SINGLE-PORT MULTINODE BROADCASTING

V.V. DIMAKOPOULOS

1-99

Preprint no. 1-99/1999

Department of Computer Science
University of Ioannina
451 10 Ioannina, Greece

Single-Port Multinode Broadcasting[*]

Vassilios V. Dimakopoulos

Technical Report 1-99

Department of Computer Science, University of Ioannina
P.O. Box 1186, GR-45110 Ioannina, Greece
Tel: +30 651 97309, Fax: +30 651 48131
E-mail: dimako@cs.uoi.gr

February, 1999

Abstract

Multinode broadcasting, an important collective communication problem, involves simultaneous broadcastings from all the nodes in a network. In this work we present new results and algorithms for the minimum-time solution of the problem in packet-switched networks that follow the single-port model. In particular, we show that the problem can be optimally solved in *any* network that has a hamiltonian cycle; most of the popular multiprocessor interconnection networks possess this property. We also construct a general algorithm for the solution of the problem in *arbitrary* multidimensional networks and provide conditions that ensure its optimality.

Keywords: collective communications, hamiltonian graphs, multidimensional networks, multinode broadcasting

[*] Submitted for publication

1. Introduction

The advent of distributed-memory multiprocessors has spawned an increasing amount of research in information dissemination problems. Given a network of processors (or *nodes*) where some of them own pieces of information, the problem is to spread the information to a group of recipients using the links of the network. The term 'collective communications' has been coined to signify the fact that such problems involve more than two nodes.

Among the variety of situations that can arise in practice (e.g. in parallel numerical algorithms [7, 1]) broadcasting and multinode broadcasting intrigued the research community as early as the 1950's [8]. In *broadcasting* there is one node that owns a piece of information (hereafter called 'message' or 'packet') and needs to send it to the other nodes in the network. *Multinode broadcasting*, which is the subject of this paper, involves simultaneous broadcastings from all nodes, that is, every node has to send a single message to all the other nodes in the network.

Two surveys on collective communication problems, including multinode broadcasting, were given in [8, 6]. Multinode broadcasting is also known as *all-to-all broadcasting* and *gossiping*. Traditionally, though, the term 'gossiping' implies certain assumptions about the communication cost. In particular, it is assumed that whenever two neighboring nodes communicate (known as a 'call') they can exchange any number of packets in one time unit. Such a model is not suited for our purposes here; we are interested in obtaining the minimum possible *time* needed to solve the problem, instead of the minimum number of calls. Clearly, in practice the more packets two nodes exchange the more the time delay will be. As a consequence, we will make the more realistic assumption that in one time unit a node can only send one packet.

We are going to consider packet-switched networks that follow the constant communication model [6] where:

- communication links are bidirectional
- a message requires one time unit (or *step*) to be transfered between two nodes
- only adjacent nodes can exchange messages.

Furthermore, nodes will be assumed to have *single-port* capabilities, that is, a node can only send and/or receive one message at a time. Under such a scheme two basic possibilities arise based on the nature of the links:

- *full-duplex* where a node can send *and* simultaneously receive a message in a step
- *half-duplex* where a node can send *or* receive a message in a step but not both simultaneously.

We are going to study the multinode broadcasting problem for both duplex models since they yield different solutions. For each of the models we derive lower bounds on the time needed to complete a multinode broadcasting operation and we give algorithms to achieve the optimal times for two large classes of networks. In particular, we show that optimality can be achieved in any network that possesses a hamiltonian cycle. Rings, tori, hypercubes, cube-connected cycles, butterflies and deBrujin graphs [9] are but a few examples of popular interconnection networks which can benefit from our method. To the best of our knowledge, this is the first optimal algorithm for most of these networks under the model we consider.

In addition we provide a general algorithm that solves the multinode broadcasting problem in *any* multidimensional (or cartesian product) network in a modular way. Such networks are probably the most popular as is evident from their utilization in commercial parallel machines (e.g. Cray T3D, Intel Paragon). For multidimensional networks a modular solution for the *total exchange* communication problem was given in [5]. Assuming we are given multinode broadcasting algorithms for each of the dimensions, we show how to construct an algorithm for the multidimensional graph. We also prove that if the algorithms for each of the dimensions are time-optimal, then the derived algorithm can also be optimal.

The paper is organized as follows: in Section 2 we derive the lower bounds for our problem in both models. We then proceed to give a simple optimal solution for rings and general hamiltonian graphs in Section 3. For multidimensional (or cartesian product) networks we give a general algorithm in Section 4 and derive conditions under which it behaves optimally. Finally, Section 5 summarizes the results.

2. Lower Bounds

Let us consider a network (or graph) $G = (V, E)$ where V is the node (or vertex) set and E is the link (or edge) set. An edge e in E connecting nodes v and u is written as the unordered pair $e = (v, u)$ and nodes v and u are said to be *adjacent* to each other or just *neighbors*. We will finally assume that the number of nodes is $n = |V|$.

In the multinode broadcasting problem each node will receive $n - 1$ messages, one from each of the other nodes. Under the single-port assumption messages at any node can only arrive one by one. Consequently, there will be at least $n - 1$ time units required to receive all those messages. If T_{FD}, T_{HD} are the times needed to perform multinode broadcasting in G under the full- and half-duplex models, we have the lower bound of:

$$T_{HD}, T_{FD} \geq n - 1 \quad \text{steps.} \tag{1}$$

For the case of half-duplex links the lower bound can be further tightened. Each of the n broadcast messages must be received by $n - 1$ nodes. In other words, each message implies

$n-1$ receptions. For the receptions to occur, there must clearly occur $n-1$ transmissions, too. In total, for all n messages there will occur:

$$n(n-1) \text{ receptions and } n(n-1) \text{ transmissions.}$$

If the number of nodes is *even* then at most n actions (receptions or transmissions) can occur at each step. To make it clearer, since no node is allowed to simultaneously receive and transmit messages, at most half of the nodes can send a message and at most half of the nodes can receive a message. As a result, at most $n/2$ transmissions and $n/2$ receptions may be had at each step. This gives the lower bound of:

$$T_{HD}^{even} \geq 2(n-1). \tag{2}$$

If n is odd then at most $n-1$ actions (receptions or transmissions) can occur at each step, that is, one node can not participate at all, giving the lower bound of:

$$T_{HD}^{odd} \geq 2n. \tag{3}$$

3. Multinode Broadcasting in Hamiltonian Networks

A graph is hamiltonian if it contains a hamiltonian cycle, that is, a cycle that passes through every node exactly once. A ring is trivially hamiltonian. Another well-known example is the hypercube; a hamiltonian cycle can be constructed through the use of Gray codes. Networks like cube-connected cycles, butterflies and deBrujin graphs are also known to contain a hamiltonian cycle [9]. The same is true for graphs that are cartesian products of hamiltonian graphs [3]. Thus, k-ary n-cubes and general tori are hamiltonian, being products of simple rings.

Clearly, an n-node hamiltonian graph G contains an n-node ring as a subgraph. What we show here is that in any ring and under any of the models we consider, the lower bounds of (1)–(3) are tight, i.e. multinode broadcasting can be performed in optimal time. As a result, and since the bounds of (1) - (3) depend only on the number of nodes, *multinode broadcasting can be performed optimally in any hamiltonian graph through the links of its hamiltonian cycle.* It is therefore enough to construct optimal algorithms for general rings, and this we do in the following subsections.

3.1. Rings with full-duplex links

Formally, an n-node ring has node set $V = \{0, 1, \ldots, n-1\}$ and node i is adjacent to nodes $i \oplus 1$ and $i \ominus 1$, where $\oplus/\ominus$ are the operations of addition/subtraction modulo n.

If the network follows the full-duplex model, a very simple algorithm can be employed to perform multinode broadcasting in the minimum time of Eq. (1). Each node is assumed to have a buffer which initially holds the message it has to broadcast. At each step node i (for all $i = 0, 1, \dots, n-1$) sends the message contained in the buffer to neighbor $i \oplus 1$ and simultaneously receives a message from neighbor $i \ominus 1$. The received message is placed in the buffer.

It is trivial to see that at each step messages are rotated around the ring; after $n-1$ steps all messages will have passed through node i (for all i), thus concluding the operation. An example is given in Fig. 1 for $n = 4$ nodes. We have followed the notation m_i to denote the broadcast message of node i.

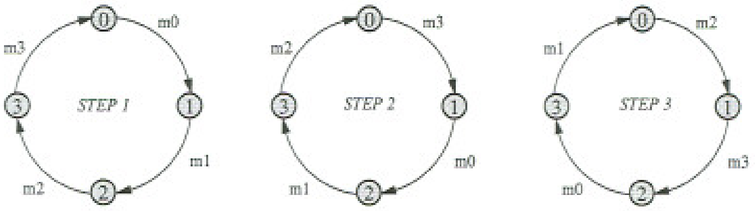


Figure 1. *Full duplex multinode broadcasting in a 4-node ring*

3.2. Rings with half-duplex links

3.2.1. Even number of nodes

If the number of nodes is even, it can be easily seen that the lower bound of $2(n-1)$ steps under the half-duplex model can be achieved by emulating the algorithm for the full-duplex case. Each step of the latter algorithm is emulated by two steps in the half-duplex case as follows: in the first step we let even-numbered nodes perform transmissions (and hence odd-numbered nodes are the receptors) of their message; in the second step the situation is reversed, having odd-numbered nodes transmit. Thus, $2(n-1)$ steps will be needed in total, which is optimal as seen by (2). An example for $n = 4$ nodes is given in Fig. 2. Notice that because in odd-numbered steps some nodes receive a message without sending the one they already hold, two-message buffers (FIFO) are needed.

3.2.2. Odd number of nodes

In a ring with an odd number of nodes, we cannot directly emulate the algorithm for the full-duplex case because there must exist one node at each step that does not participate at all. Nevertheless, it turns out that one can use another simple algorithm that guarantees optimal

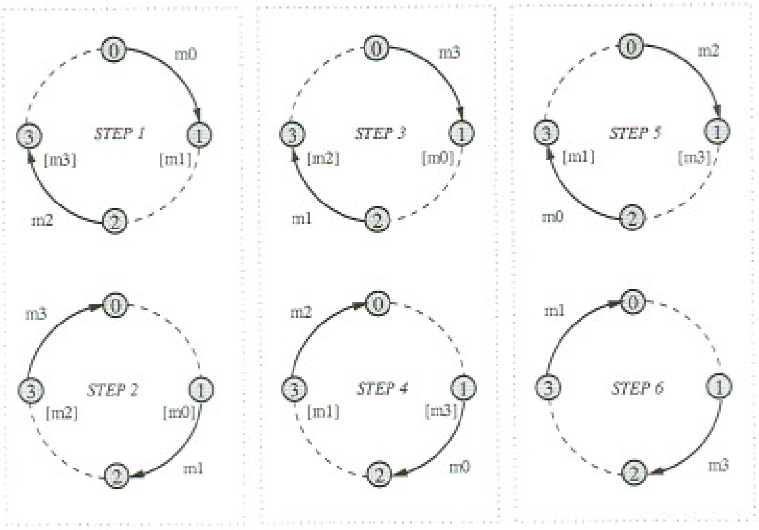


Figure 2. *Half duplex multinode broadcasting in a 4-node ring*

time. The algorithm is best understood through the example in Fig. 3 for $n = 5$ nodes.

As in the case of even rings, each node has a two-message buffer with a first-in first-out (FIFO) discipline. In the first step the transmitting nodes are nodes 1, 3, ..., $n-2$. Node 0 does not participate at all and the rest of the nodes are the receivers. After each step the pattern is rotated around the ring, e.g. in the second step, nodes 2, 4, ..., $n-1$ are the transmitters. In general, in the jth step ($1 \leq j \leq 2n$) the transmitters are nodes $j \oplus 0$, $j \oplus 2$, $j \oplus 4$, ..., $j \oplus (n-1)$, node $j \ominus 1$ does not participate and the remaining nodes are the receivers. The algorithm proceeds this way for $2n$ steps in total. Notice that during those $2n$ steps the pattern repeats itself twice. In Fig. 3 the receivers/transmitters pattern rotates around the ring for 5 steps and then repeats itself from the 6th step till the final 10th step.

Theorem 1 *The algorithm is an optimal multinode broadcasting algorithm for odd rings with half-duplex links.*

The proof is rather involved and, for presentation purposes, it is given in the Appendix.

4. Multinode Broadcasting in Multidimensional Networks

In this section we are going to develop a general multinode broadcasting algorithm for *any* multidimensional graph. Although simple, the algorithm will be shown to be optimal if certain conditions are met.

Given k graphs $G_i = (V_i, E_i)$, $i = 1, 2, \ldots, k$, their (cartesian) product is defined as the graph $G = G_1 \times \cdots \times G_k = (V, E)$ whose vertices are labeled by a k-tuple $(v_1, \ldots, v_k)$, called

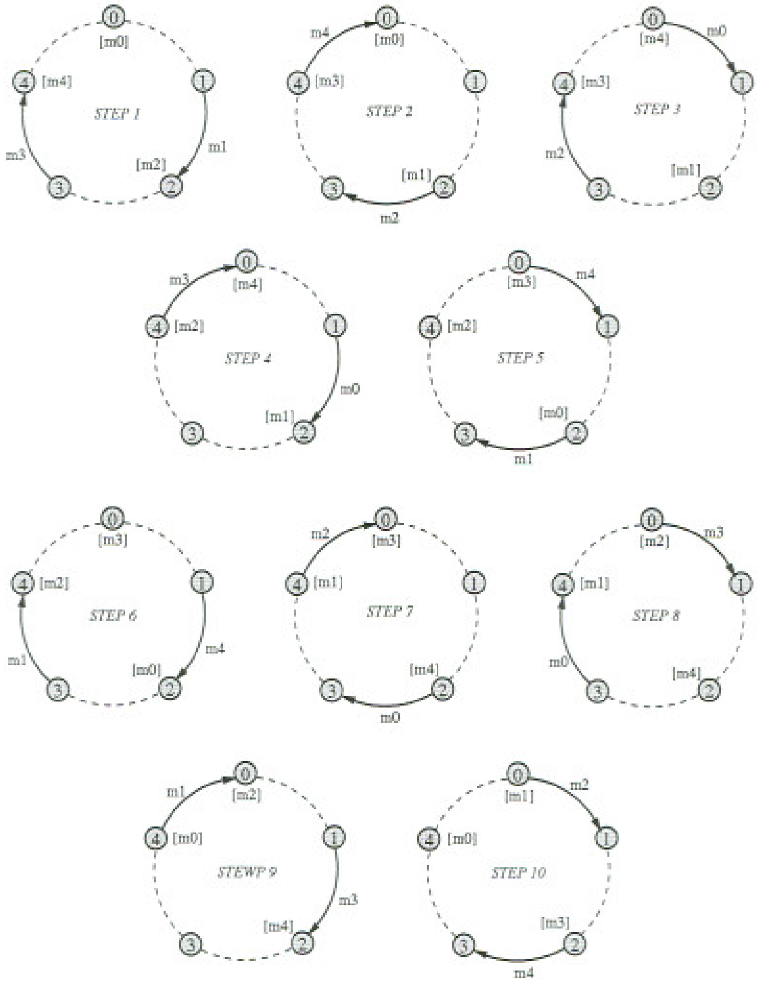


Figure 3. *Half duplex multinode broadcasting in a 5-node ring*

address, and:

$$
\begin{aligned}
V &= \Big\{(v_1,\ldots,v_k) \mid v_i \in V_i, i = 1,\ldots,k\Big\} \\
E &= \Big\{\big((v_1,\ldots,v_k),(u_1,\ldots,u_k)\big) \mid \\
&\qquad\qquad \exists j \text{ s.t. } (v_j,u_j) \in E_j \text{ and } v_i = u_i \text{ for all } i \neq j\Big\}.
\end{aligned}
$$

We will call such products of graphs *multidimensional* graphs and G_i will be called the ith *dimension* of the product. The ith component of the address tuple of a node will be called the ith *address digit* or the ith *coordinate*. The definition of E above in simple words states that two nodes are adjacent if they differ in exactly one address digit. Their differing coordinates should be adjacent in the corresponding dimension. An example is given in Fig. 4. Dimension 1 is a graph A consisting of a two-node path with $V_A = \{a, b\}$ while dimension 2 (B) consists of a three-node ring with $V_B = \{0, 1, 2\}$. Their product has node set

$$V = \Big\{(a,0),(a,1),(a,2),(b,0),(b,1),(b,2)\Big\}.$$

According to the definition, node $(a, 0)$ has the following neighbors: since node a is adjacent to node b in the first dimension, node $(a, 0)$ will be adjacent to node $(b, 0)$; since node 0 is adjacent to both nodes 1 and 2 in the second dimension, node $(a, 0)$ will also be adjacent to nodes $(a, 1)$ and $(a, 2)$.

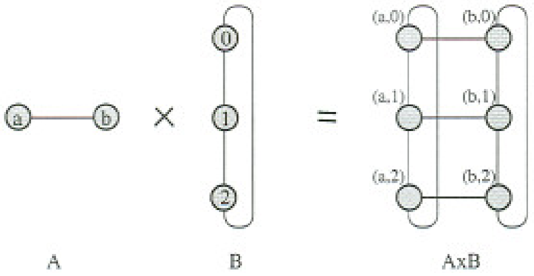


Figure 4. *A two-dimensional graph*

Hypercubes are products of two-node linear arrays (or rings), tori are products of rings. If all dimensions of the torus consist of the same ring, we obtain k-ary n-cubes [4]. Meshes are products of linear arrays [9]. Generalized hypercubes are products of complete graphs [2]. Multidimensional graphs have $n = n_1 n_2 \cdots n_k$ nodes, where $n_i = |V_i|$ is the number of nodes in G_i, $i = 1, 2, \ldots, k$.

It will be convenient to use the *don't care* symbol '$*$' as a shorthand notation for a set of addresses. An appearance of this symbol at an element of an address tuple represents all legal values of this element. In the last example, $(a, *) = \{(a, 1), (a, 2), (a, 3)\}$, $(*, 1) = \{(a, 1), (b, 1)\}$ while $(*, *)$ denotes the whole node set of the graph.

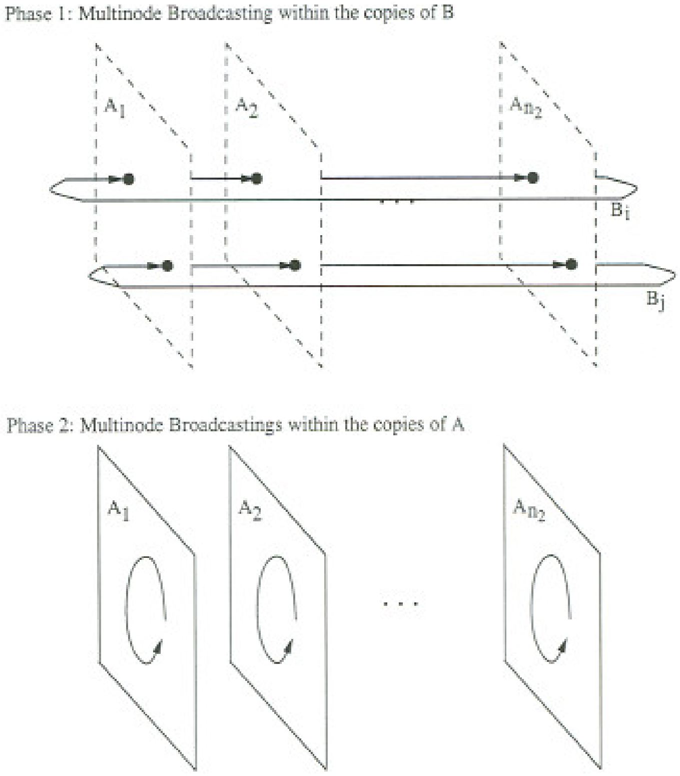
4.1. The Algorithm

Let $G = A \times B$. A k-dimensional network $G_1 \times \cdots \times G_k$ can still be expressed as the product of two graphs by taking $A = G_1$ and $B = G_2 \times \cdots \times G_k$, so we may consider two dimensions without loss of generality. Let $A = (V_A, E_A)$, $B = (V_B, E_B)$, $G = (V, E)$, $n_1 = |V_A|$, $n_2 = |V_B|$ and $n = n_1 n_2$. Finally, let

$$\begin{aligned} V_A &= \{v_i \mid i = 1, 2, \ldots, n_1\} \\ V_B &= \{u_i \mid i = 1, 2, \ldots, n_2\}. \end{aligned}$$

Graph G can be viewed as n_2 (interconnected) copies of A. For example, in Fig. 4 graph $A \times B$ consists of three copies of A, where the corresponding nodes are interconnected according to the edges in B. Let A_j be the jth copy of A with node set $(*, u_j)$, where $*$ takes all values in V_A. Similarly, G can be viewed as n_1 copies of B, and we let B_i be the ith copy of B with node set $(v_i, *)$.

Let us assume that there exist multinode broadcasting algorithms for each of the dimensions. Our method utilizes the algorithms for the dimensions so as to synthesize an algorithm for the whole graph. In other words, the problem of performing multinode broadcasting in $G = A \times B$ is decomposed to the simpler problem of performing multinode broadcasting in A and in B. This is a highly desirable simplification since multidimensional networks are quite complex structures.


Phase 1: Multinode Broadcasting within the copies of B

Phase 2: Multinode Broadcastings within the copies of A

Figure 5. *The two phases of the multinode broadcasting algorithm*

The algorithm we are going to give for G consists of two phases, and is illustrated in Fig. 5. In Phase 1 nodes perform a multinode broadcasting in the second dimension. Notice that because there are no nodes in common between the copies of B what we will describe for a certain copy B_i occurs simultaneously for the other of copies of B, too. Consider node (v_i, u_j) in B_i, and let $m(v_i, u_j)$ denote its own broadcast message. When Phase 1 is complete, this node will contain all broadcast messages from the nodes in B_i, that is, it will contain the messages of nodes $(v_i, *)$: $m(v_i, u_1), m(v_i, u_2), \ldots, m(v_i, u_{n_2})$. This holds for every node in the graph. In particular, every node in A_j will contain n_2 messages, each one originating from a different node. In other words, at the end of Phase 1 all $n = n_1 \times n_2$ broadcast messages will have been received by the n_1 nodes in A_j (for all j), each node holding n_2 of them.

```
Phase 1:
1 Do in parallel for all $B_i$, $i = 1, 2, \ldots, n_1$
2     In $B_i$ perform multinode broadcasting;

Phase 2:
3 For $r = 1, 2, \ldots, n_2$
4     Do in parallel for all $A_j$, $j = 1, 2, \ldots, n_2$
5         In $A_j$ perform multinode broadcasting with
          node $(v_i, u_j)$ broadcasting $m(v_i, u_r)$ $(i = 1, 2, \ldots, n_1)$;
```

Figure 6. *The algorithm for graph* $G = A \times B$

It should now follow clearly that the way to distribute those messages to every node in A_j is through multinode broadcastings within A_j. Thus Phase 2 consists of a series of n_2 multinode broadcastings in the first dimension in order for each node to broadcast the n_2 messages it holds.

The algorithm is summarized in Fig. 6, and is a general solution to the multinode broadcasting problem for any multidimensional network. If the network has $k > 2$ dimensions $G = G_1 \times \cdots \times G_k$, the algorithm can be used recursively by taking $A = G_1$ and $B = G_2 \times \cdots \times G_k$. Phase 1 (lines 1–2) can be performed by invoking the algorithm with $A = G_2$ and $B = G_3 \times \cdots \times G_k$ and so forth.

Notice that the algorithm is independent of the link model in use. Because the two phases are executed one after the other, and within Phase 2 the multinode broadcastings are also executed serially, only one multinode broadcasting operation is in effect at any step. As a result, the whole algorithm is consistent with the link model of the algorithms for each dimensions. For example, if the algorithms for A and B operate on half-duplex links, so does our algorithm for $G = A \times B$.

4.2. Optimality conditions

We proceed now to determine the time required for the general algorithm in Fig. 6 and the conditions under which it behaves optimally. Let T_A and T_B denote the number of steps needed to perform multinode broadcasting in A and B correspondingly.

Theorem 2 *The multinode broadcasting algorithm for $G = A \times B$ requires:*

$$T = T_B + n_2 T_A$$

time units.

Proof. The result is straightforward: Phase 1 performs multinode broadcasting within B_i (for all $i = 1, 2, \ldots, n_1$ in parallel), taking thus time equal to T_B. Phase 2 performs n_2 multinode broadcasting operations within A_j (for all $j = 1, 2, \ldots, n_2$ in parallel), each requiring T_A steps. ■

Theorem 3 *Under the full-duplex model, if multinode broadcasting in A and B can be performed in time equal to the lower bound of Eq. (1) then the same is true for $G = A \times B$.*

Proof. If T_A and T_B achieve the lower bound of Eq. (1) then $T_A = n_1 - 1$ and $T_B = n_2 - 1$. From Theorem 2 we obtain:

$$T = (n_2 - 1) + n_2(n_1 - 1) = n_1 n_2 - 1 = n - 1,$$

as required. ■

Theorem 4 *Under the half-duplex model, if both dimensions have an even number of nodes and multinode broadcasting in each one can be performed in time equal to the lower bound of Eq. (2) then the same is true for $G = A \times B$.*

Proof. If both dimensions have an even number of nodes then by Eq. (2) we must have $T_A = 2(n_1 - 1)$ and $T_B = 2(n_2 - 1)$. From Theorem 2 we obtain:

$$T = 2(n_2 - 1) + 2n_2(n_1 - 1) = 2n_1 n_2 - 2 = 2(n - 1),$$

which is optimal. ■

Theorem 5 *Under the half-duplex model, if only one dimension has an even number of nodes and multinode broadcasting in each dimension can be performed in time equal to the lower bound of Eq. (2) or Eq. (3) then the algorithm for $G = A \times B$ is optimal within two steps.*

Proof. Without loss of generality assume that n_2 is odd and n_1 is even. Otherwise, we might rename the dimensions by considering graph $G = B \times A$ which is isomorphic to $A \times B$. Thus, according to Eqs. (2) and (3) we must have $T_A = 2(n_1 - 1)$ and $T_B = 2n_2$. From Theorem 2 we obtain:

$$T = 2n_2 + 2n_2(n_1 - 1) = 2n_1 n_2 = 2n.$$

Since n is even, the lower bound of Eq. (2) shows that T is suboptimal by 2 steps. ■

5. Conclusion

In this work we studied the problem of multinode broadcasting which involves simultaneous broadcastings from every node in a network. We have considered packet-switched networks with single-port capabilities, whereby a node can at most send and/or receive one message at each step. We note that this is exactly what most present-day machines are capable of. Under this model, we consider the case where simultaneous transmission and reception is allowed at each node (full-duplex) and the case where a node is only allowed to either transmit or receive at each step (half-duplex).

For both duplex cases we obtained optimal algorithms for rings which are easily extended to every hamiltonian graph, such as tori, hypercubes, butterflies, etc. Under the model we consider these algorithms are the only known optimal solutions to the multinode broadcasting problem. Some results in [1] were only approximate for a few of the graphs we considered.

We also provided a general solution to the problem for multidimensional networks. We derived a modular method that utilizes multinode broadcasting algorithms for each of the dimensions. This algorithm applies in any multidimensional network, whether it is hamiltonian or not. Our scheme is simple but it nevertheless maintains optimality always under the full duplex model and in most cases under the half-duplex model. An interesting course of further research is the case where the network is half-duplex and all dimensions contain an odd number of nodes. In this setting our method cannot maintain optimality even if multinode broadcasting in every dimension can be performed optimally.

6. Appendix

Proof of Theorem 1 We are going to prove the theorem by showing that the messages are rotated around the ring and reach their most distant (clockwise) destinations in exactly $2n$ steps. For example, the message of node 2 (m_2) will have reached node 1 at or before the ($2n$)th step, and clearly, on its way it will have passed through (and broadcast to) all nodes in the graph.

Let 'T' denote that a node is transmitting a message, 'R' denote that it is receiving a message and 'N' denote that it is not participating. During the first n steps of the algorithm node 0 follows the behavior pattern:

$$B_0 = \{\text{NRTRT}\cdots\text{RT}\},$$

that is, in the first step it does not participate, in the second step it receives a message, in the third step it sends a message and so on. During the next n steps, it repeats the actions of B_0 once more. In the same way, it can easily be seen that any node i follows the behavior sequence B_i which is the sequence B_0 rotated by i positions to the right. For example, node 1 has:

$$B_1 = \{\mathrm{TNRTRT}\cdots\mathrm{R}\}.$$

What we are going to show is that the n messages are rotated around the ring by $\lfloor n/2 \rfloor$ positions after exactly n steps. That is, node $\lfloor n/2 \rfloor$ will contain the message of node 0 (m_0), node $\lfloor n/2 \rfloor + 1$ will contain m_1 and so on. This is enough to prove the theorem since the remaining n steps are identical; another rotation of the messages by $\lfloor n/2 \rfloor$ positions will advance the messages to their most distant destinations, concluding thus the algorithm.

Let us hence isolate the first n steps. According to the algorithm and the FIFO discipline at the message buffers, no message m_i can advance ahead of message m_{i+1}. Consequently, the clockwise order of messages is preserved. Another observation is that every node i goes through its behavior sequence B_i exactly once. Initially, all nodes contain exactly one message in their buffers, the one they are about to broadcast. But in the behavior sequence, the number of T's is equal to the number of R's, i.e. a node will perform exactly $\lfloor n/2 \rfloor$ receptions and $\lfloor n/2 \rfloor$ transmissions. Based on this, and because there are no consecutive R's or T's in the behavior sequence, we reach the conclusion that after n steps every node will end up again with exactly one message in its buffers.

Summarizing, we have shown that after n steps the messages will have rotated around the ring; every node will contain exactly one message and the clockwise order will be preserved. Let us say that the messages advanced (rotated) by d positions, that is, m_i will reside at node $i \oplus d$. The only remaining issue is to prove that $d = \lfloor n/2 \rfloor$.

Each message must travel to the most distant node in the clockwise direction. This means that it will eventually cross $n-1$ links in total. Thus the *total distance* for all messages to reach their final receptors is $TD = n(n-1)$. Each transmission by a node advances a message one step closer to its final receptor and consequently decreases the remaining total distance by 1. Because in the first n steps of the algorithm each node performs exactly $\lfloor n/2 \rfloor$ transmissions, the remaining total distance after those n steps will be equal to $RTD = TD - n\lfloor n/2 \rfloor = n\lfloor n/2 \rfloor$.

If the messages had rotated by d positions then each one would have been $n-1-d$ links away from its final destination. Consequently, the remaining total distance would have been equal to $n(n-1-d)$. Equating this quantity with RTD gives the desired result of $d = \lfloor n/2 \rfloor$.

References

[1] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods* (Prentice - Hall, Englewoods Cliffs, N.J., 1989).

[2] L. N. Bhuyan and D. P. Agrawal, Generalized hypercube and hyperbus structures for a computer network, *IEEE Trans. Comput.* **C-33** (1984) 323–333.

[3] F. Buckley and F. Harary, *Distance in Graphs* (Addison - Wesley, Reading, Mass., 1990).

[4] W. J. Dally and C. L. Seitz, Deadlock-free message routing in multiprocessor interconnection networks, *IEEE Trans. Comput.* **C-36** (1987) 547–553.

[5] V. V. Dimakopoulos and N. J. Dimopoulos, A theory for total exchange in multidimensional interconnection networks, *IEEE Trans. Parall. Distrib. Syst.* **9** (1998) 639–649.

[6] P. Fraigniaud and E. Lazard, Methods and problems of communication in usual networks, *Discrete Appl. Math.* **53** (1994) 79–133.

[7] D. B. Gannon and J. van Rosendale, On the impact of communication complexity on the design of parallel numerical algorithms, *IEEE Trans. Comput.* **C-33** (1984) 1180–1194.

[8] S. M. Hedetniemi, S. T. Hedetniemi and A. L. Liestman, A survey of gossiping and broadcasting in communication networks, *Networks* **18** (1988) 319–349.

[9] F. T. Leighton, *Introduction to Parallel Algorithms and Architectures: Arrays, Trees, Hypercubes* (Morgan Kaufmann, San Diego, CA, 1992).