Technical Report No. 94-362

# OPTIMAL COMMUNICATION PRIMITIVES ON THE GENERALIZED HYPERCUBE NETWORK

by
Paraskevi Fragopoulou, Selim G. Akl, and Henk Meijer

Department of Computing & Information Science
Queen's University
Kingston, Ontario, Canada

June 1994

This research was supported by the Telecommunications Research Institute of Ontario and by the Natural Sciences and Engineering Research Council of Canada.

**Abstract**

Efficient interprocessor communication is crucial to increasing the performance of parallel computers. In this paper, a special framework is developed on the *generalized hypercube*, a network that is currently receiving considerable attention. Using this framework as the basic tool, a number of spanning graphs with special properties to fit various communication needs, are constructed on the network. The importance of these spanning graphs is demonstrated with the development of optimal algorithms for four fundamental communication problems, namely, the *single node* and *multinode broadcasting* and the *single node* and *multinode scattering*, on the generalized hypercube network. *Broadcasting* is the distribution of the same group of messages from a source processor to all other processors, and *scattering* is the distribution of distinct groups of messages from a source processor to each other processor. We consider broadcasting and scattering from a single processor of the network (single node broadcasting and scattering) and simultaneously from all processors of the network (multinode broadcasting and scattering). For the multinode broadcasting and scattering algorithms a special technique is developed on the generalized hypercube so that messages originating at individual nodes are interleaved in such a manner that no two messages contend for the same edge at any given time. The communication problems are studied under the *all-port* communication assumption, meaning that in one time step a processor can exchange messages of fixed length with all of its neighbors simultaneously. Under this assumption the full bandwidth of the communication network is used. Lower bounds are derived for the above problems under the stated assumptions, in terms of time and number of message transmissions, and optimal algorithms are designed.

**Key words and phrases:** communication algorithm, interconnection network, generalized hypercube, parallel algorithm, spanning tree.

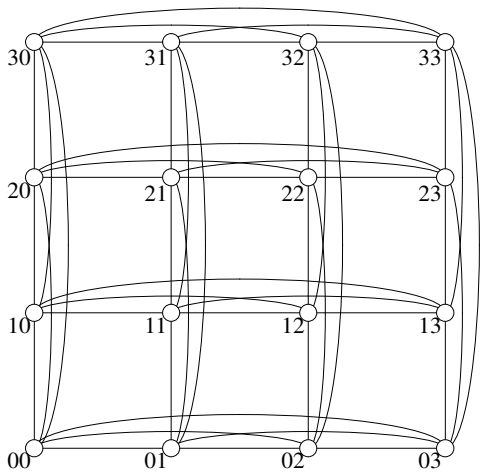


Figure 1: The $GH_{2,4}$ network.

# 1 Introduction

It is widely recognized that interprocessor communication is one of the main obstacles in increasing the performance of parallel computers in which the processors are linked by an interconnection network. The communication problems emerging from a wide range of parallel algorithms are not arbitrary but define regular communication primitives. It is crucial for the high performance of parallel computers to efficiently execute these primitives. In this paper, we concentrate on four fundamental communication primitives, namely the *single node* and *multinode broadcasting*, and the *single node* and *multinode scattering*, on the popular *generalized hypercube* network. These appear in problems such as matrix operations (e.g. matrix-vector and matrix-matrix multiplication, factorization, inversion, transposition), solutions of systems of equations (e.g. Gaussian elimination), image manipulation (e.g. histogramming), some database operations (e.g. polling, master-slave operations) etc. *Broadcasting* is the distribution of the same group of messages from a source processor to all other processors, and *scattering* is the distribution of distinct groups of messages from a source processor to all other processors. We consider broadcasting and scattering from a single source processor of the network (single node broadcasting and scattering) and simultaneously from all processors of the network (multinode broadcasting and scattering). The cases where a source node wishes to transmit one or more than one messages are distinguished.

The interconnection network under consideration is the *generalized hypercube* network, which has been proven to be a flexible topology for the interconnection of processors [5, 14]. An $n$-dimensional, $k$-ary generalized hypercube, denoted by $GH_{n,k}$, has $N = k^n$ processors, each one labeled by an $n$-digit number in radix $k$ arithmetic. Two processors are connected if their labels differ in exactly one digit, i.e. processor $v_{n-1}...v_{i+1}v_iv_{i-1}...v_0$ is connected to processors $v_{n-1}...v_{i+1}v_i'v_{i-1}...v_0$ for every $0 \leq i \leq n-1$, $0 \leq v_i' \leq k-1$ and $v_i \neq v_i'$. It can be easily observed that the network is a generalization of the popular binary hypercube. The $n$-digit binary numbers that represent nodes of the binary hypercube are replaced by $n$-digit $k$-ary numbers to represent nodes of the generalized hypercube. The $GH_{2,4}$ network can be seen in Fig. 1.

All of the communication problems are studied under the *all-port* assumption (as opposed to the *one-port* assumption), meaning that in one time step a processor can exchange messages of fixed length with all of its neighbors simultaneously. This assumption is used in several recently constructed multiprocessors in order to use all of the available bandwidth. As pointed out by several authors [21], if at each time step a processor can exchange messages of fixed length with only one of its neighbors, i.e. if the communication is based on the one-port assumption, the used bandwidth of any network topology is the same as the bandwidth of a ring with the same number of processors. The algorithms are derived for the *store-and-forward* communication model, i.e. a processor must receive the entire message before it can process it and retransmit it. The communication is *bidirectional*, meaning that an edge can be used for message transmission in both directions at each time step and can be viewed as two unidirectional edges. Each message requires unit time to be transmitted on an edge, i.e. the *unit* cost model is assumed.

A common approach to implement communication algorithms on interconnection networks is to embed spanning trees with special properties on those networks. The root of the tree is usually the origin or the destination of the information, while the edges are used to direct the transmission of messages from parent to children processors or vice-versa. All of the algorithms presented in this paper are based on the construction of spanning trees with special properties and the use of appropriate scheduling disciplines to achieve optimal results. A special framework is developed to facilitate the construction of the spanning trees and the design of the communication algorithms. The main results obtained in this paper for the preceding communication problems and when each source processor wishes to transmit $M$ messages to each one of its destination processors, are summarized in table 1. The number of messages $M$ is usually assumed to be large.

| Problem | Time steps | Message transmissions |
|---|---|---|
| Single node broadcasting | $M+n-1$ | $M(k^n-1)$ |
| Multinode broadcasting | $\lceil \frac{M(k^n-1)}{n(k-1)} \rceil$ | $M(k^n-1)k^n$ |
| Single node scattering | $\lceil \frac{M(k^n-1)}{n(k-1)} \rceil$ | $Mn(k-1)k^{n-1}$ |
| Multinode scattering | $Mk^{n-1}$ | $Mn(k-1)k^{2n-1}$ |

Table 1: The main results obtained on the $GH_{n,k}$ network.

The first column gives the number of time steps required for each algorithm to complete and the second column gives the number of message transmissions performed. We will show that each of these numbers is equal to a lower bound for the problem, except the number of time steps required for the single node broadcasting which is only asymptotically optimal. When each source processor wishes to transmit only one message to each of its destination processors, the number of time steps required is also asymptotically optimal. The multinode broadcasting and scattering problems are of special interest. A special technique is developed on the generalized hypercube (lemma 5) so that messages originating at individual nodes are interleaved in such a manner that no two messages contend for the same edge at any time during the execution of an algorithm. This technique demonstrates that the utilization of all communication edges of a network simultaneously is possible, and that efficient algorithmic techniques that take advantage of this capability can be developed. In the single node scattering problem, where the edges incident to the source

node constitute a bottleneck for the transmission of the messages, the spanning graphs offer the capability to transmit an equal number of messages over each edge incident to the source node, and as a consequence optimal number of time steps is achieved. The same algorithms can be used to derive optimal solutions for the single node and multinode reduction over an associative operator, and for the single node and multinode gathering problems by inversing the transmission of the messages.

A survey on adaptive communication algorithms on the generalized hypercube network can be found in [14]. The method of spanning graph construction has been previously used to design communication algorithms on other interconnection networks, such as the binary hypercube [16], the multidimensional torus [10], and the star network [6, 7].

The remainder of this paper is organized as follows. Notations and definitions that are used throughout the paper are introduced in section 2. Sections 3 and 4 present the construction of a spanning tree and a spanning graph, respectively, on the generalized hypercube network. Section 5 is devoted to the derivation of lower bounds and the design of optimal algorithms based on the spanning graphs, for all of the communication problems under consideration. Finally, we conclude in section 6 along with a summary of the results obtained in this paper and some suggestions for further research.

# 2 Notations and definitions

An $n$-dimensional $k$-ary generalized hypercube $GH_{n,k}$, is an undirected graph of $k^n$ nodes, each one labeled by an $n$-digit number in radix $k$ arithmetic. Each node $v$ is connected to $n(k-1)$ other nodes with which it differs in only one digit, i.e. $v = v_{n-1}...v_{i+1}v_iv_{i-1}...v_0$ is connected to $v' = v_{n-1}...v_{i+1}v'_iv_{i-1}...v_0$ for all $0 \leq i \leq n-1$, $0 \leq v'_i \leq k-1$ and $v_i \neq v'_i$, Fig. 1. The network is edge and node symmetric with degree $n(k-1)$ (number of edges at each node) and diameter $n$ (maximum shortest distance between any pair of nodes). $GH_{n,k}$ belongs to the class of Cayley graphs [2, 17]. For networks in this class, nodes correspond to the elements of a finite group and edges correspond to a set of generators that act on the elements of the group [2]. In this context, the $k(n-1)$ generators that define the edges of $GH_{n,k}$ are denoted by $g_i^j$, $0 \leq i \leq n-1$, $1 \leq j \leq k-1$. Generator $g_i^j$ connets node $v = v_{n-1}...v_{i+1}v_iv_{i-1}...v_0$ to node $v' = v_{n-1}...v_{i+1}((v_i + j) \bmod k)v_{i-1}...v_0$, which results by adding $j \bmod k$ to the $i^{th}$ digit of $v$. In this case we say that edge $(v, v')$ is of dimension $g_i^j$, or $dim(v, v') = g_i^j$. Thus, each node of $GH_{n,k}$ is connected to $n(k-1)$ other nodes through dimensions $g_i^j$, $0 \leq i \leq n-1$, $1 \leq j \leq k-1$. In what follows, node 00...0 of $GH_{n,k}$ is referred to as node 0. It can be easily observed that the network is a generalization of the popular binary hypercube. The binary hypercube contains pairs of connected nodes in each dimension, while the generalized hypercube contains a complete subnetwork of $k$ nodes in each dimension.

We now define an operation on nodes of the generalized hypercube network, namely the *translation* operation, that will be of primary importance for the construction of the spanning graphs and the description of the communication algorithms. Having a spanning graph rooted at node 0 of $GH_{n,k}$, we will derive an isomorphic spanning graph, with the same properties, rooted at any other node $s$ of $GH_{n,k}$, using a translation of the graph rooted at node 0 with respect to $s$. As a consequence, it is sufficient to construct a spanning graph rooted at node 0 of $GH_{n,k}$. The translation operation on $GH_{n,k}$ is analogous to the exclusive-OR operation on nodes of the binary hypercube [16, 3, 4].

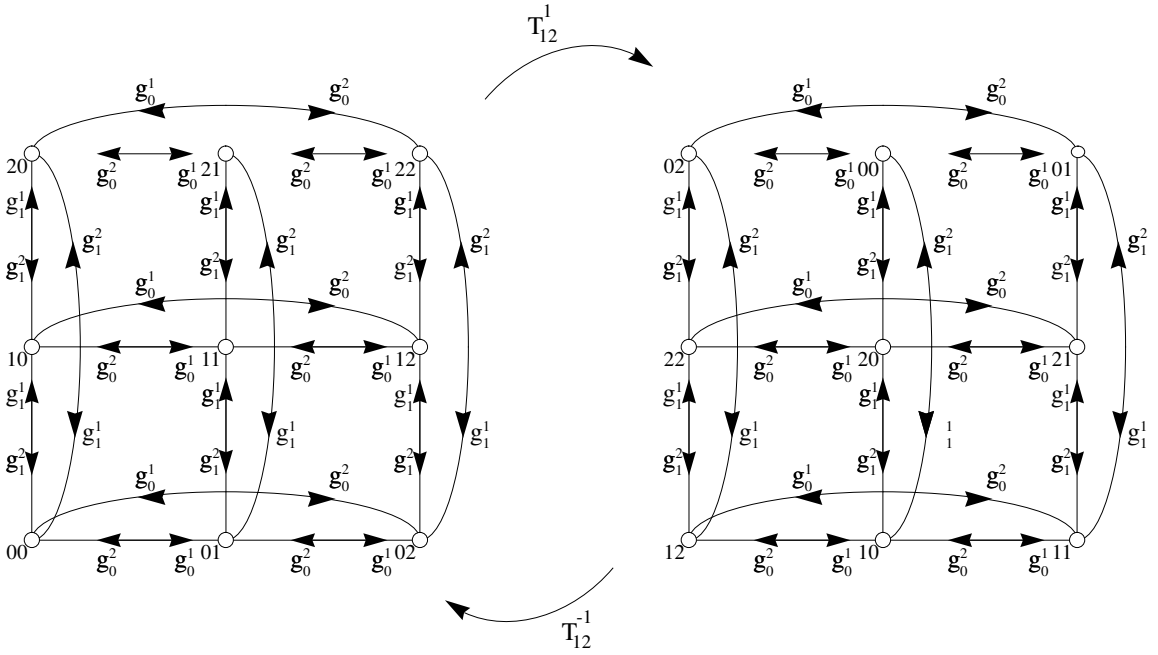


Figure 2: Application of the $T_{12}$ operation on the $GH_{2,3}$ network.

**Definition 1:** The *translation* of a node $v$ with respect to node $s$, denoted by $T_s(v)$, is defined to be node $t = T_s(v)$, so that $t_i = (v_i + s_i) \bmod k$, $0 \leq i \leq n-1$. The *inverse translation* of a node $v$ with respect to node $s$, denoted by $T_s^{-1}(v)$, is defined to be node $t = T_s^{-1}(v)$, so that $t_i = (v_i - s_i) \bmod k$, $0 \leq i \leq n-1$. By translation of a network with respect to $s$ we mean that each node of the network is translated with respect to $s$. For example, for nodes $v = 231$ and $s = 132$ of $GH_{3,4}$, $T_s(v) = 323$ and $T_s^{-1}(v) = 103$.

**Lemma 1:** The translation operation preserves the dimension of each edge. If edge $(v, u)$ is of dimension $g_i^j$, then edge $(T_s(v), T_s(u))$ is also of dimension $g_i^j$.

**Proof:** Assume that edge $(v, u)$ has dimension $g_i^j$, $0 \leq i \leq n-1$, $1 \leq j \leq k-1$. This means that $v$ and $u$ differ only in their $i^{th}$ digit by $j \bmod k$. From the definition of translation with respect to node $s$, it is easily derived that $T_s(v)$ and $T_s(u)$ also differ only in their $i^{th}$ digit by $j \bmod k$ and as a consequence edge $(T_s(v), T_s(u))$ is of dimension $g_i^j$. □

The translation operation is an automorphism on the generalized hypercube that preserves the topology of the network and the dimension of each edge. An example of the application of the $T_{12}$ operation on the $GH_{2,3}$ network is shown in Fig. 2. The property of edge dimension preservation is apparent.

We now define another operation on $GH_{n,k}$, namely the rotation operation, that will also be of primary importance for the construction of the spanning trees, and for the development of the multinode broadcasting and scattering algorithms. As emphasized in the introduction, these algorithms are designed so that messages originating at individual nodes are interleaved in such a manner that no two messages contend for the same edge at any given time. The properties of the rotation operation, as explained below, will help achieve this attribute. The rotation operation on nodes of the generalized hypercube has properties similar to those of the left cyclic shift operation on nodes of the binary hypercube [16, 3, 4].

**Definition 2:** Consider function $r$ from the set $\{0, 1, ..., k-1\}$ to itself:

$$r(i) = \begin{cases} 0, & \text{if } i = 0, \\ i \bmod (k-1) + 1, & \text{otherwise.} \end{cases}$$

(notice that $r$ maps digit 0 to itself and the remaining digits as follows: $1 \rightarrow 2 \rightarrow 3 \rightarrow ... \rightarrow k-1 \rightarrow 1$). The *rotation* of a node $v = v_{n-1}...v_{i+1}v_iv_{i-1}...v_0$, denoted by $R(v)$, is defined to be node $v_{n-2}...v_{i+1}v_iv_{i-1}...v_0r(v_{n-1})$. This can be viewed as a left cyclic shift of the digits of $v$ with the wraparound digit being mapped though function $r$. By rotation of a network we mean that the rotation operation is applied to each node of the network. By $R^i = R \circ R^{i-1}$ we denote $i$ applications of rotation. For example, for nodes $v = 221$ and $u = 012$ of $GH_{3,4}$, $R(v) = 213$ and $R(u) = 120$.

**Lemma 2:** The rotation operation has the following properties:

1. If $(v, u)$ is an edge of dimension $g_i^j$, $0 \leq i \leq n-1$, $1 \leq j \leq k-1$, so that $v_i = 0$ and $u_i = j$, then edge $(R(v), R(u))$ is an edge of dimension $g_{i'}^{j'}$ so that:

$$i' = (i+1) \bmod n,$$

$$j' = \begin{cases} r(j), & \text{if } i = n-1, \\ j, & \text{otherwise.} \end{cases}$$

2. The rotation operation preserves the distance of each node from node 0.

**Proof:** We prove each property separately.

1. Assume that $0 \leq i \leq n-2$. If we express $(v, u)$ and $(R(v), R(u))$ as follows:

$$\begin{array}{ll} (v_{n-1}...v_{i+1}\underline{0}v_{i-1}...v_0, & v_{n-1}...v_{i+1}\underline{j}v_{i-1}...v_0), \\ (v_{n-2}...v_{i+1}\underline{0}v_{i-1}...v_0r(v_{n-1}), & v_{n-2}...v_{i+1}\underline{j}v_{i-1}...v_0r(v_{n-1})), \end{array}$$

   it is clear that if $v$ and $u$ differ in their $i^{th}$, $0 \leq i \leq n-2$, digit by $j$ then $R(v)$ and $R(u)$ differ in their $(i+1)^{st}$ digit also by $j$.

   Assume that $i = n-1$. If we express $(v, u)$ and $(R(v), R(u))$ as follows:

$$\begin{array}{ll} (\underline{0}v_{n-2}...v_i...v_0, & \underline{j}v_{n-2}...v_i...v_0), \\ (v_{n-2}...v_i...v_0\underline{0}, & v_{n-2}...v_i...v_0\underline{r(j)}), \end{array}$$

   it is clear that if $v$ and $u$ differ in their $(n-1)^{st}$ digit by $j$ then $R(v)$ and $R(u)$ differ in their $0^{th}$ digit by $r(j)$.

2. The distance of a node $v$ from node 0 is equal to the number of nonzero digits in the label of $v$. This is a modified definition of the Hamming distance for nodes of the generalized hypercube. Since the rotation operation preserves the number of nonzero digits in the label of a node (bijection $r$ maps digit 0 to itself), it also preserves the distance of each node from node 0. Another way to see this is the following. The rotation operation is an automorphism on $GH_{n,k}$ that maps node 0 to itself. As an extension to this, nodes obtained as rotations of each other are all at the same distance from node 0.□

**Lemma 3:** If $(v, u)$ is a directed edge of dimension $g_i^j$, so that $v_i = 0$ and $u_i = j$, then the $n(k-1)$ directed edges derived from $(v, u)$ by consecutive applications of the rotation operation are all of different dimensions.

**Proof:** This is derived in a straightforward manner from the first part of lemma 2, which describes the impact of the rotation operation on the dimension of an edge of this type. For example, for edge $(01, 11)$ of $GH_{2,4}$, the $n(k-1) = 6$ edges produced by consecutive applications of the rotation operation are $(01, 11) \xrightarrow{R} (10, 12) \xrightarrow{R} (02, 22) \xrightarrow{R} (20, 23) \xrightarrow{R} (03, 33) \xrightarrow{R} (30, 31)$ and their corresponding dimensions are $g_1^1 \xrightarrow{R} g_0^2 \xrightarrow{R} g_1^2 \xrightarrow{R} g_0^3 \xrightarrow{R} g_1^3 \xrightarrow{R} g_0^1$. □

To summarize, the translation and the rotation operations are automorphisms on $GH_{n,k}$ that preserve the distance between its nodes. The translation operation preserves the dimension of each edge (lemma 1), while the rotation operation alters it in a regular fashion (lemma 2). Finally, the topology of $GH_{n,k}$ or one of its subnetworks remains unchanged under translation or rotation.

The nodes of $GH_{n,k}$ are grouped into equivalence classes under the operation of rotation as follows:

**Definition 3:** An ordered group of nodes, each one derived from its subsequent one cyclically, by the application of a rotation is called a *necklace*.

The term necklace was initially used for groups of nodes of the shuffle exchange graph [18].

**Lemma 4:** Necklaces have the following properties:

1. A necklace contains *at most* $n(k-1)$ nodes.
2. The size of a necklace always divides $n(k-1)$.
3. All nodes of a necklace are at the same distance from node 0.

**Proof:** We prove each property separately.

1. From the definition of rotation it can be verified that $R^{n(k-1)}(v) = v$ for every node $v$ of $GH_{n,k}$. However, we say *at most* $n(k-1)$ rotations because the same node can emerge after less than $n(k-1)$ rotations. For example, for node $v = 21$ of $GH_{2,4}$, $R^3(v) = 21$ and the same node emerges after only 3 rotations and not $n(k-1) = 6$.
2. The proof for this property can be found in group theory. A rotation operation is an automorphism on $GH_{n,k}$ of order $n(k-1)$. A necklace is an orbit under the action of rotation. The size of an orbit always divides the order of the automorphism [13, 19].
3. This property is derived in a straightforward manner from the property of distance preservation of the rotation operation (lemma 2). □

In what follows a *full necklace* is a necklace that contains $n(k-1)$ distinct nodes. A *nonfull necklace* is a necklace that contains less than $n(k-1)$ nodes. A node belongs to a nonfull necklace if its label has a nontrivial symmetry with respect to the rotation operation. Nodes that belong to nonfull necklaces constist of a substring of $\frac{n}{m}$ digits $v_{\frac{n}{m}-1}...v_1v_0$ ($m$ is a divisor of $n$), which is repeated $m$ times with its nonzero digits modified as follows: $r^{m-1}(v_{\frac{n}{m}-1})...r^{m-1}(v_1)r^{m-1}(v_0)...r^i(v_{\frac{n}{m}-1})...r^i(v_1)r^i(v_0)...r(v_{\frac{n}{m}-1})...r(v_1)r(v_0)v_{\frac{n}{m}-1}...v_1v_0$ [9]. For example, node 330220 of $GH_{6,4}$ which belongs to a nonfull necklace that contains 12 nodes constists of the substring 220 of three digits which is repeated twice as follows: $r(2)r(2)r(0)220 = 330220$.

From the properties of the rotation operation we conclude that the nodes of $GH_{n,k}$ at each distance from node 0 are collections of necklaces. In table 2, the necklaces of $GH_{2,4}$, and those of $GH_{3,3}$ at each distance $d$, $0 \leq d \leq n$, from node 0 are given enclosed in parentheses.

The necklaces of $GH_{2,4}$

$d = 0:$ ( $\overset{0}{\underline{00}}$ )

$d = 1:$ ( $\overset{0}{\underline{30}}$, $\overset{1}{03}$, $\overset{2}{20}$, $\overset{3}{02}$, $\overset{4}{10}$, $\overset{5}{01}$ )

$d = 2:$ ( $\overset{0}{\underline{33}}$, $\overset{1}{23}$, $\overset{2}{22}$, $\overset{3}{12}$, $\overset{4}{11}$, $\overset{5}{31}$ )

( $\overset{0}{\underline{32}}$, $\overset{1}{13}$, $\overset{2}{21}$ )

The necklaces of $GH_{3,3}$

$d = 0:$ ( $\overset{0}{\underline{000}}$ )

$d = 1:$ ( $\overset{0}{\underline{200}}$, $\overset{1}{020}$, $\overset{2}{002}$, $\overset{3}{100}$, $\overset{4}{010}$, $\overset{5}{001}$ )

$d = 2:$ ( $\overset{0}{\underline{220}}$, $\overset{1}{022}$, $\overset{2}{102}$, $\overset{3}{110}$, $\overset{4}{011}$, $\overset{5}{201}$ )

( $\overset{0}{\underline{210}}$, $\overset{1}{021}$, $\overset{2}{202}$, $\overset{3}{120}$, $\overset{4}{012}$, $\overset{5}{101}$ )

$d = 3:$ ( $\overset{0}{\underline{222}}$, $\overset{1}{122}$, $\overset{2}{112}$, $\overset{3}{111}$, $\overset{4}{211}$, $\overset{5}{221}$ )

( $\overset{0}{\underline{212}}$, $\overset{1}{121}$ )

Table 2: The necklaces of $GH_{2,4}$ and $GH_{3,3}$.

The following definition aims to distinguish one particular node of each necklace.

**Definition 4:** The binary correspondent of a node $v$ of $GH_{n,k}$ is the binary number obtained if we substitute each nonzero digit in $v$ with digit 1. The *generator node* of a necklace is defined to be the largest among the nodes of the necklace that have the largest binary correspondent.

For example, for necklace ( 210, 021, 202, 120, 012, 101 ) of $GH_{3,3}$ the generator node is 210 because this is the largest from nodes 210, 120, that have the largest binary correspondent among the nodes of the necklace.

**Definition 5:** The *displacement* of a node $v$, denoted by $D(v)$, is defined to be the minimum number of rotation operations required to derive from this node the generator node of the necklace to which it belongs.

**Definition 6:** The *period* of a node $v$, denoted by $P(v)$, is defined to be the number of nodes contained in the necklace to which it belongs.

In table 2, the generator node of each necklace is underlined and the displacement of each node is marked on top of its label.

**Definition 7:** An *unfolded necklace* is an ordered group of exactly $n(k-1)$ nodes, not necessarily distinct, each one obtained from its subsequent one cyclically, by the application of a rotation.

Each necklace has a corresponding unfolded necklace. For full necklaces, the corresponding unfolded necklace is the necklace itself. For nonfull necklaces that contain $P$ nodes, the corresponding unfolded necklace is the necklace repeated $\frac{k(n-1)}{P}$ times. This is possible since the size of a necklace is always a divisor of $k(n-1)$ (lemma 4). In table 3, the unfolded necklaces of $GH_{2,4}$, and those of $GH_{3,3}$ are given. A comparison with table 2 will help clarify the difference between a necklace and its corresponding unfolded necklace.

| The unfolded necklaces of $GH_{2,4}$ | | The unfolded necklaces of $GH_{3,3}$ | |
|---|---|---|---|
| $d=0:$ | ( 00, 00, 00, 00, 00, 00 ) | $d=0:$ | ( 000, 000, 000, 000, 000, 000 ) |
| $d=1:$ | ( 30, 03, 20, 02, 10, 01 ) | $d=1:$ | ( 200, 020, 002, 100, 010, 001 ) |
| $d=2:$ | ( 33, 23, 22, 12, 11, 31 ) | $d=2:$ | ( 220, 022, 102, 110, 011, 201 ) |
| | ( 32, 13, 21, 32, 13, 21 ) | | ( 210, 021, 202, 120, 012, 101 ) |
| | | $d=3:$ | ( 222, 122, 112, 111, 211, 221 ) |
| | | | ( 212, 121, 212, 121, 212, 121 ) |

Table 3: The unfolded necklaces of $GH_{2,3}$ and $GH_{3,3}$.

The property of the rotation operation that $n(k-1)$ directed edges each of which is obtained as a rotation of its preceding one are all of different dimensions (lemma 3), along with the property of edge dimension preservation of the translation operation (lemma 1) will be used extensively in the development of the multinode broadcasting and scattering algorithms. These properties will help guarantee that messages originating at individual nodes will be interleaved in such a manner that no two messages will contend for the same edge at any given time. Below we explain how this attribute can be achieved.

In a multinode broadcasting or scattering algorithm, all nodes of the network are source of messages. Under the all-port communication model $n(k-1)k^n$ directed edges are available on $GH_{n,k}$ for message transmission at each time step. Messages originating at each one of the $k^n$ nodes of $GH_{n,k}$ are transmitted through at most $n(k-1)$ directed edges at each time step. Let us denote by $E_i(0)$ the set of $n(k-1)$ directed edges on which messages originating at node 0 are transmitted at time step $i$ of the algorithm. Since a multinode algorithm proceeds symmetricly from each node of the network, the $n(k-1)$ directed edges on which messages originating at node $s$ are transmitted at time step $i$, denoted by $E_i(s)$, is obtained from $E_i(0)$ using the operation of translation with respect to $s$ (if $(v,u) \in E_i(0)$ then $(T_s(v), T_s(u)) \in E_i(s)$). The following lemma is enough to guarantee that no conflicts arise during the execution of an algorithm.

**Lemma 5:** At each time step $i$, if the $n(k-1)$ directed edges in $E_i(0)$ are all of different dimensions, then the sets of $n(k-1)$ directed edges $E_i(s)$, where $s$ ranges over all nodes of $GH_{n,k}$, are disjoint.

**Proof:** Assume two different edges $(v,u) \neq (v',u')$ of $E_i(0)$ for some $i$, and take edges $(T_s(v), T_s(u)) \in E_i(s)$ and $(T_{s'}(v'), T_{s'}(u')) \in E_i(s')$, which are obtained by $(v,u)$ and $(v',u')$ respectively, under translation with respect to two different nodes of $GH_{n,k}$, $s$ and $s'$. Also assume that $(T_s(v), T_s(u)) = (T_{s'}(v'), T_{s'}(u'))$. From the property of preservation of the dimension of each edge under translation we conclude that $dim(v,u) = dim(T_s(v), T_s(u)) = dim(T_{s'}(v'), T_{s'}(u')) = dim(v',u')$, which contradicts our assumption that $(v,u)$ and $(v',u')$ are two different edges of $E_i(0)$. □

The multinode broadcasting and scattering algorithms will be developed so that at each time step $i$, the set $E_i(0)$ contains $n(k-1)$ directed edges that are rotations of each other and as a consequence of different dimensions. According to lemma 5, this will guarantee that at each time step $i$, the sets of $n(k-1)$ directed edges $E_i(s)$, where $s$ ranges over all nodes of $GH_{n,k}$, are disjoint and as a consequence no two messages will compete for the same edge at any time step $i$ during the execution of the algorithm.

We are now ready to proceed to the construction of the spanning graphs which will be the basic tools

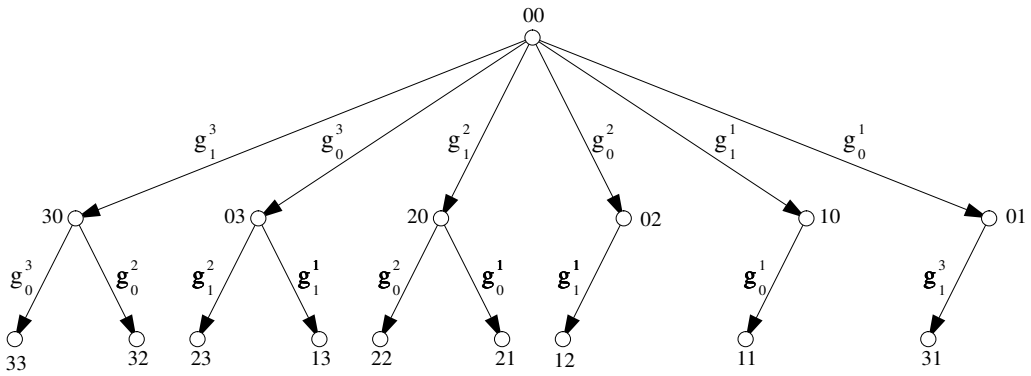


Figure 3: The $BST_0$ spanning tree on the $GH_{2,4}$ network.

for the development of the communication algorithms. We start by constructing a shortest path, balanced to within a constant factor spanning tree using the framework defined in this subsection. Subsequently, we extend the spanning tree to a shortest path spanning graph.

# 3 Spanning tree construction

We define a shortest path, balanced to within a constant factor spanning tree, rooted at node 0 of $GH_{n,k}$, and denoted by $BST_0$. The spanning tree is balance to within a constant factor, meaning that the ratio in the number of nodes between the largest and the smallest of the $n(k-1)$ subtrees of the root is less than a constant. The framework developed in the previous subsection will be the basic tool for the construction of the spanning tree with the stated properties. Each one of the $n(k-1)$ subtrees of $BST_0$ contains all nodes that have the same displacement. The $i^{th}$, $0 \leq i < n(k-1)$, subtree of $BST_0$ is defined to be the subtree that contains all nodes $v$ of $GH_{n,k}$ with displacement $D(v) = i$. Furthermore, an isomorphic spanning tree rooted at any other node $s$ of $GH_{n,k}$, and denoted by $BST_s$, can be easily derived from $BST_0$ using the operation of translation with respect to $s$. We are now ready to proceed to a formal definition of $BST_0$.

**Definition 8:** A shortest path spanning tree, balanced to within a constant factor, rooted at node 0 of $GH_{n,k}$, and denoted by $BST_0$, is defined through the following parent function. For node $v$, let $p_i$ be the position of its first nonzero digit cyclically to the left of position $(n-1-i) \bmod n$.

$$\text{parent}^{BST_0}(v) = \begin{cases} \emptyset, & \text{if } v = 0, \\ v_{n-1}...v_{p_i+1}0v_{p_i-1}...v_0, & \text{if } v \neq 0 \text{ and } D(v) = i. \end{cases}$$

The $BST_0$ spanning tree on the $GH_{2,4}$ network can be seen in Fig. 3.

**Lemma 6:** $BST_0$ has the following properties:

1. The $\text{parent}^{BST_0}$ function defines a shortest path spanning tree rooted at node 0 of $GH_{n,k}$.
2. The $i^{th}$, $0 \leq i < n(k-1)$, subtree of $BST_0$ contains all nodes $v$ of $GH_{n,k}$ with displacement $D(v) = i$. Furthermore, if node $v$ with displacement $D(v) = i$, $0 < i < n(k-1)$, has parent node $u$ in the $i^{th}$ subtree of $BST_0$ then node $R(v)$ with displacement $i-1$ has parent node $R(u)$ in the $(i-1)^{st}$ subtree of $BST_0$. From this we conclude that corresponding nodes of the subtrees of $BST_0$ form necklaces.

3. $BST_0$ is balanced to within a constant factor.

4. All nodes that belong to nonfull necklaces, except node 0, are leaf nodes.

**Proof:** We prove each property separately.

1. From the definition of the parent$^{BST_0}$ function we notice that the parent of each node is obtained by changing a nonzero digit in the label of the node to zero. As a consequence, the parent$^{BST_0}(v)$ function generates a shortest path to node 0. Furthermore, each node has only one parent node, hence the graph is a spanning tree.

2. The parent of node $v$ with displacement $D(v) = i$ in the $i^{th}$, $0 < i < n(k-1)$, subtree of $BST_0$ is obtained by changing to zero its first nonzero digit cyclically to the left of position $(n-1-i) \bmod n$. Assume that the digit that is changed into zero is in position $p$ of $v$ and that its parent is node $u$. The parent of node $R(v)$ with displacement $D(v) = i-1$ in the $(i-1)^{st}$ subtree of $BST_0$ is obtained by changing to zero its first nonzero digit cyclically to the left of position $(n-1-(i-1)) \bmod n$. The digit that is changed into zero in $R(v)$ is in position $(p+1) \bmod n$ and its parent is node $R(u)$ [9]. For example, node 103302 of $GH_{6,4}$ has dispacement 2, and its parent node in the second subtree of $BST_0$ is obtained by changing to zero its first nonzero digit cyclically to the left of position 3. As a consequence its parent is node 003302. Node $R(103302) = 033022$ has dispacement 1 and its parent in the first subtree of $BST_0$ is obtained by changing to zero its first nonzero digit cyclically to the left of position 4. As a consequence its parent is node 033020. It is true that $R(003302) = 033020$.

3. We must prove that each subtree of the root contains $O(\frac{k^n}{n(k-1)})$ nodes. From the definition of $BST_0$, the $i^{th}$, $0 \leq i < n(k-1)$, subtree contains all nodes of $GH_{n,k}$ with displacement $D(v) = i$. From the $n(k-1)$ nodes that belong to a full necklace, each one belongs to a different subtree. Nodes that create the imbalance among the subtrees are the ones that belong to nonfull necklaces. We now derive an upper bound for the number of nodes that belong to nonfull necklaces. As explained in section 2, these nodes consist of a substring of $n/m$ digits, which is repeated $m$ times with its nonzero digits modified. So for $m$ prime divisor of $n$ (all the other divisors of $n$ are included in this case) an estimate for the number of nodes that belong to nonfull necklaces is:

$$\sum_{m \geq 2,\ m|n}^{n} k^{n/m} = O(\sqrt{k^n}).$$

   As a consequence, each subtree contains at least $\frac{k^n}{n(k-1)} - O(\frac{\sqrt{k^n}}{n(k-1)}) = O(\frac{k^n}{n(k-1)})$ nodes. This upper bound is not tight and the imbalance among the subtrees is in reality much smaller. From table 4 we notice that the ratio between the number of nodes of the largest subtree of $GH_{n,k}$ and $\frac{k^n}{n(k-1)}$ rapidly converges to 1 as the number of nodes increases.

4. Nodes that are at maximum distance from node 0, do not contain any zero digits in their labels, and are always leaf nodes, since $BST_0$ is a shortest path tree. We have to prove that nodes of nonfull necklaces that contain zero digits in their labels are leaf nodes as well. Without loss of generality we prove that the generator node $v$ of a nonfull necklace is a leaf node of $BST_0$. Node $v$ has displacement

| $n$ | $k$ | Number of nodes $k^n$ | Number of nodes of nonfull necklaces | Number of necklaces | Size of minimum subtree | Size of maximum subtree | $k^n/n(k-1)$ | Ratio |
|---|---|---|---|---|---|---|---|---|
| 4 | 3 | 81 | 1 | 11 | 10 | 10 | 10.00 | 1.00 |
| 4 | 4 | 256 | 16 | 24 | 20 | 23 | 21.25 | 1.08 |
| 4 | 5 | 625 | 1 | 40 | 39 | 39 | 39.00 | 1.00 |
| 4 | 6 | 1296 | 36 | 68 | 63 | 67 | 64.75 | 1.03 |
| 4 | 7 | 2401 | 1 | 101 | 100 | 100 | 100.00 | 1.00 |
| 5 | 3 | 243 | 3 | 26 | 24 | 25 | 24.20 | 1.03 |
| 5 | 4 | 1024 | 4 | 70 | 68 | 69 | 68.20 | 1.01 |
| 5 | 5 | 3125 | 5 | 158 | 156 | 157 | 156.20 | 1.01 |
| 5 | 6 | 7776 | 1 | 312 | 311 | 311 | 311.00 | 1.00 |
| 5 | 7 | 16807 | 7 | 562 | 560 | 561 | 560.20 | 1.00 |
| 6 | 3 | 729 | 9 | 63 | 60 | 62 | 60.67 | 1.02 |
| 6 | 4 | 4096 | 64 | 232 | 224 | 231 | 227.50 | 1.02 |
| 6 | 5 | 15625 | 25 | 654 | 650 | 653 | 651.00 | 1.00 |
| 6 | 6 | 46656 | 246 | 1566 | 1547 | 1565 | 1555.17 | 1.01 |
| 6 | 7 | 117649 | 1 | 3269 | 3268 | 3268 | 3268.00 | 1.00 |
| 7 | 3 | 2187 | 3 | 158 | 156 | 157 | 156.14 | 1.01 |
| 7 | 4 | 16384 | 4 | 782 | 780 | 781 | 780.14 | 1.00 |
| 7 | 5 | 78125 | 5 | 2792 | 2790 | 2791 | 2790.14 | 1.00 |
| 7 | 6 | 279936 | 6 | 8000 | 7998 | 7999 | 7998.14 | 1.00 |
| 7 | 7 | 823543 | 7 | 19610 | 19608 | 19609 | 19608.14 | 1.00 |
| 8 | 3 | 6561 | 1 | 411 | 410 | 410 | 410.00 | 1.00 |
| 8 | 4 | 65536 | 256 | 2744 | 2720 | 2743 | 2730.62 | 1.00 |
| 8 | 5 | 390625 | 1 | 12208 | 12207 | 12207 | 12207.00 | 1.00 |
| 8 | 6 | 1679616 | 1296 | 42026 | 41958 | 42025 | 41990.38 | 1.00 |
| 8 | 7 | 5764801 | 1 | 120101 | 120100 | 120100 | 120100.00 | 1.00 |
| 9 | 3 | 19683 | 27 | 1098 | 1092 | 1097 | 1093.44 | 1.00 |
| 9 | 4 | 262144 | 1 | 9710 | 9709 | 9709 | 9709.00 | 1.00 |
| 9 | 5 | 1953125 | 125 | 54262 | 54250 | 54261 | 54253.45 | 1.00 |
| 9 | 6 | 10077696 | 216 | 223960 | 223944 | 223959 | 223948.78 | 1.00 |

Table 4: Comparison between the smallest and the largest subtrees of $BST_0$ for sample values of $n$ and $k$.

zero, belongs to the first subtree of $BST_0$, and consists of a substring of $n/m$ digits, which is repeated $m$ times with its nonzero digits modified. If node $v$ had a child node then one of its final zero digits, which belongs to the last substring of $m$ digits, becomes nonzero in the label of the child node. However the resulting node does not have displacement zero, it does not belong to the first subtree of $BST_0$, and as a consequence it cannot be a child node of $v$. For example, node 330220 of $GH_{6,4}$ is a generator node of a nonfull necklace that contains 12 nodes. If this node had a child node, then its last zero digit would change to a nonzero digit in its child node, i.e. 330221, 330222, or 330223. However from definition 4, none of these nodes could be the generator node of the necklace it belong to, since it is not the node with the largest binary correspondent among the nodes of its necklace (definition 4). □

The properties of $BST_0$ are apparent in Fig. 3. A simple comparison with table 2 will help verify that corresponding nodes of the subtrees of $BST_0$ form necklaces.

Using the $BST_0$ spanning tree and the parent$^{BST_0}$ function we can easily derive a $BST_s$, rooted at any other node $s$ of $GH_{n,k}$. This spanning tree is isomorphic to $BST_0$ and has the same properties as it. To derive $BST_s$, we simply apply the operation of translation with respect to $s$, on $BST_0$. If edge $(v, u)$ belongs to the $i^{th}$ subtree of $BST_0$, then edge $(T_s(v), T_s(u))$ belongs to the $i^{th}$ subtree of $BST_s$. Since the dimension of each edge is preserved under translation, these edges are of the same dimension. Each node $v$ can derive its parent in $BST_s$ by computing node parent$^{BST_0}(T_s^{-1}(v))$ and obtaining the translation of the resulting node with respect to $s$.

# 4 Spanning graph construction

The definition of $BST_0$ is extended to a spanning graph, rooted at node 0 of $GH_{n,k}$, and denoted by $BSG_0$. This is a special type of graph which is composed of $n(k-1)$ spanning trees, rooted at the nodes adjacent to node 0. The $i^{th}$, $0 \leq i < n(k-1)$, spanning tree of $BSG_0$ contains nodes $v$, for which $D(v) = i \bmod P(v)$. All spanning trees of $BSG_0$ are isomorphic, and each one can be derived from its next one, cyclically, by the application of a rotation operation. Furthermore, an isomorphic spanning graph rooted at any other node $s$ of $GH_{n,k}$, and denoted by $BSG_s$, can be easily derived from $BSG_0$ using the operation of translation with respect to $s$. We are now ready to proceed to a formal definition of $BSG_0$.

**Definition 9:** A shortest path spanning graph, rooted at node 0 of $GH_{n,k}$, and denoted by $BSG_0$, is defined though the following parent function. By parent$^{BSG_0}(v, i)$ we denote the parent of node $v$ in the $i^{th}$, $0 \leq i < n(k-1)$, spanning tree of $BSG_0$. For node $v$, let $p_i$ be the position of its first nonzero digit cyclically to the left of position $(n-1-i) \bmod n$.

$$\text{parent}^{BSG_0}(v,i) = \begin{cases} \emptyset, & \text{if } v = 0, \\ v_{n-1}...v_{p_i+1}0v_{p_i-1}...v_0, & \text{if } v \neq 0 \text{ and } D(v) = i \bmod P(v). \end{cases}$$

The $BSG_0$ spanning graph on the $GH_{2,4}$ network can be seen in Fig. 3.

**Lemma 7:** $BSG_0$ has the following properties:

1. The parent$^{BSG_0}$ function defines a shortest path graph rooted at node 0 of $GH_{n,k}$. Nodes that belong to full necklaces have a single path to node 0 through $BSG_0$. Nodes with period $P$ that belong to nonfull necklaces have $\frac{n(k-1)}{P}$ paths to node 0 through $BSG_0$.

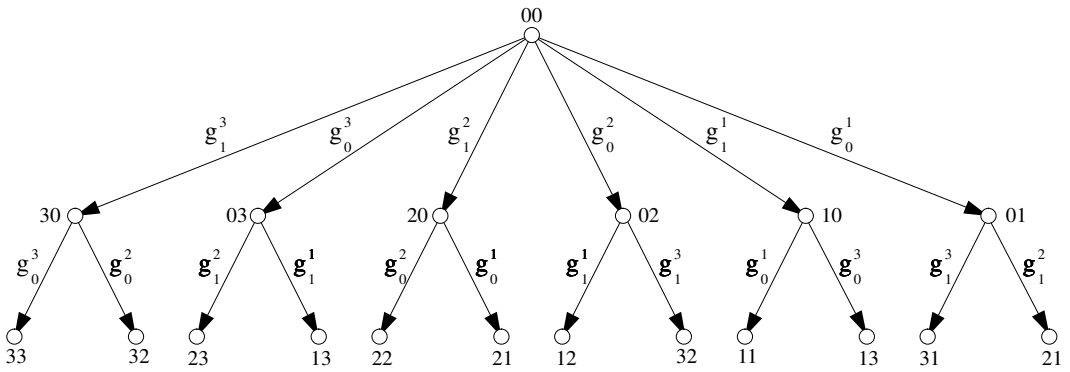


Figure 4: The $BSG_0$ spanning graph on the $GH_{2,4}$ network.

2. The $i^{th}$, $0 \leq i < n(k-1)-1$, spanning tree of $BSG_0$ can be obtained as a rotation of its $(i+1)^{st}$ spanning tree. Since rotation is an automorphism on $GH_{n,k}$ all spanning trees of $BSG_0$ are isomorphic. Furthermore, $n(k-1)$ corresponding directed edges of the spanning trees are rotations of each other and as a consequence of different dimensions (lemma 3), and $n(k-1)$ corresponding nodes of the spanning trees form an unfolded necklace (definition 7).

3. All nodes that belong to nonfull necklaces are leaf nodes.

**Proof:** We prove each property separately.

1. From the definition of the parent$^{BSG_0}$ function we notice that the parent(s) of each node is(are) obtained by changing a nonzero digit in the label of the node to zero. As a consequence the parent$^{BSG_0}$ function generates shortest paths from each node to node 0. From the definition of $BSG_0$ we now that the $i^{th}$, $0 \leq i < n(k-1)$, spanning tree of $GH_{n,k}$ includes all nodes $v$ for which $D(v) = i \bmod P(v)$. A node $v$ that belongs to a full necklace has period $P(v) = n(k-1)$ and as a consequence belongs only to spanning tree $T_{D(v)}$. A node $v$ with period P, that belongs to a nonfull necklace belongs to $\frac{n(k-1)}{P}$ spanning trees of $BSG_0$.

2. It is enough to prove that if node $v$ with displacement $D(v) = i \bmod P(v)$, has parent node $u$ in the $i^{th}$, $0 < i < n(k-1)$, spanning tree of $BSG_0$, then node $R(v)$ with displacement $D(v)-1 = (i-1) \bmod P(v)$, has parent node $R(u)$ in the $(i-1)^{st}$ spanning tree of $BSG_0$ [9]. The proof for this property is similar to the proof of the second part of lemma 6. From the above and using induction, it can be shown that each spanning tree can be obtained as a rotation of its next one in $BST_0$. From this we conclude that $n(k-1)$ corresponding directed edges of the spanning trees are rotations of each other and as a consequence of different dimensions (lemma 3). Furthermore, $n(k-1)$ corresponding nodes of the spanning trees are rotations of each other and as a consequence form an unfolded necklace (definition 7).

3. Since $BSG_0$ is an extension of $BST_0$, the proof of this property for $BSG_0$ is derived in a similar manner to the proof of the same property of $BST_0$. □

The properties of $BSG_0$ are apparent in Fig. 4. A simple comparison with table 3 will help verify that

corresponding nodes of the spanning trees of $BSG_0$ form unfolded necklaces. Corresponding edges of the spanning trees of $BSG_0$ are rotations of each other and as a consequence of different dimensions (lemma 3).

Using the $BSG_0$ graph and the parent$^{BSG_0}$ function we can easily derive a $BSG_s$, rooted at any other node $s$ of $GH_{n,k}$. This graph is isomorphic to $BSG_0$ and has the same properties as it. To derive $BSG_s$, we simply apply the operation of translation with respect to $s$, on $BSG_0$. If edge $(v,u)$ belongs to the $i^{th}$ spanning tree of $BSG_0$, then edge $(T_s(v),T_s(u))$ belongs to the $i^{th}$ spanning tree of $BSG_s$. Since the dimension of each edge is preserved under translation, these edges are of the same dimension. Each node $v$ can derive its parent in the $i^{th}$ spanning tree of $BSG_0$ by computing node parent$^{BSG_0}(T_s^{-1}(v),i)$ and obtaining the translation of the resulting node with respect to $s$.

The importance of the $BSG_s$ graph lies in several different properties it possesses. The fact that each of the $n(k-1)$ spanning trees of $BSG_s$ contain the same number of nodes is used in the single node and multinode scattering algorithms in order for each source node to transmit an equal number of its messages over each one of its incident edges. A node that belongs to a number of different spanning trees of $BSG_s$ receives an equal part of its messages from $s$ through the edges of each spanning tree. Furthermore, as mentioned in section 2, messages originating at individual nodes in a multinode broadcasting or scattering algorithm will be interleaved in such a manner that no two messages contend for the same edge at any time during the execution of the algorithm. A necessary condition in order to achieve this attribute was presented in lemma 5. Recall that by $E_i(s)$ we denote the set of $n(k-1)$ directed edges on which messages originating at node $s$ are transmitted at time step $i$ of a multinode broadcasting or scattering algorithm. Since a multinode algorithm proceeds symmetricly from all nodes of the network, each $E_i(s)$ is obtained from $E_i(0)$ by a translation with respect to $s$. According to lemma 5, if the $n(k-1)$ directed edges in $E_i(0)$ are all of different dimensions, then the sets of $n(k-1)$ directed edges $E_i(s)$, for fixed $i$ (time step), and $s$ ranging over all nodes of $GH_{n,k}$, are disjoint. In other words, at each time step $i$, messages originating at individual nodes are transmitted through different edges of $GH_{n,k}$. By lemma 7, the $n(k-1)$ spanning trees of $BSG_0$ are rotations of each other, and as a consequence $n(k-1)$ corresponding directed edges of the spanning trees of $BSG_0$ are all of different dimensions. This property is true for any $BSG_s$ graph, since the dimension of each edge is preserved under translation. We conclude that in order to avoid conflicts of messages originating at individual nodes during a multinode broadcasting or scattering algorithm, it is enough to use $n(k-1)$ corresponding directed edges of the spanning trees of $BSG_0$. Finally, the fact that $BSG_0$ is a shortest path graph offers the potential to achieve the lower bound for the number of message transmissions required for each communication problem.

# 5 Communication Algorithms

## 5.1 Lower bounds

In a single node broadcasting problem on $GH_{n,k}$, each of the $k^n-1$ destination nodes receives $M$ messages from the source node and a lower bound for the number of message transmissions is $M(k^n-1)$. The source node has $n$ node disjoint paths, of length at most $n+2$, to each one of the other nodes. In order to achieve the minimum number of time steps for this problem, the $M$ messages are split into $n(k-1)$ groups, each one containing $\lceil \frac{M}{n(k-1)} \rceil$ messages, which are pipelined in the network. Each of the $n(k-1)$ groups of messages

reaches each destination node through a different node disjoint path. As a consequence, a lower bound for the number of time steps required for this problem is $\lceil \frac{M}{n(k-1)} \rceil + n + 1$.

In a multinode broadcasting problem on $GH_{n,k}$, each node receives a total of $M(k^n - 1)$ messages, $M$ messages from each one of the $k^n - 1$ other nodes. As a consequence, a lower bound for the number of message transmissions is $M(k^n - 1)k^n$. Since each node of $GH_{n,k}$ has $n(k-1)$ incident edges, a lower bound for the number of time steps required for this problem is $\lceil \frac{M(k^n-1)}{n(k-1)} \rceil$.

In a single node scattering problem on $GH_{n,k}$, the source node transmits a total of $M(k^n - 1)$ messages, $M$ messages to each one of the other nodes. Since each node of $GH_{n,k}$ has $n(k-1)$ incident edges, a lower bound for the number of message transmissions required for this problem is $\lceil \frac{M(k^n-1)}{n(k-1)} \rceil$. A message destined to a specific node must travel a number of edges equal to the shortest distance between that node and the source node. Therefore, a lower bound for the number of message transmissions required is the sum of the shortest distances of all nodes to the source node, multiplied by $M$, since each node receives $M$ messages from the source. At distance $d$, $1 \leq d \leq n$, from a source node of $GH_{n,k}$ there are $(k-1)^d \binom{n}{d}$ nodes and we conclude that a lower bound for the number of message transmission is:

$$M \sum_{d=1}^{n} d(k-1)^d \binom{n}{d} = Mn(k-1)k^{n-1}.$$

A multinode scattering problem can be viewed as $k^n$ single node scattering problems, one from each node of $GH_{n,k}$. A lower bound for the number of message transmissions is derived from the lower bound for the number of message transmissions required for the single node scattering problem, multiplied by $k^n$. This lower bound is equal to $Mn(k-1)k^{2n-1}$. Each node has $n(k-1)$ incident edges and at most $k^n n(k-1)$ message transmissions can be performed at each time step. Consequently, a lower bound for the number of time steps required for this problem is $Mk^{n-1}$.

Table 5 summarizes the lower bounds for all of the above problems. These lower bounds were derived in a similar manner to the lower bound derived in [4] for the binary hypercube.

| Problem | Time steps | Message transmissions |
|---|---|---|
| Single node broadcasting | $\lceil \frac{M}{n(k-1)} \rceil + n + 1$ | $M(k^n - 1)$ |
| Multinode broadcasting | $\lceil \frac{M(k^n-1)}{n(k-1)} \rceil$ | $M(k^n - 1)k^n$ |
| Single node scattering | $\lceil \frac{M(k^n-1)}{n(k-1)} \rceil$ | $Mn(k-1)k^{n-1}$ |
| Multinode scattering | $Mk^{n-1}$ | $Mn(k-1)k^{2n-1}$ |

Table 5: Lower bounds on the the $GH_{n,k}$ network.

## 5.2 Single node broadcasting

In a single node broadcasting, a source node $s$ transmits the same group of $M$ messages to each other node. We use $BST_s$ to develop the single node broadcasting algorithm.

The single node broadcasting algorithm from node $s$ proceeds as follows:

1. The $M$ messages the source node $s$ wishes to broadcast are communicated over all of its incident edges

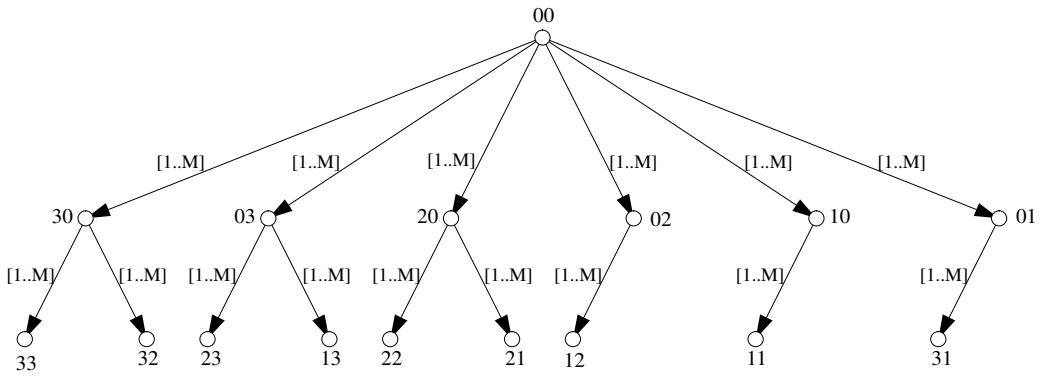


Figure 5: Single node broadcasting on the $GH_{2,4}$ network using $BST_0$.

simultaneously and are pipelined down each one of the subtrees of $BST_s$. We have to mention that the message header always carries the identity of the source node.

2. As soon as an intermediate node $v$ receives a message header with the identity of the source node $s$, it identifies its children nodes in $BST_s$ as a translation with respect to $s$ of the children of node $T_s^{-1}(v)$ in $BST_0$. Subsequently, it forwards each message it receives from its parent to all of its children nodes in $BST_s$ simultaneously.

The propagation of the messages down $BST_s$ continues until all leaf nodes of $BST_s$ receive the $M$ messages. An example of a single node broadcasting algorithm on the $GH_{2,4}$ can be seen in Fig. 5.

Each destination node receives the $M$ messages once, and as a consequence the number of message transmissions performed is $M(k^n - 1)$, which is optimal. However, the number of time steps required is $M + n - 1$ which is only asymptotically optimal, since the algorithm does not take advantage of the node disjoint paths that exist between $s$ and the other nodes of the network.

## 5.3 Multinode broadcasting

In a multinode broadcasting algorithm, each node of the network transmits $M$ messages to all the other nodes. Each node $s$ uses $BSG_s$ for the transmission of its messages. $BSG_0$ can be replicated at any other node $s$ of $GH_{n,k}$ using the operation of translation with respect to $s$, as explained in section 4. As mentioned in section 2, the messages originating at individual nodes of the network will be interleaved in such a manner, that no two messages will contend for the same edge at any time during the execution of the algorithm (lemma 5).

The multinode broadcasting algorithm proceeds as follows:

1. Each source node $s$ transmits the $M$ messages it wishes to broadcast to all of its neighbors simultaneously. The identity of the source node $s$, along with a number to indicate the spanning tree of $BSG_s$ in which the messages are transmitted, are always included in the message header.

2. When an intermediate node $v$ of a $BSG_s$ receives a group of $M$ messages originating at node $s$, it stores a copy, and performs the following procedures. The messages have to be forwarded to the first child of node $v$ in $BSG_s$. If node $T_s^{-1}(v)$ has period $P$ (definition 6) then the group of $M$ messages is

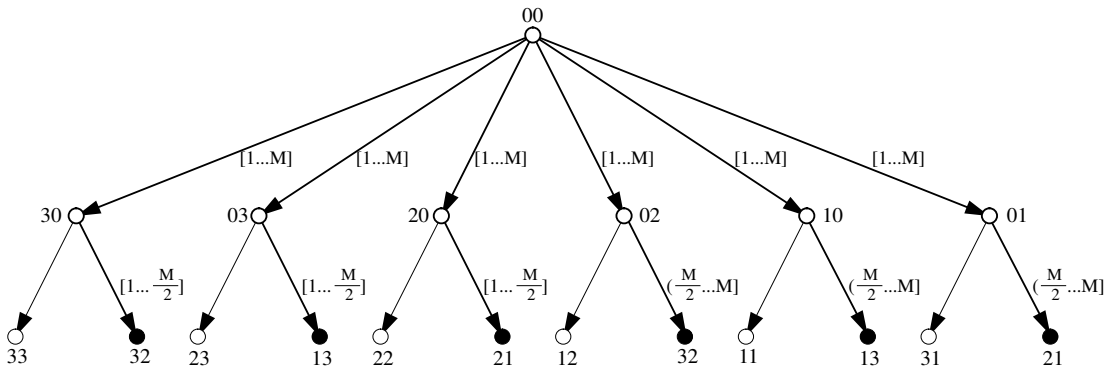


Figure 6: Multinode broadcasting on the $GH_{2,4}$ network using $BSG_0$.

split into $\frac{n(k-1)}{P}$ subgroups of $\frac{MP}{n(k-1)}$ messages each. Node $v$ of the $i^{th}$ spanning tree of $BSG_s$ sends the $(i \text{ div } P)^{th}$ subgroup of messages to its first child node in $BSG_s$.

When an intermediate node $v$ receives an acknowledgement from one of its children nodes in $BSG_s$, it forwards the messages it received in the past from node $s$ to its next child in $BSG_s$ following the splitting technique described in the previous paragraph. When an acknowledgement is received from the last child node of $v$ in $BSG_s$, node $v$ sends an acknowledgement with the identity of $s$ to its parent node in $BSG_s$.

3. When a leaf node of $BSG_s$ receives a group of messages broadcast by node $s$, it sends an acknowledgement with the identity of $s$ to its parent node in $BSG_s$.

The algorithm terminates when each source node receives acknowledgements from all its neighbors. In this algorithm, the transmission of messages in each $BSG_s$ corresponds to a simultaneous depth first traversal of its spanning trees. In order to prove that using this algorithm, no two messages contend for the same edge at any time step during its execution, we have to show that the requirement of lemma 5 is satisfied. Let us remind that by $E_i(s)$ we denote the set of $n(k-1)$ directed edges on which messages originating at node $s$ are transmitted at time step $i$ of a multinode broadcasting algorithm. Since a multinode algorithm proceeds symmetricly from all nodes of $GH_{n,k}$, the $n(k-1)$ directed edges in each $E_i(s)$, are obtained as a translation with respect to $s$ of the $n(k-1)$ directed edges of $E_i(0)$. According to lemma 5, if at each time step $i$, the $n(k-1)$ directed edges in $E_i(0)$ are all of different dimensions, then the sets of $n(k-1)$ directed edges $E_i(s)$, for $s$ ranging over all nodes of $GH_{n,k}$ are disjoint, and as a consequence messages originating at individual nodes are transmitted over disjoint sets of edges at time step $i$. The multinode broadcasting algorithm described above, proceeds symmetricly from all nodes of $GH_{n,k}$, since each $BSG_s$ is a translation with respect to $s$ of $BSG_0$. This means that, if an edge $(v,u)$ is used for the transmission of a message originating at node 0 during time step $i$, then edge $(T_s(v), T_s(u))$ is used for the transmission of a message originating at node $s$ of $GH_{n,k}$ at time step $i$. At each time step, messages originating at node 0 are transmitted over $n(k-1)$ corresponding directed edges of the $n(k-1)$ spanning trees of $BSG_0$. From the properties of $BSG_0$ (lemma 7), these edges are rotations of each other and as a consequence of different dimensions, and the requirement of lemma 5 is satisfied. An example of a multinode broadcasting algorithm

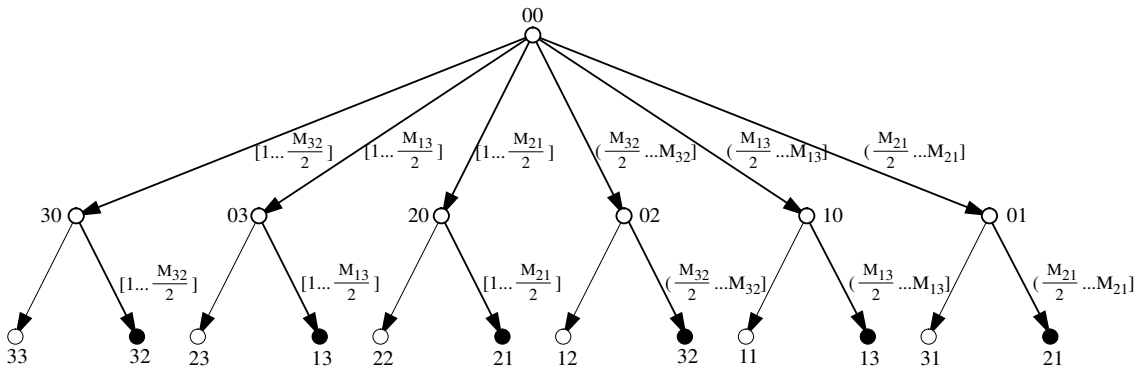


Figure 7: Single node scattering on the $GH_{2,4}$ network using $BSG_0$.

on the $GH_{2,4}$ network can be seen in Fig. 6. This figure helps illustrate the technique of message splitting performed by the algorithm.

The number of message transmissions performed is $M(k^n - 1)k^n$, which is optimal, since each of the $k^n$ nodes of $GH_{n,k}$ receives the $M$ messages originating at any other node once. The number of time steps required is $\lceil \frac{M(k^n-1)}{n(k-1)} \rceil$, which is also optimal.

If each source node $s$ wishes to broadcast one message to all the other nodes, then $BST_s$ is used with a similar method. The algorithm achieves again the minimum number of message transmissions, $(k^n - 1)k^n$, but it is only asymptotically optimal, $O(\frac{k^n-1}{n(k-1)})$.

## 5.4 Single node scattering

In a single node scattering algorithm, a source node $s$ transmits distinct groups of $M$ messages to each other node. Node $s$ uses $BSG_s$ for the transmission of its messages. Each source node keeps a table of approximately $\frac{k^n}{n(k-1)}$ nodes. The table includes the nodes of the first spanning tree of $BSG_0$, sorted in reverse ordering of their distance from node 0. The nodes in the table correspond to the transmission order of the first port of $BSG_0$, and each one is accompanied by a number to indicate its period $P$. Recall that nodes with period $P$ that belong to nonfull necklaces have $\frac{n(k-1)}{P}$ paths to node 0 through $BSG_0$. The interesting property of this algorithm is that nodes that belong to nonfull necklaces with period $P$ receive $\frac{MP}{n(k-1)}$ of their $M$ messages through each one of the $\frac{n(k-1)}{P}$ paths from node 0.

The single node scattering algorithm proceeds as follows:

For each node $v$ in the table of $\frac{k^n}{n(k-1)}$ entries do the following:

1. If the source is node 0, then it transmits messages destined to nodes $v$, $R(v)$, $R^2(v)$,...,$R^{n(k-1)-1}(v)$, simultaneously. If $v$ belong to a full necklace then all of these nodes are distinct and node 0 transmits the $M$ messages destined to node $R^i(v)$, $0 \leq i < n(k-1)$, through its $i^{th}$ port. However, if node $v$ has period $P$ and belongs to a nonfull necklace, then these nodes are not distinct but they are $P$ distinct nodes repeated $\frac{n(k-1)}{P}$ times, in other words it is the unfolded necklace of a nonfull necklace that contains $P$ nodes (definition 7). In this case each of the $P$ groups of $M$ messages node 0 has to transmit is split into $\frac{n(k-1)}{p}$ subgroups, each containing $\frac{MP}{n(k-1)}$ messages. The $i^{th}$, $0 \leq i < \frac{n(k-1)}{P}$, subgroup of the $j^{th}$, $0 \leq j < P$, group of messages is transmitted over port $iP + j$ of node 0. As a

consequence, each of the $P$ nodes of a nonfull necklace receives $\frac{MP}{n(k-1)}$ of its $M$ messages through each of the $\frac{n(k-1)}{P}$ paths from node 0 through $BSG_0$.

If the source node is any other node $s$ of $GH_{n,k}$, then $s$ transmits messages destined to nodes $T_s(v)$, $T_s(R(v))$, $T_s(R^2(v))$,...,$T_s(R^{n(k-1)-1}(v))$, simultaneously, using the same technique of message splitting described above for node 0.

We have to mention that each message header includes the identity of the destination node of the messages and a number that indicates the spanning tree of $BSG_s$ in which it is transmitted.

2. As soon as an intermediate node $v$ receives a new message header, it performs the following procedures. If node $v$ is the destination of the message it stores a copy and removes it from the network. If $v$ is not the destination of the message, the identity of the child node to which the message will be forward has to be determined. Node $v$ of the $i^{th}$ spanning tree of $BSG_s$ identifies the first digit to the left of digit $(n-1-i) \bmod n$ in its label that is not equal to the corresponding digit of the destination node. The message is forwarded to the child node of $v$ with this digit equal to the corresponding digit of the destination node. Subsequent messages that follow the same message header are forwarded to the same child node.

3. As soon as a source node have transmitted the messages to nodes $T_s(R^i(v))$, $0 \leq i < n(k-1)$, through its incident edges, it starts transmitting messages to nodes $T_s(R^i(v))$, $0 \leq i < n(k-1)$, for the next entry $u$ in the table.

An instance of the single node scattering on $GH_{2,4}$ for messages transmitted from node 0 to nodes 32, 13, and 21 is shown in Fig. 7, in order to demonstrate the message splitting technique described above.

Since $BSG_s$ is a shortest path spanning graph, each message follows a shortest path to its destination node and as a consequence the minimum number of message transmissions, $Mn(k-1)k^{n-1}$, is achieved. Furthermore, an equal number of the $M(k^n-1)$ messages the source node has to transmit is transmitted over each one of its incident edges. This, combined with the fact that messages destined to nodes that are the furthest from the source are transmitted first, helps achieve the minimum number of time steps, $\lceil \frac{M(k^n-1)}{n(k-1)} \rceil$.

If the source node $s$ wishes to transmit one message to each one of the other nodes then $BST_s$ is used with a similar method. The algorithm achieves again the minimum number of message transmissions, $n(k-1)k^{n-1}$. However, the time is only asymptotically optimal, $O(\frac{k^n-1}{n(k-1)})$, since $BST_s$ is balanced only to within a constant factor.

## 5.5 Multinode scattering

In a multinode scattering algorithm each node transmits distinct groups of $M$ messages to each other node. Each node $s$ uses $BSG_s$ for the transmission of its messages. $BSG_0$ can be replicated at any other node $s$ of $GH_{n,k}$ using the operation of translation with respect to $s$, as explained in section 4. As in the multinode broadcasting algorithm, messages originating at individual nodes will be interleaved in such a manner that no two messages will contend for the same edge at any time during the execution of the algorithm (lemma 5). The method used for the multinode scattering algorithm is similar to the one used for the single node scattering algorithm, but simultaneously executed from all nodes of the network. Each node keeps a table of

approximately $\frac{k^n}{n(k-1)}$ nodes. The nodes in the table correspond to the transmission order of the first port of $BSG_0$, and each one is accompanied by a number to indicate its period $P$.

The multinode scattering algorithm from each node of the network proceeds as follows:

For each node $v$ in the table of $\frac{k^n}{n(k-1)}$ entries do the following:

1. Source node $s$ determines the destination of the messages to be transmitted over its $i^{th}$, $0 \leq i < n(k-1)$, port as $T_s(R^i(v))$. For node $v$ with period $P$, each of the $P$ groups of $M$ messages that have to be transmitted by the source node is split into $\frac{n(k-1)}{P}$ subgroups of $\frac{MP}{n(k-1)}$ messages each. The $i^{th}$, $0 \leq i < \frac{n(k-1)}{P}$, subgroup of the $j^{th}$, $0 \leq j < P$, group of messages is transmitted over the $(iP+j)^{th}$ port of the source node.

   We have to mention that the identity of the destination node and a number that indicates the spanning tree of $BSG_0$ in which the messages are transmitted are included in the message header.

2. As soon as an intermediate node $v$ receives a new message header, it has to wait until it receives the messages that follows it. If node $v$ is the destination node of the messages, it stores a copy and removes them from the network. If node $v$ is not the destination node of the messages, it has to identify the child node to which the messages have to be forwarded. Node $v$ of the $i^{th}$ spanning tree of $BSG_s$, locates the first digit to the left of digit $(n-1-i) \bmod n$ in its label that is not equal to the corresponding digit of the destination node. The messages are forwarded to the child node of $v$ with this digit equal to the corresponding digit of the destination node.

3. When the messages transmitted from a source node $s$ have reached their destination nodes $T_s(R^i(v))$, $0 \leq i < n(k-1)$, then $s$ can transmit messages to nodes $T_s(R^i(u))$, $0 \leq i < n(k-1)$, for the next entry $u$ in the table. For example, if the distance from the source to the current destination nodes is $d$ then the messages to the next group of nodes is transmitted $d\frac{MP}{n(k-1)}$ time steps after the starting transmission time of the current group of messages.

From the properties of $BSG_0$, we know that the $n(k-1)$ paths that lead to nodes $R^i(v)$, $0 \leq i < n(k-1)$, through its spanning trees $i$, $0 \leq i < n(k-1)$, respectively, are rotations of each other (lemma 7), and as a consequence, the $n(k-1)$ directed edges at each level of these paths are of different dimensions. Each node in a path receives all the messages from its parent node before it starts transmitting them to the next node down the path. As a consequence, at each time step, $n(k-1)$ directed edges that are all at the same level of the paths are used. Since these edges are all of different dimensions the requirement of lemma 5 is satisfied, and no two messages contend for the same edge during the execution of the algorithm.

Each message follows a shortest path to its destination node and the minimum number of message transmissions, $Mn(k-1)k^{2n-1}$, is achieved. Furthermore, an equal number of the $M(k^n-1)$ messages that each source node has to transmit are transmitted over each one of its incident edges and the minimum number of time steps, $Mk^{n-1}$, is achieved.

When each source node wishes to transmit one message to each one of the other nodes a similar method is followed, but the $BST_s$ spanning tree is used. Although the minimum number of messages transmissions, $n(k-1)k^{2n-1}$, is achieved, the time is only asymptotically optimal, $O(k^{n-1})$, since $BST_s$ is balanced only to within a constant factor.

# 6 Conclusions

A general framework was developed on the generalized hypercube network, that led to the construction of a shortest path, balanced to within a constant factor, spanning tree, and a shortest path spanning graph. Several definitions such as the ones for the translation and the rotation operations and the grouping of the nodes into necklaces were developed.

The applications of the spanning graphs to the development of optimal communication algorithms was demonstrated by giving a number of algorithm for the single and multinode broadcasting, and for the single and multinode scattering problems, under the all-port communication assumption, and the store-and-forward model. These are algorithms in which all nodes of the network know in advance the communication pattern. The method is mostly useful for communication problems that require a group or all nodes of the network to be sources of messages, such as the multinode broadcasting and scattering problems. The property that corresponding edges of the subtrees are of different dimensions, along with lemma 5, give the necessary condition for messages to be interleaved so that conflict are avoidance. The spanning graphs can be used for the development of algorithms for a number of other communication problems, or under a variety of communication models, such as the one-port model. It was also pointed out that the algorithms developed in this paper are applicable to the solution of a wide range of problems such as matrix computations, image manipulations, linear algebra, and database operations, to name a few.

Our algorithms illustrate that it is advantageous to use all of the communication links of a network simultaneously in communication intensive tasks, and that flexible techniques that take advantage of this capability can be developed. This leads to a considerable increase in network bandwidth utilization, while at the same time decreasing the routing time required for the completion of the algorithms.

We are confident that a general framework that leads to the construction of spanning graphs with similar properties can be potentially developed for networks that belong to a subclass of the Cayley graphs. This will offer a uniform solution to a wide range of communication problems on a wide range of networks. Future research could move towards various directions, the most important being the generalization of the developed framework to a class of interconnection networks that exhibit specific characteristics, and the application of this framework to the solution of other types of problems.

# References


[1] S.B. Akers, D. Harel, and B. Krishnamurthy, "The Star Graph: An Attractive Alternative to the Hypercube", in *Proceedings of the International Conference on Parallel Processing*, St. Charles, IL, pp. 393-400, 1987.

[2] S.B. Akers and B. Krishnamurthy, "A Group Theoretic Model for Symmetric Interconnection Networks", *IEEE Transactions on Computers*, vol. 38, no. 4, pp. 555-566, 1989.

[3] D.P. Bertsekas and J.N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*, Prentice-Hall, 1989.

[4] D.P. Bertsekas, C. Ozveren, G.D. Stamoulis, P. Tseng, and J.N. Tsitsiklis, "Optimal Communication Algorithms for Hypercubes", *Journal of Parallel and Distributed Computing*, vol. 11, no. 4, pp. 263-275, 1991.

[5] L.N. Bhuyan and D.P. Agrawal, "Generalized Hypercube and Hyperbus Structures for a Computer Network", *IEEE Transactions on Computers*, vol. 33, no. 4, pp. 323-333, 1984.

[6] P. Fragopoulou and S.G. Akl, "Optimal Communication Algorithms on the Star Interconnection Network", in *Proceedings of the IEEE Symposium on Parallel and Distributed Processing*, Dallas, TX, pp. 702-711, 1993.

[7] P. Fragopoulou and S.G. Akl, "Optimal Communication Algorithms on Star Graphs Using Spanning Tree Constructions", to appear in the *Journal of Parallel and Distributed Computing.*

[8] P. Fragopoulou and S.G. Akl, "Edge-Disjoint Spanning Trees on the Star Network with Applications to Fault Tolerance", Technical Report no. 93-354, Department of Computing and Information Science, Queen's University, Kingston, ON, Canada, 1993.

[9] P. Fragopoulou, S.G. Akl, and H. Meijer, "Optimal Communication Primitives on the Generalized Hypercube Network", Technical Report, no. 94-362 Department of Computing and Information Science, Queen's University, Kingston, ON, Canada, 1994.

[10] P. Fragopoulou and S.G. Akl, "A Framework for Optimal Communication on the Multidimensional Torus Network", Technical Report no. 94-363, Department of Computing and Information Science, Queen's University, Kingston, ON, Canada, 1994.

[11] P. Fraigniaud and E. Lazard, "Methods and Models of Communication in Usual Networks", to appear in *Discrete Applied Mathematics.*

[12] P. Fraigniaud, "Complexity Analysis of Broadcasting in Hypercubes with Restricted Communication Capabilities", *Journal of Parallel and Distributed Computing*, vol. 16, pp. 15-26, 1992.

[13] L. Garding and T. Tambour, *Algebra for Computer Science*, Springer-Verlag, 1988.

[14] P.T. Gaughan and S. Yalamanchili, "Adaptive Routing Protocols for Hypercube Interconnection Networks", Computer, pp. 12-23, 1993.

[15] S.L. Johnson, "Communication Efficient Basic Linear Algebra Computations on Hypercube Architectures", *Journal of Parallel and Distributed Computing*, vol. 4, pp. 133-172, 1987.

[16] S.L. Johnson and C.T. Ho, "Optimum Broadcasting and Personalized Communication in Hypercubes", *IEEE Transactions on Computers*, vol. 38, no. 9 , pp. 1249-1268, 1989.

[17] S. Lakshmivarahan, J.S. Jwo, and S.K. Dhall, " Symmetry in Interconnection Networks based on Cayley Graphs of Permutation Groups: A Survey", *Parallel Computing*, vol. 19, no. 4, pp. 361-407, 1993.

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

[19] K.A. Ross and C.R.B. Wright, *Discrete Mathematics*, Prentice-Hall, 1985.

[20] Y. Saad and M.H. Schultz, "Data Communication in Parallel Architectures", *Parallel Computing*, vol. 11, pp. 131-150, 1989.

[21] Y. Saad and M.H. Schultz, "Data Communication in Hypercubes", *Journal of Parallel and Distributed Computing*, vol. 6, pp. 115-135, 1989.

[22] Q.F. Stout and B. Wagar, "Intensive Hypercube Communication", *Journal of Parallel and Distributed Computing*, vol. 10, pp. 167-181, 1993.