1$, $2n-2$ and $2n-2$ respectively. The labeling of the nESBT graph defined here is the same as the labeling for *one-port* one-to-all broadcasting, Figure 12. The amount of data transferred during cycle i is equal to $\frac{M}{n}\times$ (number of edges labeled i). The maximum packet size and the number of start-ups can be derived easily from the labels of the edges.

Lemma 23 A lower bound for data transmission time of all-to-all broadcasting based on $G^{id}(*)$ and "any" scheduling discipline is

$$\max_{\forall d \in \mathcal{D}} E^{id}(d) M t_c.$$

Proof: $E^{id}(d)M$ elements need to be sent across every cube edge in dimension d. ∎

Theorem 8 *The communication time for all-to-all broadcasting based on $G^{id}(*)$ of height h,* n-port *communication and the defined scheduling discipline requires a time of*

$$T = \sum_{l=0}^{h-1} \left(M t_c \times \max_{\forall d \in \mathcal{D}} e^{id}(d,l) + \left\lceil \frac{M}{B} \times \max_{\forall d \in \mathcal{D}} e^{id}(d,l) \right\rceil \tau \right).$$

If $B \geq \max_{0 \leq l \leq h-1,\ \forall d \in \mathcal{D}} (M \times e^{id}(d,l))$ then

$$T = \left(\sum_{l=0}^{h-1} \max_{\forall d \in \mathcal{D}} e^{id}(d,l) \right) M t_c + h\tau.$$

Proof: Each node broadcasts its data set M according to its own *o-graph*. During the l^{th} routing cycle all nodes at level $l+1$ of each *o-graph* receives messages sent out from the roots during the 0^{th} routing cycle. By lemma 8, the amount of data contending for a communication link in dimension d is $M \times e^{id}(d,l)$. ∎

Theorem 7 gives a lower bound for the communication time for any routing and scheduling discipline, and lemma 23 a lower bound expressed in terms of an *a-graph*. Theorem 8 gives an

upper bound. Next we give complexity estimates for some *a-graphs*, and show that broadcasting based on $G^{SBT}(*)$ with the scheduling disciplines in Table 11 is optimum within a factor of 2 for *one-port* communication, and that the schedulings of the $G^{SBG}(*)$ and $G^{nRSBT}(*)$ graphs are optimum within a factor of 2 for *n-port* communication.

Corollary 8 For a given *a-graph* satisfying $e^{id}(x,l) = e^{id}(y,l)$, $\forall x, y \in \mathcal{D}$, and $0 \leq l < h$, the communication time for *all-to-all broadcasting* based on the graph $G^{id}(*)$ of height h and *n-port* communication and the scheduling disciplines of Table 11 is

$$T = \frac{(N-1)M}{n} t_c + h\tau, \text{ if } B \geq \max_{0 \leq l < h}(M \times e^{id}(d,l)).$$

Proof: The corollary follows from theorem 8. ∎

From Table 3 and corollary 8, the nRH, nRSBT, nESBT and the SBG routings all yield the lower bound for the data transfer time with *n-port* communication. In fact, following corollary 8, we have the following corollary.

Corollary 9 For $G^{id}(*)$ with $G^{id}(0)$ composed of n distinctly rotated spanning trees, the data transfer time for all-to-all broadcasting is minimum with *n-port* communication and the defined scheduling.

Routing according to a greedy *o-graph*, while it is a necessary condition for the *all-to-all personalized communication* to attain the minimum data transfer time, is not necessary for the *all-to-all broadcasting*. The nRSBT and the SBG both with minimum height also attain the minimum number of start-ups.

6.2 The Complexity of All-to-All Broadcasting

Tables 12 and 13 summarize the complexity of all-to-all broadcasting.

Comm.	Routing	T
one-port	H	$(N-1)Mt_c + \lceil \frac{M}{B} \rceil (N-1)\tau$
	SBT	$(N-1)Mt_c + \sum_{i=0}^{n-1} \lceil \frac{2^i M}{B} \rceil \tau$
	nRSBT	$(N-1)Mt_c + (\sum_{i=0}^{n-1} \lceil \frac{(2^{i+1}-1)M}{nB} \rceil + \sum_{i=n}^{2n-2} \lceil \frac{(N-2^{i-n+1})M}{nB} \rceil)\tau$
	nESBT	$(N-1)Mt_c + (\sum_{i=0}^{n-1} \lceil \frac{2^i M}{nB} \rceil + \sum_{i=n}^{2n-1} \lceil \frac{(N-1-2^{i-n})M}{nB} \rceil)\tau$
n-port	nRH	$\frac{(N-1)M}{n} t_c + \lceil \frac{M}{nB} \rceil (N-1)\tau$
	SBT	$\frac{NM}{2} t_c + \sum_{i=0}^{n-1} \lceil \binom{n-1}{i} \frac{M}{B} \rceil \tau$
	nRSBT	$\frac{(N-1)M}{n} t_c + \sum_{i=1}^{n} \lceil \binom{n}{i} \frac{M}{nB} \rceil \tau$
	nESBT	$\frac{(N-1)M}{n} t_c + (\sum_{i=2}^{n} \lceil \binom{n}{i} \frac{M}{nB} \rceil + \lceil \frac{M}{nB} \rceil + \lceil \frac{(n-1)M}{nB} \rceil)\tau$
	SBG	$\frac{(N-1)M}{n} t_c + \sum_{i=1}^{n} \lceil \binom{n}{i} \frac{M}{nB} \rceil \tau$

Table 12: The complexity of all-to-all broadcasting.

The *one-port* $G^H(*)$ routing is employed in the matrix multiplication algorithm by Dekel [5], [21]. Messages with different source nodes are routed through different H paths. Messages are exchanged along a sequence of dimensions such as 0, 1, 0, 2, 0, 1, 0, 3, ..., etc. The SBT communication amounts to a single exchange per dimension [28].

Comm.	Routing	B_{opt}	T_{min}
one-port	H	M	$(N-1)Mt_c + (N-1)\tau$
	SBT	$\frac{NM}{2}$	$(N-1)Mt_c + n\tau$
	nRSBT	$\frac{(N-1)M}{n}$	$(N-1)Mt_c + (2n-1)\tau$
	nESBT	$\frac{(N-2)M}{n}$	$(N-1)Mt_c + 2n\tau$
	SBG	$\frac{(N-1)M}{n}$	$(N-1)Mt_c + (2n-1)\tau$
n-port	nRH	$\frac{M}{n}$	$\frac{(N-1)M}{n}t_c + (N-1)\tau$
	SBT	$\frac{NM}{\sqrt{2\pi(n-1)}}$	$\frac{NM}{2}t_c + n\tau$
	nRSBT	$\sqrt{\frac{2}{\pi}}\frac{NM}{n^{3/2}}$	$\frac{(N-1)M}{n}t_c + n\tau$
	nESBT	$\sqrt{\frac{2}{\pi}}\frac{NM}{n^{3/2}}$	$\frac{(N-1)M}{n}t_c + (n+1)\tau$
	SBG	$\sqrt{\frac{2}{\pi}}\frac{NM}{n^{3/2}}$	$\frac{(N-1)M}{n}t_c + n\tau$

Table 13: The optimum complexity of all-to-all broadcasting.

With *one-port* communication SBT routing for all-to-all broadcasting is optimum within a factor of two. The H, nESBT, nRSBT and SBG routings all have the minimum data transfer time. The number of start-ups is $(N-1)$, $2n$, $2n-1$ and $2n-1$, respectively, with a packet size of $\frac{M(N-2)}{n}$, $\frac{M(N-1)}{n}$ and $\frac{M(N-1)}{n}$, respectively. With a packet size of order $\frac{MN}{n}$, the number of start-ups for SBT, nESBT, nRSBT and SBG routings are all comparable. Note that if the packet size is smaller than the data set to be broadcast, then the SBT routing is of the same complexity as the H routing. But, if $B \geq \frac{1}{2}MN$ then $T_{min} = (N-1)Mt_c + n\tau$, which is optimum within a factor of 2. The start-up time is reduced by a factor of $\frac{N-1}{n}$ compared to the H routing.

With *n-port* communication the nRH, nESBT, nRSBT, and SBG routings achieve the lower bound transmission time for a sufficiently large packet size. Both the nRSBT and SBG routings also attain the minimum number of start-ups n with the same packet size $B \geq \sqrt{\frac{2}{\pi}}\frac{NM}{n^{3/2}}$. The nESBT routing has one more start-up. The nRH routing, though optimum for data transfer time, has $N-1$ start-ups. The SBT routing yields a slow-down of approximately $\frac{n}{2}$ for the data transfer time compared to the lower bound routing.

7 All-to-All Personalized Communication

7.1 Time Bounds

Theorem 9 *A lower bound for* one-port *all-to-all personalized communication is* $\max(\frac{nNM}{2}t_c, n\tau)$. *The packet size must be at least* $\frac{NM}{2}$ *to attain this lower bound.*

Proof: The bandwidth requirement for distributing personalized data from one node is

$$\sum_{i=0}^{n} i\binom{n}{i}M = \frac{nNM}{2}.$$

The total bandwidth requirement is $\frac{nN^2M}{2}$. During each cycle only N edges of the n-cube can communicate in the case of *one-port* communication; $\frac{nNM}{2}$ is the minimum number of element transfers in sequence. The number of start-ups is at least n. The maximum packet size can be derived by dividing the total bandwidth requirement $\frac{nN^2M}{2}$ by the number of cycles n, and the number of directed edges that can be used in each routing cycle N. ∎

Theorem 10 *A lower bound for* n-port *all-to-all personalized communication is* $\max(\frac{NM}{2}t_c, n\tau)$. *The packet size must be at least* $\frac{NM}{2n}$ *to attain this lower bound.*

Proof: From theorem 9 the total bandwidth requirement is $\frac{nN^2M}{2}$. During each routing cycle nN directed edges can communicate concurrently. The maximum packet size is derived by dividing the total bandwidth requirement by the number of cycles n and the total number of links nN. ∎

Theorem 11 *The time for* n-port *all-to-all personalized communication based on* $G^{id}(*)$ *of height* h *and* postorder *scheduling is*

$$T = \sum_{l=0}^{h-1} \left(Mt_c \times \max_{\forall d \in \mathcal{D}} v^{id}(d,l) + \left\lceil \frac{M}{B} \times \max_{\forall d \in \mathcal{D}} v^{id}(d,l) \right\rceil \tau \right).$$

If $B \geq \max_{0 \leq l \leq h-1,\ \forall d \in \mathcal{D}} (M \times v^{id}(d,l))$ *then*

$$T = \left(\sum_{l=0}^{h-1} \max_{\forall d \in \mathcal{D}} v^{id}(d,l) \right) Mt_c + h\tau.$$

Proof: For each $G^{id}(s)$, the total amount of data transmitted across all edges in dimension d during routing cycle l is $v^{id}(d,l) \times M$, $0 \leq l \leq h-1$. For $G^{id}(*) = \cup_{s \in \mathcal{N}} Tr(s, G^{id}(0))$, each *a-graph* edge is mapped to N distinct cube edges with N distinct exclusive-or operations on both end-points. The amount of data transmitted across each cube edge in dimension d during routing cycle l is $v^{id}(d,l) \times M$. ∎

Theorem 12 *The time for* n-port *all-to-all personalized communication based on* $G^{id}(*)$ *of height* h *and* reverse-breadth-first *scheduling is*

$$T = \sum_{l=0}^{h-1} \left(Mt_c \times \max_{\forall d \in \mathcal{D}} u^{id}(d,l) + \left\lceil \frac{M}{B} \times \max_{\forall d \in \mathcal{D}} u^{id}(d,l) \right\rceil \tau \right).$$

If $B \geq \max_{0 \leq l \leq h-1,\ \forall d \in \mathcal{D}} (M \times u^{id}(d,l))$ *then*

$$T = \left(\sum_{l=0}^{h-1} \max_{\forall d \in \mathcal{D}} u^{id}(d,l) \right) Mt_c + h\tau.$$

Proof: Similar to the proof of theorem 11. ∎

Theorem 13 *The all-to-all personalized communication based on* N *translated* o-graphs *will attain the lower bound for the data transfer time iff the* o-graph *is greedy.*

Proof: The bandwidth requirement for each node is

$$\sum_{l=0}^{h-1} \sum_{d=0}^{n-1} v^{id}(d,l) = \sum_{l=0}^{h-1} \sum_{d=0}^{n-1} u^{id}(d,l) = \sum_{l=1}^{h} l \times \text{ (the number of nodes at level } l\text{)}.$$

Hence, non-greedy *o-graphs* require more data transfer than greedy *o-graphs.* ∎

Theorem 14 *All-to-all personalized* n-port *communication based on* $G^{id}(*)$ *where* $G^{id}(0) = \cup_{d \in \mathcal{D}} Ro^d(T^{id'}(0))$ *and* $T^{id'}(0)$ *is greedy, can attain both the minimum data transmission time,* $\frac{NM}{2}t_c$, *and the minimum number of start-ups,* n, *for* $B \geq \frac{(N-1)M}{n}$ *both for* postorder *and* reverse-breadth-first *schedulings.*

Proof: From theorem 11 and corollary 3, the data transfer time is

$$\sum_{l=0}^{n-1}\sum_{i=l+1}^{n}\frac{M}{n}\binom{n}{i}t_c = \frac{NM}{2}t_c.$$

The packet size $\frac{(N-1)M}{n}$ occurs during routing cycle 0. Similarly, the *reverse-breadth-first* scheduling discipline can be shown to be optimum, and has the same value of the maximum packet size. It occurs during the last routing cycle. ∎

Corollary 10 All-to-all personalized *n-port* communication based on $G^{nRSBT}(*)$ and $G^{SBG}(*)$ can attain the time $\frac{NM}{2}t_c + n\tau$, which is within a factor of 2 of the lower bound.

Notice that in *one-port* communication the data transfer time is always optimal if the routing is based on N translated greedy *o-graphs* and appropriate scheduling. But, not all greedy *o-graphs* have the same number of start-ups. Only the SBT graph allows a minimum of n start-ups for sufficiently large packet size. The minimum number of start-ups can be decided by the same labeling rules as were used in all-to-all broadcasting. The minimum number of start-ups is the same as for the all-to-all broadcasting. The difference is that the amount of data transferred during cycle i is equal to the sum of weighted subtree sizes with the root of each subtree connected through an edge labeled i to its parent.

7.2 The Complexity of All-to-All Personalized Communication

Tables 14 and 15 summarize the complexity estimates.

Comm.	Routing	T
one-port	H	$\frac{(N-1)NM}{2}t_c + \sum_{i=1}^{N-1}\lceil\frac{iM}{B}\rceil\tau$
	SBT	$\frac{nNM}{2}t_c + \lceil\frac{NM}{2B}\rceil n\tau$
	nRSBT	$\frac{nNM}{2}t_c + (\sum_{i=0}^{n-1}\lceil\frac{(i+1)NM}{2nB}\rceil + \sum_{i=n}^{i=2n-2}\lceil\frac{(2n-i-1)NM}{2nB}\rceil)\tau$
	nESBT	$(\frac{nN}{2}+N-2)Mt_c + (\sum_{i=0}^{n-1}\lceil(\frac{(i+2)N}{2}-1)\frac{M}{nB}\rceil + \sum_{i=1}^{n}\lceil(\frac{iN}{2}-1)\frac{M}{nB}\rceil)\tau$
	SBG	$\approx \frac{nNM}{2}t_c + \max(2n-1, \frac{nNM}{2B})\tau$
n-port	nRH	$\frac{(N-1)NM}{2n}t_c + \sum_{i=1}^{N-1}\lceil\frac{iM}{nB}\rceil\tau$
	SBT	$(\sum_{l=0}^{\lfloor\frac{n-1}{2}\rfloor}\binom{2l}{l}2^{n-2l-1} + \sum_{l=\lfloor\frac{n+1}{2}\rfloor}^{n-1}\binom{n-1}{l})Mt_c$ $+(\sum_{l=0}^{\lfloor\frac{n-1}{2}\rfloor}\lceil\binom{2l}{l}\frac{2^{n-2l-1}M}{B}\rceil + \sum_{l=\lfloor\frac{n+1}{2}\rfloor}^{n-1}\lceil\binom{n-1}{l}\frac{M}{B}\rceil)\tau$
	nRSBT	$\frac{NM}{2}t_c + \lceil\frac{NM}{2nB}\rceil n\tau$
	nESBT	$(\frac{N}{2}+\frac{N-2}{n})Mt_c + (\sum_{i=2}^{n}\lceil\sum_{j=i}^{n}\binom{n}{j}\frac{M}{nB}\rceil + \lceil\frac{(N-2)M}{nB}\rceil + \lceil\frac{(N-1)M}{nB}\rceil)\tau$
	SBG	$\frac{NM}{2}t_c + \sum_{i=1}^{n}\lceil\sum_{j=i}^{n}\binom{n}{j}\frac{M}{nB}\rceil\tau$

Table 14: The complexity of all-to-all personalized communication.

With *one-port* communication and H routing both the data transfer time and the start-up times are off by a factor of $\frac{N}{n}$ compared to the optimum for *one-port* communication. The complexity for *n-port* communication and nRH routing holds for both the *postorder* and *reverse-breadth-first* schedulings. The routing fully utilizes the cube bandwidth; however, much of the data transfer is not through the shortest path, i.e., non-greedy.

Figure 13 shows all-to-all personalized communication on a 3-cube based on 8 SBT's. The shaded area represents the portion of the data residing in processor 0 (denoted P_0). The task

Comm.	Routing	B_{opt}	T_{min}
one-port	H	$(N-1)M$	$\frac{(N-1)NM}{2}t_c + (N-1)\tau$
	SBT	$\frac{NM}{2}$	$\frac{nNM}{2}t_c + n\tau$
	nRSBT	$\frac{NM}{2}$	$\frac{nNM}{2}t_c + (2n-1)\tau$
	nESBT	$\frac{M}{n}(\frac{N(n-1)}{2}-1)$	$(\frac{nN}{2}+N-2)Mt_c + 2n\tau$
	SBG	$\frac{NM}{2}$	$\frac{nNM}{2}t_c + (2n-1)\tau$
n-port	nRH	$\frac{(N-1)M}{n}$	$\frac{(N-1)NM}{2n}t_c + (N-1)\tau$
	SBT	$\frac{NM}{2}$	$O(\sqrt{n})NMt_c + n\tau$
	nRSBT	$\frac{NM}{2n}$	$\frac{NM}{2}t_c + n\tau$
	nESBT	$\frac{(N-1)M}{n}$	$(\frac{N}{2}+\frac{N-2}{n})Mt_c + (n+1)\tau$
	SBG	$\frac{(N-1)M}{n}$	$\frac{NM}{2}t_c + n\tau$

Table 15: The optimum complexity of all-to-all personalized communication.

is to exchange the j^{th} block of data of processor i with the i^{th} block of processor j for any two distinct processors i and j. If initially processor i owns the i^{th} block column as in figure 13-(1) then on completion of the all-to-all personalized communication processor i contains the i^{th} block row as in figure 13-(4).

Lemma 24 All-to-all personalized *one-port* communication based on $G^{SBT}(*)$ can be accomplished in time $\frac{nNM}{2}t_c + n\tau$, which is within a factor of 2 of the lower bound.
Proof: During the first routing cycle, $\frac{NM}{2}$ data is exchanged along the lowest dimension. Then, the procedure is applied recursively with the data set doubling for each cycle of the recursion and the dimension of the cube decreasing by 1. Let $T(i,M)$ be the time required by the stated personalized all-to-all routing algorithm with initially M data per node in an i-cube. Clearly, $T(n,M) = 2^{n-1}Mt_c + \tau + T(n-1,2M)$ and $T(1,M) = Mt_c + \tau$. Hence, $T(n,M) = \frac{nNM}{2}t_c + n\tau$. ∎

In the case of *n-port* communication we can find the communication complexity from the previously derived formula. The interesting quantity for *postorder* scheduling is $\sum_{l=0}^{n-1} \max_d v^{id}(d,l)$.

$$\sum_{l=0}^{n-1} \max_d v^{SBT}(d,l) = \sum_{l=0}^{\lfloor \frac{n-1}{2} \rfloor} \binom{2l}{l} 2^{n-2l-1} + \sum_{l=\lfloor \frac{n+1}{2} \rfloor}^{n-1} \binom{n-1}{l},$$

which is of $O(\sqrt{n}N)$. Since $u^{SBT}(d,l) = v^{SBT}(n-d-1,l)$, Table 3, the *reverse-breadth-first* scheduling yields the same result.

The maximum packet size for nRSBT communication can be reduced from $\frac{(N-1)M}{n}$ to $\frac{NM}{2n}$ by using a scheduling such that each SBT used in composing the nRSBT graph is the same as the scheduling for the SBT graph in *one-port one-to-all personalized communication*. The n SBT's are scheduled concurrently. Note that the same scheme, if applied to all-to-all broadcasting, will increase the maximum packet size from $\sqrt{\frac{2}{\pi}}\frac{NM}{n^{3/2}}$ to $\frac{NM}{2n}$. The time for nESBT communication is obtained from theorems 11 and 12.

7.3 Summary of All-to-All Personalized Communication

With *one-port* communication, the SBT, nRSBT, SBG and nESBT routings with *postorder* and *reverse-breadth-first* schedulings for all-to-all personalized communication are optimum within

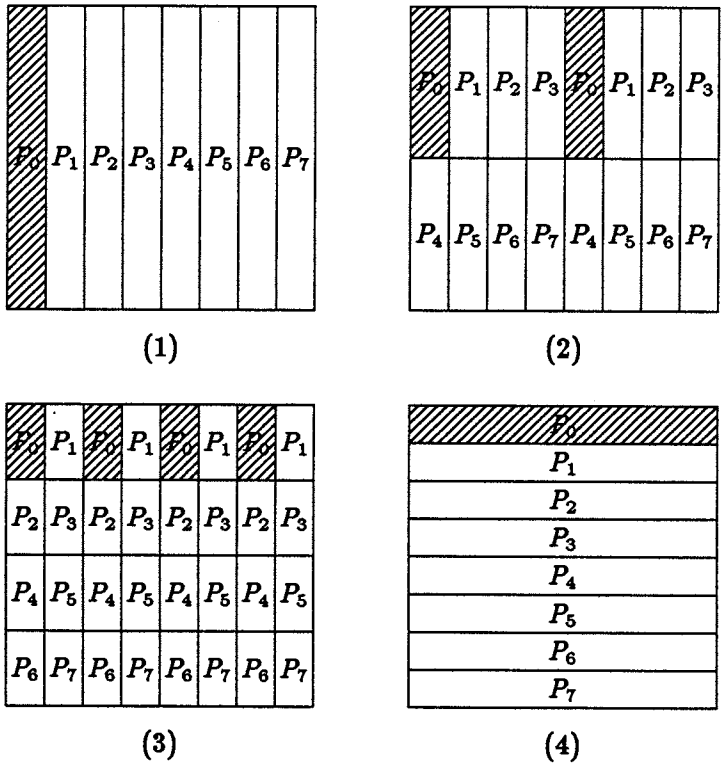

Figure 13: All-to-all personalized communication in a 3-cube based on SBT's.

constant factors, Table 15. The SBT has the lowest complexity with a sufficiently large packet size. Only greedy *o-graphs* attain the optimum data transfer time. In fact, the optimum data transfer time can always be attained for the greedy *o-graphs* with appropriate scheduling. The number of start-ups is equal to the maximum edge label plus one for a graph labeling such that edges in the same dimension carry the same label, and outgoing edges always carry a higher label than incoming edges for the spanning trees making up the *o-graph*. Both the nRSBT and the SBG routings attain the optimum data transfer. The nESBT and H routings have a data transfer time that exceed the lower bound by a factor of $\frac{n+2}{n}$, and $\frac{N}{n}$. The number of routing cycles for the SBT communication is at least n, for the nRSBT and SBG communications it is $2n-1$ and for the nESBT communication it is $2n$.

With *n-port* communication, the nESBT, nRSBT and SBG routings are optimum within a constant factor of ≈ 2 for *postorder* and *reverse-breadth-first* schedulings. Both the nESBT and nRH are non-greedy, but the extra distance of the edges from the root of the nESBT is constant. The SBT routing, though greedy, does not evenly utilize edges in the same dimension, and hence is non-optimum. The data transfer time for the nESBT, the SBT, and the nRH routings are a factor of $\frac{n+2}{n}$, $\sqrt{n}$, and $\frac{N}{n}$ higher than the optimum time.

8 Experimental Results

Some of the communication algorithms presented here have been implemented on an Intel iPSC [16] with 128 nodes, connected as a 7-dimensional Boolean cube. It has a message passing programming model. Up to $16k$ bytes, an *external packet*, can be passed in each communication,

however, the operating system subdivides messages into *internal packets* of size $1k$ bytes. There is a communication overhead (start-up time) associated with each packet. For an external packet we recorded a start-up time averaging $8ms$. For internal packets the start-up time was approximately $6ms$ (at the time our programs were tested). Interprocessor communication channels are $10M$-bit ethernet channels. Although there are 7 ports per processor, the storage bandwidth can only support $2-3$ ports concurrently. However, we have effectively been unable to realize this potential with the available operating system. The concurrency in communication on different ports of the same processor amounts to an overlap of about 20%.

For *one-to-all broadcasting* the communication time increases almost linearly for external packet sizes below $1k$ bytes, Figure 14. Figure 15 shows the measured time of the SBT and nESBT communications for an external packet size of $1k$ bytes and for cube dimensions ranging from 2 to 6. As predicted, measured speed-up is approximately n.

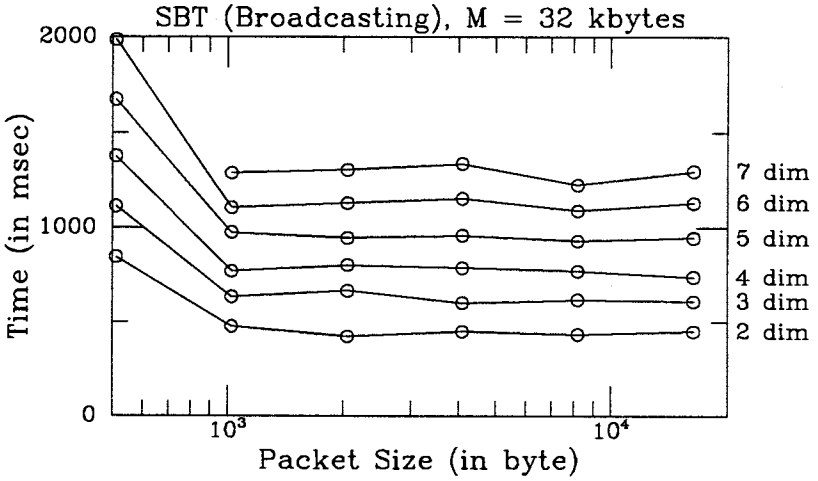

Figure 14: One-to-all broadcasting for the SBT routing.

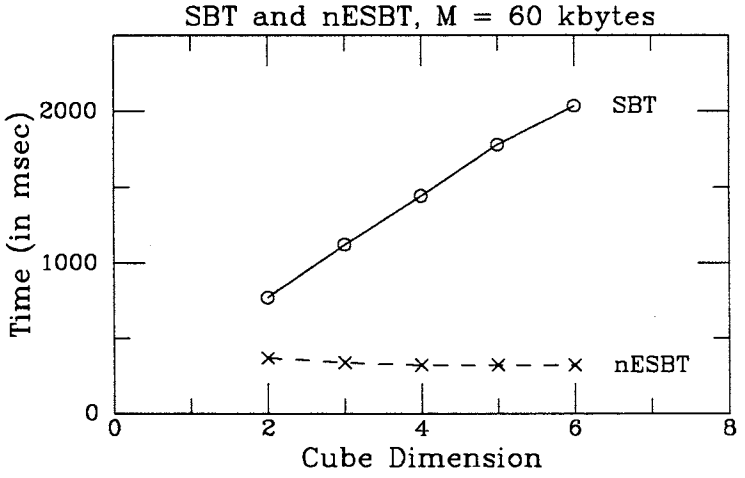

Figure 15: One-to-all broadcasting for the SBT and nESBT routings.

For *one-to-all personalized communication* based on SBT routing we schedule port communications in a *binary-reflected* Gray code order to take advantage of the partial overlap in communication on different ports. In the SBG routing, the root determines which node belongs to which subtree. If n is a prime number the subtrees are isomorphic (excluding node $(\bar{s}_{n-1}\bar{s}_{n-2}\ldots\bar{s}_0)$)

and the root only needs to keep one table of length $\approx \frac{N}{n}$ with each entry of size n bits. The order of the entries corresponds to the transmission order for each port. The table entries point to the messages transmitted over port 0. The pointers for the other ports are obtained by (right) cyclic shifts of the table entries. A one step cyclic rotation is used for port 1, two steps for port 2, etc. For n not a prime number there are also other cyclic nodes. The period $P(i)$ for each table entry needs to be found, and the message divided into $P(i)$ pieces. Our implementation uses a single path to every node (SBnT).

Internal nodes can either route according to the destination address if it is included, or use tables. If the destination is included, then a node first checks if it is the destination. Otherwise, the output port is determined by finding the base from $base^{SBnT}((myaddress) \oplus (source))$ and then finding the first bit that is equal to 1 in $((myaddress) \oplus (destination))$ to the left (cyclically) of the bit corresponding to the base. If tables are used instead of a destination field, then for *postorder* scheduling it suffices that each internal node keeps a count for each port. Since the number of ports used in each subtree is at most $n-3$ and the number of nodes in the entire subtree is approximately $\frac{N}{n}$, a bound on the table size in each node is n^2 bits. A *reverse-breadth-first* scheduling can be implemented by internal nodes keeping a table of how many nodes there are at a given level in each of its subtrees. The table has at most n^2 entries. An upper bound for the number of nodes in a subtree at any level is $\frac{N}{n^{3/2}}$, and the total table size in a node is at most n^3 bits. Hence, without a more sophisticated encoding the *postorder* scheduling discipline requires less table space. It is used for the measurements presented in Figure 16.

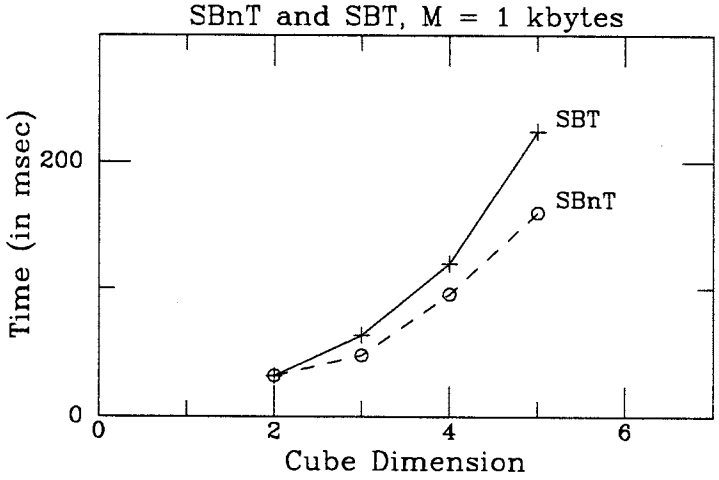

Figure 16: One-to-all personalized communication for the SBnT and SBT routings.

With *one-port* one-to-all personalized communication the expected time for SBT routing and SBG routing is the same for $B = M$. The observed advantage of the SBG over the SBT routing is due to the fact that the SBG can take better advantage of the overlap between communication on different ports. In the SBT case, even though messages were communicated over different ports in a binary-reflected Gray code order, the nodes adjacent to the root may not be finished with retransmitting the last packet received when a new packet arrives, in practice. In the SBG, a subtree receives a packet once every n cycles, and full advantage of the 20% overlap in communication actions is taken.

For *all-to-all broadcasting* the execution times are expected to be of the same order for all forms of all-to-all broadcasting on the Intel iPSC with $B = M$. The results of implementing the SBT, SBG and two H routings are shown in Figure 17.

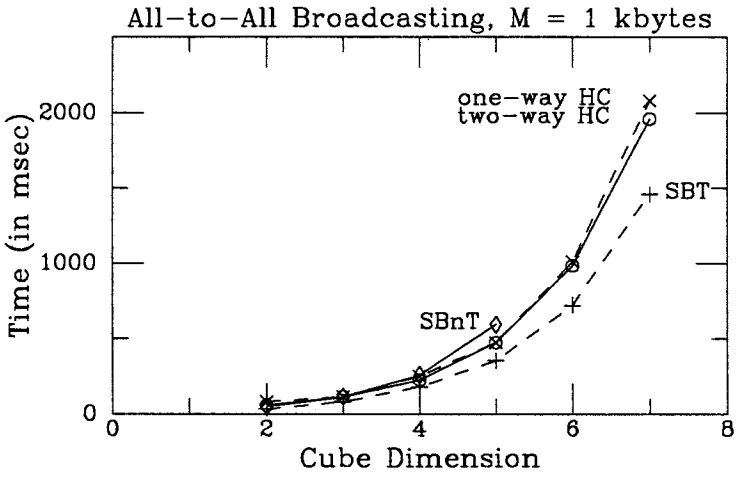

Figure 17: All-to-all broadcasting on the Intel iPSC.

We have also implemented the ***all-to-all personalized communication*** based on the SBT graph on the iPSC. The result is presented in [22].

9 Summary and Conclusions

We have presented three new communication graphs for Boolean cubes and defined scheduling disciplines for: (1) one-to-all broadcasting; (2) one-to-all personalized communication; (3) all-to-all broadcasting; and (4) all-to-all personalized communication, so that the communication tasks are completed within a small constant factor of the best known lower bounds. For each case we considered two communication models: communication restricted to one-port at-a-time for each processor; and concurrent communication on all ports of every processor. One of the new communication graphs consists of n edge-disjoint spanning binomial trees. Another is a balanced tree. Each subtree of the root of the balanced n-tree we define has approximately $\frac{N}{n}$ nodes.

For communication restricted to ***one-port at-a-time***, the nESBT routing is optimum for case 1 within a factor of 4 of the best known lower bound, a speed-up of up to n over the SBT routing. For cases 2, 3, and 4, the SBT routings with appropriate scheduling are shown to be optimum within a factor of 2. The nESBT, nRSBT, and SBG routings are all optimum within a factor of 4. For ***concurrent communication on all ports***, the nESBT routing is optimum within a factor of 4 for case 1, a speed-up of up to n over the SBT routing. For cases 2, 3, and 4, the nRSBT and SBG routings are optimum within a factor of 2. The nESBT routing is optimum within a factor of ≈ 2. The speed-up of the data transmission time for the three routings over the SBT routing is $\frac{n}{2}$, $\frac{n}{2}$, and $O(\sqrt{n})$ for cases 2, 3, and 4 respectively. Table 16 summarizes the results.[1]

The SBG routing has the additional property that the order of the time complexity holds for any data volume, while this is not true for the routings using n paths to every node, such as the nESBT and nRSBT routings. The nESBT routing offers n edge-disjoint paths between the source and any destination node, and hence inherently has a good degree of fault tolerance with respect to communication links.

[1]For the nESBT routing, T_{min} is valid only if $1 \leq B_{opt} \leq M$ for *one-port* and $1 \leq B_{opt} \leq \frac{M}{n}$ for *n-port* communication.

Routing for the four communication operations can also be based on Two-rooted Complete Binary Trees (TCBT) [3,6]. Communication algorithms based on the TCBT may yield performance comparable to the algorithms presented here for *one-port* communication. For some such scheduling algorithms see [24].

The packet size is very important for the communication complexity. With *one-port* communication and a packet size less or equal to the data set to be communicated to every node, all considered routings have approximately the same complexity for one-to-all personalized communication and all-to-all broadcasting.

Experimental results on the Intel iPSC/d7 confirm the timing model and complexity analysis. The generic communications have been applied to matrix multiplication [21], matrix transposition [22], and tridiagonal systems solvers [23].

Comm.	Data distribution	Assumption	Routing	B_{opt}	T_{min}	Factor
one-port	One-to-all Broadcasting	$M = 1$	SBT	1	$n(t_c + \tau)$	1
		$M > 1$	nESBT	$\sqrt{\frac{M\tau}{nt_c}}$	$(\sqrt{Mt_c} + \sqrt{n\tau})^2$	4
	One-to-all P.C.	$M \geq 1$	SBT	$\frac{NM}{2}$	$(N-1)Mt_c + n\tau$	2
	All-to-all B.	$M \geq 1$	SBT	$\frac{NM}{2}$	$(N-1)Mt_c + n\tau$	2
	All-to-all P.C.	$M \geq 1$	SBT	$\frac{NM}{2}$	$\frac{nNM}{2}t_c + n\tau$	2
n-port	One-to-all Broadcasting	$M \leq n$	nRSBT	1	$n(t_c + \tau)$	1
		$M > n$	nESBT	$\frac{1}{n}\sqrt{\frac{M\tau}{t_c}}$	$(\sqrt{\frac{Mt_c}{n}} + \sqrt{n\tau})^2$	4
	One-to-all Personalized comm.	$M \geq n$	nESBT	$\sqrt{\frac{2}{\pi}\frac{NM}{n^{3/2}}}$	$\frac{(N-1)M}{n}t_c + (n+1)\tau$	$\frac{2(n+1)}{n}$
		$M \geq 1, M \geq n$	SBG, nRSBT	$\sqrt{\frac{2}{\pi}\frac{NM}{n^{3/2}}}$	$\frac{(N-1)M}{n}t_c + n\tau$	2
	All-to-all Broadcasting	$M \geq n$	nESBT	$\sqrt{\frac{2}{\pi}\frac{NM}{n^{3/2}}}$	$\frac{(N-1)M}{n}t_c + (n+1)\tau$	$\frac{2(n+1)}{n}$
		$M \geq 1, M \geq n$	SBG, nRSBT	$\sqrt{\frac{2}{\pi}\frac{NM}{n^{3/2}}}$	$\frac{(N-1)M}{n}t_c + n\tau$	2
	All-to-all Personalized comm..	$M \geq n$	nESBT	$\frac{(N-1)M}{n}$	$(\frac{N}{2} + \frac{N-2}{n})Mt_c + (n+1)\tau$	$\frac{2(n+2)}{n}$
		$M \geq 1$	SBG	$\frac{(N-1)M}{n}$	$\frac{NM}{2}t_c + n\tau$	2
		$M \geq n$	nRSBT	$\frac{NM}{2n}$	$\frac{NM}{2}t_c + n\tau$	2

Table 16: Time complexity of communication algorithms. The last column shows the constant factors as compared to the best known lower bounds.

Acknowledgement

The authors express their gratitude in particular to referee D for making many helpful observations and suggestions that significantly improved the presentation. The generous support of the Office of Naval Research under contract N00014-84-K-0043 is gratefully acknowledged.

References

[1] Alfred V. Aho, John E. Hopcroft, and Jeffrey D. Ullman. *The Design and Analysis of Computer Algorithms.* Addison-Wesley, 1974.

[2] Sandeep N. Bhatt, F.R.K. Chung, F. Tom Leighton, and Arnold L. Rosenberg. Optimal simulations of tree machines. In *Proc. 27th IEEE Symp. Foundations Comput. Sci.*, pages 274–282, IEEE Computer Society, 1986.

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

[4] Sally A. Browning. *The Tree Machine: A Highly Concurrent Computing Environment.* Technical Report 1980:TR:3760, Computer Science, California Institute of Technology, January 1980.

[5] Eliezer Dekel, David Nassimi, and Sartaj Sahni. Parallel matrix and graph algorithms. *SIAM J. Computing*, 10:657–673, 1981.

[6] Sanjay R. Deshpande and Roy M. Jenevin. Scaleability of a binary tree on a hypercube. In *International Conference on Parallel Processing*, pages 661–668, IEEE Computer Society, 1986.

[7] Michael J. Fischer. *Efficiency of Equivalence Algorithms*, pages 153–167. Plenum Press, 1972.

[8] Geoffrey C. Fox and Wojtek Furmanski. *Optimal Communication Algorithms on Hypercube.* Technical Report CCCP-314, California Institute of Technology, July 1986.

[9] Geoffrey C. Fox and D. Jefferson. *Concurrent Processor Load Balancing as a Statistical Physics Problem.* Technical Report CCCP-172, California Institute of Technology, May 1985.

[10] Dennis Gannon and John Van Rosendale. On the impact of communication complexity in the design of parallel numerical algorithms. *IEEE Trans. Computers*, C-33(12):1180–1194, December 1984.

[11] John L. Gustafson, Stuart Hawkinson, and Ken Scott. The architecture of a homogeneous vector supercomputer. In *1986 Int. Conf. Parallel Processing*, pages 649–652, IEEE Computer Society, 1986.

[12] John P. Hayes, Trevor N. Mudge, Quentin F. Stout, Stephen Colley, and John Palmer. Architecture of a hypercube supercomputer. In *1986 Int. Conf. Parallel Processing*, pages 653–660, IEEE Computer Society, 1986.

[13] W. Daniel Hillis. *The Connection Machine.* MIT Press, 1985.

[14] Ching-Tien Ho and S. Lennart Johnsson. Distributed routing algorithms for broadcasting and personalized communication in hypercubes. In *1986 Int. Conf. Parallel Processing*, pages 640–648, IEEE Computer Society, 1986. Tech. report YALEU/DCS/RR-483, May 1986.

[15] Ching-Tien Ho and S. Lennart Johnsson. *Spanning Balanced Trees in Boolean cubes.* Technical Report YALEU/DCS/RR-508, Yale University, Dept. of Computer Science, January 1987.

[16] *Intel iPSC System Overview.* Intel Corp., January 1986.

[17] S. Lennart Johnsson. Communication efficient basic linear algebra computations on hypercube architectures. *Journal of Parallel and Distributed Computing*, 4(2):133–172, April 1987. (Report YALEU/DCS/RR-361, January 1985).

[18] S. Lennart Johnsson. Ensemble architectures and their algorithms: an overview. In Martin H. Schultz, editor, *Proceedings of the IMA Workshop on Numerical Algorithms for Parallel Computer Architectures*, Springer Verlag, 1987. YALE/DCS/RR-580. Revision of YALE/DCS/RR-367, February 1985.

[19] S. Lennart Johnsson. *Odd-Even Cyclic Reduction on Ensemble Architectures and the Solution Tridiagonal Systems of Equations.* Technical Report YALE/DCS/RR-339, Department of Computer Science, Yale University, October 1984.

[20] S. Lennart Johnsson. Solving tridiagonal systems on ensemble architectures. *SIAM J. Sci. Stat. Comp.*, 8(3):354–392, May 1987. (Report YALEU/DCS/RR-436, November 1985).

[21] S. Lennart Johnsson and Ching-Tien Ho. *Algorithms for Multiplying Matrices of Arbitrary Shapes Using Shared Memory Primitives on a Boolean Cube.* Technical Report YALEU/DCS/RR-569, Department of Computer Science, Yale University, October 1987. Revision of YALE/DCS/RR-530. Presented at the ARMY Workshop on Medium Scale Parallel Processors, Stanford University, January 1986.

[22] S. Lennart Johnsson and Ching-Tien Ho. Matrix transposition on Boolean n-cube configured ensemble architectures. *SIAM J. on Algebraic and Discrete Methods*, . To appear. YALE/DCS/RR-572. (Revised edition of YALEU/DCS/RR-494, November 1986.).

[23] S. Lennart Johnsson and Ching-Tien Ho. *Multiple tridiagonal systems, the Alternating Direction Method, and Boolean cube configured multiprocessors.* Technical Report YALEU/DCS/RR-532, Yale University, June 1987.

[24] S. Lennart Johnsson and Ching-Tien Ho. *Spanning Graphs for Optimum Broadcasting and Personalized Communication in Hypercubes.* Technical Report YALEU/DCS/RR-500, Yale University, Dept. of Computer Science, November 1986.

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

[26] Oliver A. McBryan and Eric F. Van de Velde. Hypercube algorithms and implementations. *SIAM J. Scientific and Statistical Computing*, 8(2):s227–s287, March 1987.

[27] Edward M. Reingold, Jurg Nievergelt, and Narsingh Deo. *Combinatorial Algorithms.* Prentice Hall, 1977.

[28] Yousef Saad and Martin H. Schultz. *Data Communication in Hypercubes.* Technical Report YALEU/DCS/RR-428, Dept. of Computer Science, Yale University, October 1985.

[29] Yousef Saad and Martin H. Schultz. *Topological properties of Hypercubes.* Technical Report YALEU/DCS/RR-389, Dept. of Computer Science, Yale University, June 1985.

[30] Charles L. Seitz. The cosmic cube. *Communications of the ACM*, 28(1):22–33, 1985.

[31] Quentin F. Stout and Bruce Wager. Passing messages in link-bound hypercubes. In *The 1986 Hypercube Conference*, SIAM, 1987.

[32] Leslie Valiant and G.J. Brebner. Universal schemes for parallel communication. In *Proc. of the 13th ACM Symposium on the Theory of Computation*, pages 263–277, ACM, 1981.

[33] Angela Y. Wu. Embedding of tree networks in hypercubes. *Journal of Parallel and Distributed Computing*, 2(3):238–249, 1985.