Technical Report No. 94-363

A FRAMEWORK FOR OPTIMAL COMMUNICATION ON THE MULTIDIMENSIONAL TORUS NETWORK

by
Paraskevi Fragopoulou and Selim G. Akl

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 *multidimensional torus*, 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 multidimensional torus 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 all other processors. 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 multidimensional torus network 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, multidimensional torus, parallel algorithm, spanning tree.

Figure 1: The $MT_{2,4}$ 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 *multidimensional torus* 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 *multidimensional torus* network, which has been proven to be a flexible topology for the interconnection of processors [5, 14]. An n-dimensional, k-ary multidimensional torus, denoted by $MT_{n,k}$, 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 by $j \bmod k$, $j \in \{-1, 1\}$, 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 all $0 \leq i \leq n-1$, and $v_i' = (v_i + j) \bmod k$, $j \in \{-1, 1\}$. The $MT_{2,4}$ 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	$M + n\lfloor \frac{k}{2} \rfloor + 1$	$M(k^n - 1)$
Multinode broadcasting	$\lceil \frac{M(k^n-1)}{2n} \rceil$	$M(k^n - 1)k^n$
Single node scattering	$\lceil \frac{M(k^n-1)}{2n} \rceil$	$\lceil \frac{Mnk^{n+1}}{4} \rceil$, if k is even $\lceil \frac{Mn(k^2-1)k^{n-1}}{4} \rceil$, if k is odd
Multinode scattering	$\lceil \frac{Mk^{n+1}}{8} \rceil$, if k is even $\lceil \frac{M(k^2-1)k^{n-1}}{8} \rceil$, if k is odd	$\lceil \frac{Mnk^{2n+1}}{4} \rceil$, if k is even $\lceil \frac{Mn(k^2-1)k^{2n-1}}{4} \rceil$, if k is odd

Table 1: The main results obtained on the $MT_{n,k}$ 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 multidimensional torus (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 given time during the execution of the 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. This along with the fact that each message follows a shortest path to its destination help achieve the minimum number of time steps required for the algorithms to complete. The same algorithms can be used to derive 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 multidimensional torus network can be found in [14]. Another collection of communication algorithms on the $MT_{n,k}$ network under a variety of communication models can be found in [11]. The method of spanning graph construction has been previously used to design communication algorithms on other interconnection networks, such as the binary hypercube [16, 4], the generalized hypercube [9], and the star [6, 7] networks.

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 multidimensional torus 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 multidimensional torus $MT_{n,k}$, 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 $2n$ other nodes with which it differs in only one digit by $j \bmod k$, $j \in \{-1, 1\}$, 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 $0 \leq i \leq n-1$, and $v'_i = (v_i + j) \bmod k$, $j \in \{-1, 1\}$, Fig. 1. The network is edge and node symmetric with degree $2n$ (number of edges at each node) and diameter $n\lfloor\frac{k}{2}\rfloor$ (maximum shortest distance between any pair of nodes). $MT_{n,k}$ 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 $2n$ generators that define the edges of $MT_{n,k}$ are denoted by g_i^j, $0 \leq i \leq n-1$, $j \in \{-1, 1\}$. Generator g_i^j connects 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 $MT_{n,k}$ is connected to $2n$ other nodes through dimensions g_i^j, $0 \leq i \leq n-1$, $j \in \{-1, 1\}$. In what follows, node 00...0 of $MT_{n,k}$ that contains only zero digits is referred to as node 0^n. 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 multidimensional torus contains a ring of k nodes in each dimension. The multidimensional torus network is also referred to as the generalized hyperrectangular network [14].

We now define an operation on nodes of the multidimensional torus 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^n of $MT_{n,k}$, we will derive an isomorphic spanning graph, with the same properties, rooted at any other node s of $MT_{n,k}$, using a translation of the graph rooted at node 0^n with respect to s. As a consequence, it is sufficient to construct a spanning graph rooted at node 0^n of $MT_{n,k}$. The translation operation on $MT_{n,k}$ is analogous to the exclusive-OR operation on nodes of the binary hypercube [16, 3, 4].

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 $MT_{3,4}$, $T_s(v) = 323$ 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, $0 \leq i \leq n-1$, $j \in \{-1, 1\}$. 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 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 multidimensional torus that preserves the topology of the network and the dimension of each edge.

We now define another operation on $MT_{n,k}$, 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 multidimensional torus has properties similar to those of the right cyclic shift operation on nodes of the binary hypercube [16, 3, 4].

Definition 2: Consider the function $r(i) = (k-i) \bmod k$ from the set $\{0, 1, ..., k-1\}$ to itself. 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 $r(v_0)v_{n-1}v_{n-2}...v_{i+1}v_iv_{i-1}...v_1$. This can be viewed as a right cyclic shift of the digits of v with the wraparound digit being mapped through 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 application of rotation. For example, for nodes $v = 321$ and $u = 012$ of $MT_{3,4}$, $R(v) = 332$ and $R(u) = 201$.

Lemma 2: The rotation operation has the following properties:

1. If (v,u) is an edge of dimension g_i^j, $0 \leq i \leq n-1$, $j \in \{-1, 1\}$, then edge $(R(v), R(u))$ is of dimension $g_{i'}^{j'}$ so that:
$$i' = (i-1) \bmod n,$$
$$j' = \begin{cases} -j, & \text{if } i = 0, \\ j, & \text{otherwise.} \end{cases}$$

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

Proof: We prove each property separately.

1. Assume that $1 \leq i \leq n-1$. If we express (v,u) and $(R(v), R(u))$ as follows:
$$\begin{array}{rl} (v_{n-1}...v_{i+1}\underline{v_i}v_{i-1}...v_0, & v_{n-1}...v_{i+1}\underline{((v_i+j) \bmod k)}v_{i-1}...v_0), \\ (r(v_0)v_{n-1}...v_{i+1}\underline{v_i}v_{i-1}...v_1, & r(v_0)v_{n-1}...v_{i+1}\underline{((v_i+j) \bmod k)}v_{i-1}...v_1), \end{array}$$

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

Assume that $i = 0$. If we express (v, u) and $(R(v), R(u))$ as follows:

$$\begin{array}{rl} (v_{n-1}...v_i...v_1\underline{v_0}, & v_{n-1}...v_i...v_1\underline{(v_0+j) \bmod k}), \\ (\underline{r(v_0)}v_{n-1}...v_i...v_1, & \underline{r[(v_0+j) \bmod k]}v_{n-1}...v_i...v_1), \end{array}$$

it is clear that if v and u differ in their 0^{th} digit by $j \bmod k$, $j \in \{-1, 1\}$, then $R(v)$ and $R(u)$ differ in their $(n-1)^{st}$ digit by $r[(v_0+j) \bmod k] - r(v_0) = [k - (v_0+j) - (k - v_0)] \bmod k = -j$.

2. The rotation operation is an automorphism on $MT_{n,k}$ that maps node 0^n to itself. As an extension to this, nodes obtained as rotations of each other are all at the same distance from node 0^n. □

Lemma 3: If (v, u) is a directed edge of dimension g_i^j, then the $2n$ directed edges derived from (v, u) by consecutive applications of the rotation operation are all of different dimensions [10].

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. For example, for edge $(01, 31)$ of $MT_{2,4}$ the $2n = 4$ edges produced by consecutive applications of the rotation operation are $(01, 31) \xrightarrow{R} (30, 33) \xrightarrow{R} (03, 13) \xrightarrow{R} (10, 11)$ and their corresponding dimensions are $g_1^{-1} \xrightarrow{R} g_0^{-1} \xrightarrow{R} g_1^1 \xrightarrow{R} g_0^1$. □

To summarize, the translation and the rotation operations are automorphisms on $MT_{n,k}$ 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 (lemmas 2 and 3). Finally, the topology of $MT_{n,k}$ or one of its subnetworks remains unchanged under translation or rotation.

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

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

Lemma 4: Necklaces have the following properties:

1. A necklace contains *at most* $2n$ nodes.

2. The size of a necklace always divides $2n$.

3. All nodes of a necklace are at the same distance form node 0^n.

Proof: We prove each property separately.

1. From the definition of rotation it can be verified that $R^{2n}(v) = v$ for every node v of $MT_{n,k}$. However, we say *at most* $2n$ rotations because the same node can emerge after less than $2n$ rotations. For example, for node $v = 131$ of $MT_{3,4}$, $R^2(131) = 131$ and the same node emerges after only 2 rotations and not $2n = 6$.

2. The proof for this property can be found in group theory. A rotation operation is an automorphism on $MT_{n,k}$ of order $2n$. 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 $2n$ distinct nodes. A *nonfull necklace* is a necklace that contains less than $2n$ nodes. A node of $MT_{n,k}$ belongs to a nonfull necklace, if its label has a nontrivial symmetry with respect to the rotation operation. It can be verified that nodes of this type consist 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: $r^{m-1}(v_{\frac{n}{m}-1})...r^{m-1}(v_1)r^{m-1}(v_0)...r^i(v_{\frac{n}{m}-1})...r^i(v_1)r^i(v_0)...r(v_{\frac{n}{m}-1})...r(v_1)r(v_0)v_{\frac{n}{m}-1}...v_1v_0$ [10]. For example, node 311331 of $MT_{6,4}$, which belongs to a nonfull necklace that contains 4 nodes constists of the substring 31 of two digits which is repeated modified three times as follows: $r^2(3)r^2(1)r(3)r(1)31 = 311331$.

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

The necklaces of $MT_{3,3}$

$$\begin{array}{ll}
d=0: & (\ \overset{0}{\underline{000}}\ ) \\
d=1: & (\ \overset{0}{\underline{200}},\ \overset{1}{020},\ \overset{2}{002},\ \overset{3}{100},\ \overset{4}{010},\ \overset{5}{001}\ ) \\
d=2: & (\ \overset{0}{\underline{220}},\ \overset{1}{022},\ \overset{2}{102},\ \overset{3}{110},\ \overset{4}{011},\ \overset{5}{201}\ ) \\
 & (\ \overset{0}{\underline{210}},\ \overset{1}{021},\ \overset{2}{202},\ \overset{3}{120},\ \overset{4}{012},\ \overset{5}{101}\ ) \\
d=3: & (\ \overset{0}{\underline{222}},\ \overset{1}{122},\ \overset{2}{112},\ \overset{3}{111},\ \overset{4}{211},\ \overset{5}{221}\ ) \\
 & (\ \overset{0}{\underline{212}},\ \overset{1}{121}\ )
\end{array}$$

The necklaces of $MT_{3,4}$

$$\begin{array}{ll}
d=0: & (\ \overset{0}{\underline{000}}\ ) \\
d=1: & (\ \overset{0}{\underline{300}},\ \overset{1}{030},\ \overset{2}{003},\ \overset{3}{100},\ \overset{4}{010},\ \overset{5}{001}\ ) \\
d=2: & (\ \overset{0}{\underline{330}},\ \overset{1}{033},\ \overset{2}{103},\ \overset{3}{110},\ \overset{4}{011},\ \overset{5}{301}\ ) \\
 & (\ \overset{0}{\underline{310}},\ \overset{1}{031},\ \overset{2}{303},\ \overset{3}{130},\ \overset{4}{013},\ \overset{5}{101}\ ) \\
 & (\ \overset{0}{\underline{200}},\ \overset{1}{020},\ \overset{2}{002}\ ) \\
d=3: & (\ \overset{0}{\underline{333}},\ \overset{1}{133},\ \overset{2}{113},\ \overset{3}{111},\ \overset{4}{311},\ \overset{5}{331}\ ) \\
 & (\ \overset{0}{\underline{320}},\ \overset{1}{032},\ \overset{2}{203},\ \overset{3}{120},\ \overset{4}{012},\ \overset{5}{201}\ ) \\
 & (\ \overset{0}{\underline{230}},\ \overset{1}{023},\ \overset{2}{102},\ \overset{3}{210},\ \overset{4}{021},\ \overset{5}{302}\ ) \\
 & (\ \overset{0}{\underline{313}},\ \overset{1}{131}\ ) \\
d=4: & (\ \overset{0}{\underline{233}},\ \overset{1}{133},\ \overset{2}{113},\ \overset{3}{211},\ \overset{4}{321},\ \overset{5}{332}\ ) \\
 & (\ \overset{0}{\underline{231}},\ \overset{1}{323},\ \overset{2}{132},\ \overset{3}{213},\ \overset{4}{121},\ \overset{5}{312}\ ) \\
 & (\ \overset{0}{\underline{220}},\ \overset{1}{022},\ \overset{2}{202}\ ) \\
d=5: & (\ \overset{0}{\underline{223}},\ \overset{1}{122},\ \overset{2}{212},\ \overset{3}{221},\ \overset{4}{322},\ \overset{5}{232}\ ) \\
d=6: & (\ \overset{0}{\underline{222}}\ )
\end{array}$$

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

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

Definition 4: The binary correspondent of a node v of $MT_{n,k}$ is the binary number obtained if we substitute each nonzero digit in v with 1. One particular node of each necklace is now distinguished as follows:

1. Select the nodes of the necklace that have the largest binary correspondent.

2. Choose the largest among the nodes selected in step (1), if the k digits that are used to label the nodes

of $MT_{n,k}$ are ordered as follows: $0 < 1 < k-1 < ...i < k-i < ... < \lceil \frac{k}{2} \rceil$. This ordering of the digits is adopted in order to reflect how each digit contributes to the distance of a node from node 0^n.

The node selected in step (2) is defined to be the *generator node* of the necklace.

For example, for necklace (230, 023, 102, 210, 021, 302) of $MT_{3,4}$ nodes 230, and 210 have the largest binary correspondent among the nodes of the necklace. The generator node of the necklace is node 230 because this is the largest from nodes 230, 210, if the four digits $\{0, 1, 2, 3\}$ used to label the nodes of $MT_{3,4}$ are ordered as $0 < 1 < 3 < 2$.

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 this node from 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 $2n$ nodes, not necessarily distinct, each one obtained from its preceding 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 $\frac{2n}{P}$ times. This is possible since the size of a necklace is always a divisor of $2n$ (lemma 4). In table 3, the unfolded necklaces of $MT_{3,3}$, and those of $MT_{3,4}$ are given. A comparison with table 2 will help clarify the differences between a necklace and its corresponding unfolded necklace.

The unfolded necklaces of $MT_{3,3}$

$d = 0$:	($\underline{000}$, 000, 000, 000, 000, 000)
$d = 1$:	($\underline{200}$, 020, 002, 100, 010, 001)
$d = 2$:	($\underline{220}$, 022, 102, 110, 011, 201)
	($\underline{210}$, 021, 202, 120, 012, 101)
$d = 3$:	($\underline{222}$, 122, 112, 111, 211, 221)
	($\underline{212}$, 121, 212, 121, 212, 121)

The unfolded necklaces of $MT_{3,4}$

$d = 0$:	($\underline{000}$, 000, 000, 000, 000, 000)
$d = 1$:	($\underline{300}$, 030, 003, 100, 010, 001)
$d = 2$:	($\underline{330}$, 033, 103, 110, 011, 301)
	($\underline{310}$, 031, 303, 130, 013, 101)
	($\underline{200}$, 020, 002, 200, 020, 002)
$d = 3$:	($\underline{333}$, 133, 113, 111, 311, 331)
	($\underline{320}$, 032, 203, 120, 012, 201)
	($\underline{230}$, 023, 102, 210, 021, 302)
	($\underline{313}$, 131, 313, 131, 313, 131)
$d = 4$:	($\underline{233}$, 133, 113, 211, 321, 332)
	($\underline{231}$, 323, 132, 213, 121, 312)
	($\underline{220}$, 022, 202, 220, 022, 202)
$d = 5$:	($\underline{223}$, 122, 212, 221, 322, 232)
$d = 6$:	($\underline{222}$, 222, 222, 222, 222, 222)

Table 3: The unfolded necklaces of $MT_{3,3}$ and $MT_{3,4}$.

The property of the rotation operation that $2n$ 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 $2nk^n$ directed edges are available on $MT_{n,k}$ for message transmission at each time step. The algorithm proceeds symmetricly from all nodes and as a consequence messages originating at each one of the k^n nodes of $MT_{n,k}$ are transmitted through at most $2n$ directed edges at each time step. Let us denote by $E_i(0^n)$ the set of $2n$ directed edges on which messages originating at node 0^n are transmitted at time step i of the algorithm. Since a multinode algorithm proceeds symmetricly from each node of the network, the $2n$ directed edges on which messages originating at node s are transmitted at time step i, denoted by $E_i(s)$, is obtained from $E_i(0^n)$ using the operation of translation with respect to s (if $(v, u) \in E_i(0^n)$ 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 $2n$ directed edges in $E_i(0^n)$ are all of different dimensions, then the sets of $2n$ directed edges $E_i(s)$, where s ranges over all nodes of $MT_{n,k}$, are disjoint.

Proof: Assume two different edges $(v, u) \neq (v', u')$ of $E_i(0^n)$ 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 $MT_{n,k}$, 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 $E_i(0^n)$ since this set contains $2n$ directed edges that are all of different dimensions [4]. □

The multinode broadcasting and scattering algorithms will be developed so that at each time step i, the set $E_i(0^n)$ contains $2n$ 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 $2n$ directed edges $E_i(s)$, where s ranges over all nodes of $MT_{n,k}$, 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 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 section. 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^n of $MT_{n,k}$, and denoted by BST_{0^n}. 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 $2n$ subtrees of the root is less than a constant. The framework developed in the previous section will be the basic tool for the construction of the spanning tree with the stated properties. The i^{th}, $0 \leq i < 2n$, subtree of BST_{0^n} is defined to be the subtree that contains all nodes v of $MT_{n,k}$ with displacement $D(v) = i$. Furthermore, an isomorphic spanning tree

Figure 2: The BST_{0^n} spanning tree on the $MT_{3,3}$ network.

rooted at any other node s of $MT_{n,k}$, and denoted by BST_s, can be easily derived from BST_{0^n} using the operation of translation with respect to s. We are now ready to proceed to a formal definition of BST_{0^n}.

Definition 8: A shortest path spanning tree, balanced to within a constant factor, rooted at node 0^n of $MT_{n,k}$, and denoted by BST_{0^n}, is defined as follows: The 0^{th} subtree of BST_{0^n} contains all nodes of $MT_{n,k}$ that have displacement zero and is defined through the following parent function. For node v with displacement zero, let p be the position of its lowest order nonzero digit.

$$\text{parent}(v) = \begin{cases} \emptyset, & \text{if } v = 0^n, \\ v_{n-1}...v_{p+1}v_p^-\,v_{p-1}...v_0, & \text{otherwise,} \end{cases}$$

$$\text{where} \quad v_p^- = \begin{cases} v_p - 1, & \text{if } 0 \le k \le \lfloor \frac{k}{2} \rfloor, \\ (v_p+1) \bmod k, & \text{otherwise.} \end{cases}$$

Any other subtree i, $0 < i < 2n$, of BST_{0^n} is defined as a rotation of subtree $i-1$, that includes only those nodes of $MT_{n,k}$ that have displacement i.

The BST_{0^n} spanning tree on the $MT_{3,3}$ network can be seen in Fig. 2.

Lemma 6: BST_{0^n} has the following properties:

1. BST_{0^n} is a shortest path spanning tree rooted at node 0^n of $MT_{n,k}$.

2. BST_{0^n} is balanced to within a constant factor.

Proof: We prove each property separately.

1. We start by proving that the parent(v) function of definition 8 constructs a shortest path tree on those nodes of $MT_{n,k}$ that have displacement zero. For node v of $MT_{n,k}$ with displacement zero, its parent node is obtained by changing the value of digit v_p which is the lowest order digit of v, to v_p^-. From definition 4 we can verify that by changing this digit the resulting node also has displacement zero. Furthermore, the parent(v) function generates a shortest path from node v to node 0^n. The distance of node $v = v_{n-1}...v_{i+1}v_iv_{i-1}...v_0$ from node 0^n, denoted by $d(v)$, is:

$$d(v) = \sum_{i=0}^{n-1} |v_i|, \qquad \text{where } |v_i| = \begin{cases} v_i, & \text{if } v_i \le \lfloor \frac{k}{2} \rfloor, \\ k - v_i, & \text{otherwise.} \end{cases}$$

The parent of node v is $v' = v_{n-1} \ldots v_{p+1} v_p^- v_{p-1} \ldots v_0$ and its distance from node 0^n is:

$$d(v') = \sum_{i=0, i \neq p}^{n-1} |v_i| + |v_p^-| = \sum_{i=0, i \neq p}^{n-1} |v_i| + |v_p| - 1 = \sum_{i=0}^{n-1} |v_i| - 1 = d(v) - 1.$$

Consequently, the parent of node v is closer to node 0^n than the node itself, hence a shortest path tree. The i^{th} subtree of BST_{0^n} is obtained as a rotation of its preceding one or from the 0^{th} subtree by the application of i rotations, after excluding nodes that do not have displacement i. The nodes that are excluded are always nodes that belong to nonfull necklaces. In order to show that the i^{th} subtree of BST_{0^n} is connected we have to prove that nodes that belong to nonfull necklaces and have displacement zero are always leaf nodes of the 0^{th} subtree of BST_{0^n}. A node v with displacement zero that belongs to a nonfull necklace consists of a substring of $\frac{n}{m}$ digits which is repeated m times with its nonzero digits modified. If node v has a child node then the value of its last nonzero digit or one of its final zero digits, which belongs to the last substring of m digits, is increased in the label of the child node (according to the ordering of digits in definition 4). However, the resulting node does not have displacement zero (definition 5), it does not belong to the first subtree of BST_{0^n}, and as a consequence it cannot be a child of node v.

2. We must prove that each subtree contains $O(\frac{k^n}{2n})$ nodes. From the definition of BST_{0^n}, the i^{th}, $0 \leq i < 2n$, subtree contains nodes v for which $D(v) = i$. From the $2n$ 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 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 $\frac{k^n}{2n} - O(\frac{\sqrt{k^n}}{2n}) = O(\frac{k^n}{2n})$ 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 in the number of nodes between the largest subtree of $MT_{n,k}$ and $\frac{k^n}{2n}$ rapidly converges to one as the number of nodes increases. □

From the definition of BST_{0^n} and the fact that each subtree is obtained as a rotation of its preceding one with some nodes excluded, we conclude that corresponding nodes of the subtrees form necklaces (definition 3) and corresponding directed edges of the subtrees are of different dimensions (lemma 3). The properties of BST_{0^n} are apparent in Fig. 2.

Using the BST_{0^n} spanning tree we can easily derive a BST_s, rooted at any other node s of $MT_{n,k}$. This spanning tree is isomorphic to BST_{0^n} 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^n}. If edge (v, u) belongs to the i^{th} subtree of BST_{0^n}, 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 (lemma 1), these edges are of the same dimension.

n	k	Number of nodes k^n	Number of nodes of nonfull necklaces	Number of necklaces	Size of minimum subtree	Size of maximum subtree	$k^n/2n$	Ratio
4	3	81	1	11	10	10	10.00	1.00
4	4	256	16	36	30	35	31.88	1.10
4	5	625	1	79	78	78	78.00	1.00
4	6	1296	16	166	160	165	161.88	1.10
4	7	2401	1	301	300	300	300.00	1.00
5	3	243	3	26	24	25	24.20	1.03
5	4	1024	34	108	99	107	102.30	1.04
5	5	3125	5	315	312	314	312.40	1.00
5	6	7776	36	784	774	783	777.50	1.01
5	7	16807	7	1684	1680	1683	1680.60	1.00
6	3	729	9	63	60	62	60.67	1.02
6	4	4096	76	352	335	351	341.25	1.03
6	5	15625	25	1307	1300	1306	1302.00	1.00
6	6	46656	96	3902	3880	3901	3887.92	1.00
6	7	117649	49	9813	9800	9812	9804.00	1.00
7	3	2187	3	158	156	157	156.14	1.00
7	4	16384	130	1182	1161	1181	1170.21	1.01
7	5	78125	5	5583	5580	5582	5580.29	1.00
7	6	279936	132	20008	19986	20007	19995.36	1.00
7	7	823543	7	58828	58824	58827	58824.43	1.00
8	3	6561	1	411	410	410	410.00	1.00
8	4	65536	256	4116	4080	4115	4095.94	1.00
8	5	390625	1	24415	24414	24414	24414.00	1.00
8	6	1679616	256	104996	104960	104995	104975.94	1.00
8	7	5764801	1	360301	360300	360300	360300.00	1.00
9	3	19683	27	1098	1092	1097	1093.44	1.00
9	4	262144	568	14602	14532	14601	14563.50	1.00
9	5	1953125	125	108523	108500	108522	108506.89	1.00
9	6	10077696	720	559928	559832	559927	559871.94	1.00

Table 4: Comparison between the smallest and the largest subtrees of BST_{0^n} for sample values of n and k.

Figure 3: The BSG_{0^n} spanning graph on the $MT_{3,3}$ network.

4 Spanning graph construction

The definition of BST_{0^n} is extended to a spanning graph, rooted at node 0^n of $MT_{n,k}$, and denoted by BSG_{0^n}. This is a special type of graph, which is composed of $2n$ subtrees, rooted at the nodes adjacent to node 0^n. The i^{th}, $0 \leq i < 2n$, subtree of BSG_{0^n} contains nodes v of $MT_{n,k}$, for which $D(v) = i \bmod P(v)$. All subtrees of BSG_{0^n} are isomorphic, and each one is derived from its preceding one cyclically, by the application of a rotation operation. Furthermore, an isomorphic spanning graph rooted at any other node s of $MT_{n,k}$, and denoted by BSG_s, can be easily derived from BSG_{0^n} using the operation of translation with respect to s. We are now ready to proceed to a formal definition of BSG_{0^n}.

Definition 9: A shortest path spanning graph, rooted at node 0^n of $MT_{n,k}$, and denoted by BSG_{0^n}, is defined as follows: The 0^{th} spanning tree of BSG_0 contains all nodes of $MT_{n,k}$ that have displacement zero and is defined through the following parent function. For node v with displacement zero, let p be the position of its lowest order nonzero digit.

$$\text{parent}(v) = \begin{cases} \emptyset, & \text{if } v = 0^n, \\ v_{n-1}...v_{p+1}v_p^- \, v_{p-1}...v_0, & \text{otherwise,} \end{cases}$$

$$\text{where} \quad v_p^- = \begin{cases} v_p - 1, & \text{if } 0 \leq k \leq \lfloor \frac{k}{2} \rfloor, \\ (v_p + 1) \bmod k, & \text{otherwise.} \end{cases}$$

Any other subtree i, $0 < i < 2n$, of BSG_{0^n} is derived as a rotation of subtree $i - 1$.

The BSG_{0^n} spanning graph on the $MT_{3,3}$ network can be seen in Fig. 2.

Lemma 7: BSG_{0^n} has the following properties:

1. BSG_{0^n} is a shortest path graph rooted at node 0^n of $MT_{n,k}$.

2. Nodes with period P that belong to nonfull necklaces have $\frac{2n}{P}$ paths to node 0^n through BSG_{0^n}.

Proof: We prove each property separately.

1. The 0^{th} subtrees of BSG_{0^n} and BST_{0^n} are the same and as a consequence the proof of this property for BSG_{0^n} is the same as the proof of the same property for BST_{0^n} (lemma 5).

2. A node v with displacement zero and period P that belongs to a nonfull necklace belongs to the 0^{th} subtree of BSG_{0^n}. Since each subtree is obtained as a rotation of its preceding one, node v also belongs to subtrees iP, $0 \leq i < \frac{2n}{P}$. Consequently, v has $\frac{2n}{P}$ paths to node 0^n through BSG_{0^n}, one path through each one of the $\frac{2n}{P}$ subtrees it belongs to. □

From the definition of BSG_{0^n} and the fact that each subtree of BSG_{0^n} is obtained as a rotation of its preceding one we conclude that corresponding nodes of the subtrees form unfolded necklaces (definition 7) and corresponding directed edges are of different dimensions (lemma 3). The properties of BSG_{0^n} are apparent in Fig. 3.

Using the BSG_{0^n} graph we can easily derive a BSG_s, rooted at any other node s of $MT_{n,k}$. This graph is isomorphic to BSG_{0^n} 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^n}. If edge (v,u) belongs to the i^{th} subtree of BSG_{0^n}, then edge $(T_s(v), T_s(u))$ belongs to the i^{th} subtree of BSG_s. Since the dimension of each edge is preserved under translation, these edges are of the same dimension.

The importance of the BSG_s graph lies in several different properties it possesses. The fact that each of the $2n$ subtrees of BSG_s contains 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 subtrees of BSG_s receives an equal part of its messages from s through the edges of each subtree. 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 $2n$ 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 $E_i(0^n)$ by a translation with respect to s. According to lemma 5, if the $2n$ directed edges in $E_i(0^n)$ are all of different dimensions, then the sets of $2n$ directed edges $E_i(s)$, for fixed i (time step), and s ranging over all nodes of $MT_{n,k}$, are disjoint. In other words, at each time step i, messages originating at individual nodes are transmitted through different edges of $MT_{n,k}$. By lemma 7, the $2n$ subtrees of BSG_{0^n} are rotations of each other, and as a consequence $2n$ corresponding directed edges of the subtrees of BSG_{0^n} 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 $2n$ corresponding directed edges of the subtrees of BSG_{0^n}. Finally, the fact that BSG_{0^n} 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 $MT_{n,k}$, 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 $2n$ disjoint paths, of length at most $(n-1)\lfloor\frac{k}{2}\rfloor + k - 1$, to each one of the other nodes. In order to achieve the minimum number of time steps for this problem, the M messages are separated into $2n$ groups, each one containing $\lceil\frac{M}{2n}\rceil$ messages, which are pipelined in the network. Each of the $2n$ 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 $\lceil\frac{M}{2n}\rceil + (n-1)\lfloor\frac{k}{2}\rfloor + k - 2$.

In a multinode broadcasting problem on $MT_{n,k}$, 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 $MT_{n,k}$ has $2n$ incident edges, a lower bound for the number of time steps required for this problem is $\lceil\frac{M(k^n-1)}{2n}\rceil$.

In a single node scattering problem on $MT_{n,k}$, the source node transmits a total of $M(k^n - 1)$ messages, M messages to each one of the $k^n - 1$ other nodes. Since each node of $MT_{n,k}$ has $2n$ incident edges, a lower bound for the number of time steps required for this problem is $\lceil\frac{M(k^n-1)}{2n}\rceil$. 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 node. Let us denote by N_d the number of nodes at distance d from the source node. A lower bound for the number of message transmissions required for a single node scattering problem is the following:

$$M \sum_{d=1}^{n\lfloor\frac{k}{2}\rfloor} dN_d = Mk^n \left(\frac{\sum_{d=1}^{n\lfloor\frac{k}{2}\rfloor} dN_d}{k^n} \right). \tag{1}$$

The expression enclosed in parentheses is the average diameter of the $MT_{n,k}$ network.

Lemma 8: The average diameter of $MT_{n,k}$ is $\frac{nk}{4}$ for k even, and $\frac{n(k^2-1)}{4k}$ for k odd.

Proof: The distance of a node v of $MT_{n,k}$ from node 0^n, denoted by $d(v)$ is:

$$d(v) = \sum_{i=0}^{n-1} |v_i|, \qquad \text{where} \quad |v_i| = \begin{cases} v_i, & \text{if } v_i \leq \lfloor\frac{k}{2}\rfloor, \\ k - v_i, & \text{otherwise.} \end{cases}$$

To estimate the average diameter we sum the above expression for all nodes v of $MT_{n,k}$ and divide by k^n.

For k odd, the average diameter of $MT_{n,k}$ is:

$$\frac{1}{k^n} \sum_v \sum_{i=0}^{n-1} |v_i| = \frac{1}{k^n} \sum_{i=0}^{n-1} \sum_v |v_i| = \frac{1}{k^n} n2k^{n-1} \sum_{i=1}^{\frac{k-1}{2}} i = \frac{n(k^2-1)}{4k}.$$

For k even, the average diameter of $MT_{n,k}$ is:

$$\frac{1}{k^n} \sum_v \sum_{i=0}^{n-1} |v_i| = \frac{1}{k^n} \sum_{i=0}^{n-1} \sum_v |v_i| = \frac{1}{k^n} nk^{n-1} \left(2\sum_{i=1}^{\frac{k}{2}} i - \frac{k}{2} \right) = \frac{nk}{4}.$$

□

If we substitute the average diameter of $MT_{n,k}$ in expression (1), we conclude that a lower bound for the number of message transmissions required for the single node scattering problem on $MT_{n,k}$ is $\lceil \frac{Mnk^{n+1}}{4} \rceil$ for k even, and $\lceil \frac{Mn(k^2-1)k^{n-1}}{4} \rceil$ for k odd.

A multinode scattering problem can be viewed as k^n single node scattering problems, one from each node of $MT_{n,k}$. 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 $\lceil \frac{Mnk^{2n+1}}{4} \rceil$ for k even and $\lceil \frac{Mn(k^2-1)k^{2n-1}}{4} \rceil$ for k odd. Each node has $2n$ incident edges and at most $2nk^n$ message transmissions can be performed at each time step. Consequently, a lower bound for the number of time steps required for this problem is $\lceil \frac{Mk^{n+1}}{8} \rceil$ for k even and $\lceil \frac{M(k^2-1)k^{n-1}}{8} \rceil$ for k odd.

Table 5 summarizes the lower bounds for all of the above problems.

Problem	Time steps	Message transmissions
Single node broadcasting	$\lceil \frac{M}{2n} \rceil + (n-1)\lfloor \frac{k}{2} \rfloor + k - 2$	$M(k^n - 1)$
Multinode broadcasting	$\lceil \frac{M(k^n-1)}{2n} \rceil$	$Mk^n(k^n - 1)$
Single node scattering	$\lceil \frac{M(k^n-1)}{2n} \rceil$	$\lceil \frac{Mnk^{n+1}}{4} \rceil$, if k is even $\lceil \frac{Mn(k^2-1)k^{n-1}}{4} \rceil$, if k is odd
Multinode scattering	$\lceil \frac{Mk^{n+1}}{8} \rceil$, if k is even $\lceil \frac{M(k^2-1)k^{n-1}}{8} \rceil$, if k is odd	$\lceil \frac{Mnk^{2n+1}}{4} \rceil$, if k is even $\lceil \frac{Mn(k^2-1)k^{2n-1}}{4} \rceil$, if k is odd

Table 5: Lower bounds on the $MT_{n,k}$ 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 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 starts to forward 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 $MT_{3,3}$ can be seen in Fig. 4.

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 $M + n\lfloor \frac{k}{2} \rfloor - 1$, 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.

Figure 4: Single node broadcasting on the $MT_{3,3}$ network using BST_{0^n}.

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^n} can be replicated at any other node s of $MT_{n,k}$ 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 split into $\frac{2n}{P}$ subgroups of $\frac{MP}{2n}$ messages each. Node v of the i^{th} spanning tree of BSG_s transmits 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 transmits 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 transmits 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

Figure 5: Multinode broadcasting on the $MT_{3,3}$ network using BSG_{0^n}.

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 $2n$ 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 $MT_{n,k}$, the $2n$ directed edges in each $E_i(s)$, are obtained as a translation with respect to s of the $2n$ directed edges of $E_i(0^n)$. According to lemma 5, if at each time step i, the $2n$ directed edges in $E_i(0^n)$ are all of different dimensions, then the sets of $2n$ directed edges $E_i(s)$, for s ranging over all nodes of $MT_{n,k}$ 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 $MT_{n,k}$, since each BSG_s is a translation with respect to s of BSG_{0^n}. This means that, if an edge (v, u) is used for the transmission of a message originating at node 0^n 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 $MT_{n,k}$ at time step i. At each time step, messages originating at node 0^n are transmitted over $2n$ corresponding directed edges of the $2n$ spanning trees of BSG_{0^n}. From the properties of BSG_{0^n} (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 on the $MT_{2,4}$ network can be seen in Fig. 5. 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 $MT_{n,k}$ receives the M messages originating at any other node once. The number of time steps required is $\lceil \frac{M(k^n-1)}{2n)} \rceil$, 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, $O(\frac{k^n-1}{2n})$.

Figure 6: Single node scattering on the $MT_{3,3}$ network using BSG_{0^n}.

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 $\frac{k^n}{2n}$ nodes. The table includes the nodes of the first spanning tree of BSG_{0^n}, sorted in reverse ordering of their distance from node 0^n. The nodes in the table correspond to the transmission order of the first port of BSG_{0^n}, 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 $\frac{2n}{P}$ paths to node 0^n through BSG_{0^n}. The interesting property of this algorithm is that nodes that belong to nonfull necklaces with period P receive $\frac{MP}{2n}$ of their M messages through each one of the $\frac{2n}{P}$ paths from node 0^n.

The single node scattering algorithm proceeds as follows:

For each node v in the table of $\frac{k^n}{2n}$ entries do the following:

1. If the source node is node 0^n, then it transmits messages destined to nodes v, $R(v)$, $R^2(v)$,...,$R^{2n-1}(v)$, simultaneously. If v belong to a full necklace then all of these nodes are distinct and node 0^n transmits the M messages destined to node $R^i(v)$, $0 \leq i < 2n$, 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 $\frac{2n}{P}$ 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^n has to transmit is split into $\frac{2n}{p}$ subgroups, each containing $\frac{MP}{2n}$ messages. The i^{th}, $0 \leq i < \frac{2n}{P}$, subgroup of the j^{th}, $0 \leq j < P$, group of messages is transmitted over port $iP + j$ of node 0^n. As a consequence, each of the P nodes of a nonfull necklace receives $\frac{MP}{2n}$ of its M messages through each of the $\frac{2n}{P}$ paths from node 0^n through BSG_{0^n}.

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

   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 $T_s(R^i(v))$, $0 \leq i < 2n$, through its incident edges, it starts transmitting messages to nodes $T_s(R^i(v))$, $0 \leq i < 2n$, for the next entry u in the table.

An instance of the single node scattering on $MT_{3,3}$ for messages transmitted from node 0^n to nodes 212 and 121 is shown in Fig. 6, 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, $\lceil \frac{Mnk^{n+1}}{4} \rceil$ for k even and $\lceil \frac{Mn(k^2-1)k^{n-1}}{4} \rceil$ for k odd, 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, $\lceil \frac{M(k^n-1)}{2n} \rceil$.

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, $\lceil \frac{nk^{n+1}}{4} \rceil$ for k even and $\lceil \frac{n(k^2-1)k^{n-1}}{4} \rceil$ for k odd. However, the time is only asymptotically optimal, $O(\frac{k^n-1}{2n})$, 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^n} can be replicated at any other node s of $MT_{n,k}$ 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 $\frac{k^n}{2n}$ nodes. The nodes in the table correspond to the transmission order of the first port of BSG_{0^n}, 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 $\frac{k^n}{2n}$ entries do the following:

1. Source node s determines the destination of the messages to be transmitted over its i^{th}, $0 \leq i < 2n$, 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 $\frac{2n}{P}$ subgroups of $\frac{MP}{2n}$ messages each. The i^{th}, $0 \leq i < \frac{2n}{P}$, subgroup of the j^{th}, $0 \leq j < P$, group of messages is transmitted over the $(iP+j)^{th}$ port of the source.

   We have to mention that the identity of the destination node and a number that indicates the spanning tree of BSG_{0^n} 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 $T_s(R^i(v))$, $0 \leq i < 2n$, then s can transmit messages to nodes $T_s(R^i(u))$, $0 \leq i < 2n$, 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 $d\frac{MP}{2n}$ time steps after the starting transmission time of the current group of messages (P is the period of v).

From the properties of BSG_{0^n}, we know that the $2n$ paths that lead to nodes $R^i(v)$, $0 \leq i < 2n$, through its spanning trees i, $0 \leq i < 2n$, respectively, are rotations of each other (lemma 7), and as a consequence, the $2n$ 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, $2n$ 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, $\lceil \frac{Mnk^{2n+1}}{4} \rceil$ for k even and $\lceil \frac{Mn(k^2-1)k^{2n-1}}{4} \rceil$ for k odd, 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, $\lceil \frac{Mk^{n+1}}{8} \rceil$ for k even and $\lceil \frac{M(k^2-1)k^{n-1}}{8} \rceil$ for k even, 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, $\lceil \frac{nk^{2n+1}}{4} \rceil$ for k even and $\lceil \frac{n(k^2-1)k^{2n-1}}{4} \rceil$ for k odd, is achieved. The time is only asymptotically optimal, $O(\lceil \frac{k^{n+1}}{8} \rceil)$ for k even and $O(\lceil \frac{(k^2-1)k^{n-1}}{8} \rceil)$ for k even, since BST_s is balanced only to within a constant factor.

6 Conclusions

A general framework was developed on the multidimensional torus network, that led to the construction of a shortest path, balanced to within a constant factor, spanning tree, and a shortest path, perfectly balanced 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 application 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 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 conflict 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 efficiently 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, Special Issue on Broadcasting and Gossiping.*

[12] P. Fraigniaud, "Complexity Analysis of Broadcasting in Hypercubes with Restricted Communication Capabilities", *Journal of Parallel and Distributed Computing*, vol. 16, no. 1, 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, vol. 26, no. 5, 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, no. 2, pp. 167-181, 1993.