# Broadcasting in wraparound meshes with parallel monodirectional links

Jean-Claude Bermond, Philippe Michallon, Denis Trystram

# Broadcasting in wraparound meshes with parallel monodirectional links

Jean-Claude Bermond [a], Philippe Michallon [b] and Denis Trystram [b]

[a] *I3S-CNRS, bât. 4, 250 rue Albert Einstein, Sophia-Antipolis 06 560 Valbonne, France*
[b] *LMC-IMAG, 46 avenue Félix Viallet, 38031 Grenoble Cedex, France*

In this paper we give an algorithm to broadcast a message in a wraparound mesh distributed-memory parallel architecture with parallel monodirectional links. This algorithm uses a general strategy based on the diffusion of the message in edge-disjoint spanning trees. We first present in this setting the results of Saad and Schultz and the improvements obtained by Simmen. We then give an asymptotically optimal broadcasting algorithm improving the preceding results. It uses in the wraparound mesh the constructions of two edge-disjoint spanning trees rooted at a given node and of minimum depth.

*Keywords.* Communication; interconnection networks; distributed memory architectures; wraparound mesh; toroidal grid; broadcasting; spanning trees.

## 1. Introduction

### *1.1. Description of the reference models*

In distributed-memory parallel computers, the communications are a bottleneck and so, very efficient algorithms have to be designed for global communication schemes. In what follows, we assume that the communication links between processors are mono-directional (half-duplex mode [5]). We suppose furthermore that a processor can simultaneously send (or receive) a message on all its links (parallel communications). We also suppose, according to the literature, that the transmission time of a message of length $L$ (number of bytes) from a processor to one of its direct neighbours is of the form: $\beta + L\tau$ (where $\beta$ corresponds to a start-up time and $\tau$ is the inverse of the bandwidth). For multiprocessors like transputer-interconnected architectures [8], $\beta$ is of the same order of magnitude as $\tau$. In other multiprocessors, $\beta$ can be much greater than $\tau$. Let us note that this notation is the most commonly used and is different from [5,6] where the start-up is denoted $\tau$and the bandwidth is denoted $\beta$.

Finally, we consider the usual store-and-forward model for routing where any byte has to be stored in any intermediate processor before being transmitted to its final destination. One

* This work is supported by the 'operation RUMEUR' of the GDR $C^3$.
*Correspondence to:* Denis Trystram, LMC-IMAG, 46 avenue Félix Viallet, 38031 Grenoble Cedex, France.

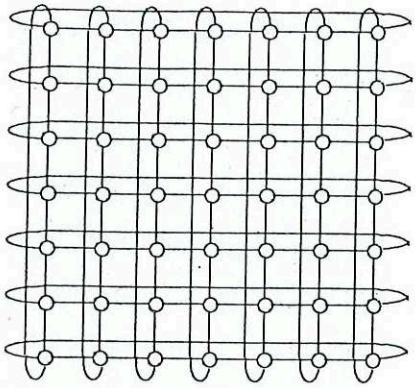

Fig. 1. Basic wraparound mesh.

of the basic communication routines (used in many parallel algorithms) is broadcasting. Broadcasting (often referred as One-To-All) consists of sending a message from a given processor called the initiator or root to all the other processors. Broadcasting algorithms have been designed for many regular topologies of interconnection networks (trees, hypercubes, rings, wraparound meshes etc.) [1–5,7,9].

### *1.2. Definitions and basic properties of wraparound meshes*

Wraparound meshes (also called toroidal grids) are among the most popular interconnection networks proposed for distributed-memory parallel computers. In the following, we will consider square wraparound meshes although all our results c ould be extended to rectangular ones. An $n$ by $n$ wraparound mesh consists of $p = n^2$ processors interconnected in a toroidal grid graph. We can simply label a processor by a couple of indices $(i, j)$ taken modulo $n$ corresponding to the Cartesian coordinates on the grid. More precisely, processor $(i, j)$ is joined to 4 neighbours: east $(i + 1, j)$, north $(i, j + 1)$, west $(i - 1, j)$ and south $(i, j - 1)$. The distance between two processors is the length of a shortest path between them. The maximum of the distances between any pair of processors is the diameter. It is easy to show that the diameter $D$ of an $n$ by $n$ waparound mesh is $n - 1$ if n is odd and $n$ if $n$ is even, that is $D = 2\lfloor n/2 \rfloor$. *Figure 1* shows a $7 \times 7$ wraparound mesh.

### *1.3. Organization of the paper*

The paper is organized as follows: In Section 2 we first recall the general broadcasting principle proposed in [3] which uses edge-disjoint spanning trees rooted at the initiator. We present in this context the result of Saad and Schultz [5], which has been corrected and improved by Simmen [6]. In Section 3 we design optimal algorithms for square wraparound meshes based on the construction of 2 edge-disjoint spanning trees of minimum depth rooted at any given vertex. Before concluding, we give in Section 4 experimental results obtained on the transputer-based parallel computer (namely, the MegaNode [8]).

## 2. General setting and existing broadcasting algorithms

### *2.1. General algorithm*

The following general algorithm is described in [1,3]. It consists of sending a message on edge-disjoint spanning trees by using pipelining. The analysis of this algorithm shows that

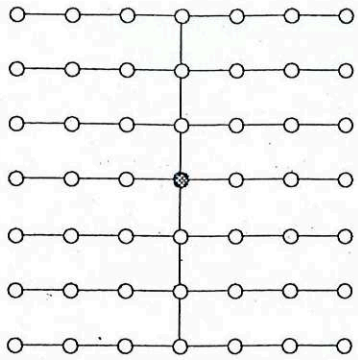

Fig. 2. Principle of the basic pipelined broadcast for the wraparound mesh.

there exists an optimal size of the packets in which the message of length $L$ should be divided.

**Proposition.** *The time to broadcast a message of length L using pipelining on d edge-disjoint spanning trees of depth h rooted at the initiator is*:

$$b(L, h, d) = \left(\sqrt{(h-1)\beta} + \sqrt{\frac{\tau L}{d}}\right)^2.$$

*2.2. A first solution*

The first idea to broadcast a message on a wraparound mesh is to first send it on a pipelined ring on the vertical dimension of the wraparound mesh, then to send it on the ring along the horizontal dimension. This can be interpreted as a pipeline on a spanning tree as depicted in *Fig. 2*. The depth of this tree is minimum or equal to the diameter D. Using the previous proposition, the broadcast time is:

$$b(L, D, 1) = \left(\sqrt{(D-1)\beta} + \sqrt{\tau L}\right)^2.$$

However, in this solution only one tree is used because of link conflicts and so half of the links are not used.

*2.3. Saad and Schultz's algorithm*

Saad and Schultz propose in [5] a better pipelined broadcasting algorithm. The initial message is split into 2 sub-messages of size $L/2$ which are each pipelined on two 'almost spanning trees' depicted in *Fig. 3*. (Note that the union of these trees covers most links and that they are edge-disjoint which allows two simultaneous pipelined broadcasts of $L/2$ data.)

The two almost spanning trees are obtained by rotation of $\pi/2$ from each other. The depth of these trees is as before equal to $D$ but now we can use two trees. All the nodes are covered except the ones belonging to the vertical line (which have received only half of the initial message) and similarly for the second tree the nodes of the horizontal line (which have received the complementary half). However, Saad and Schultz propose to overlap the empty phase of the pipeline to fill-in the incomplete horizontal and vertical lines. As soon as the last packet has been sent, the missing data (of size $L/2$) are sent using a pipeline mode with the same number of packets as before. Unfortunately and contrary to what is claimed in [5] it can be shown (see [6]) that for large $L$ this time is greater than the time needed to empty the

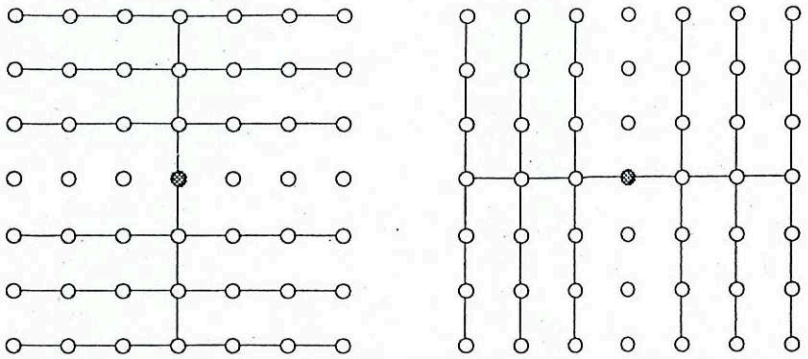

Fig. 3. The 2 almost spanning trees of the wraparound mesh.

pipeline! More precisely, a detailled calculus leads to the following result:
For $L \leq L_0$ (where $L_0 = \beta/2\tau[D^2/(D-1)]$), the fill-in can be done during the empty phase. The time of this algorithm is:

$$b(L, D, 2) = \left(\sqrt{(D-1)\beta} + \sqrt{\tau L/2}\right)^2.$$

For $L \geq L_0$, the fill-in can not be done during the empty phase. The time becomes:

$$b(L, D, 2) + \frac{\tau L}{2} + \sqrt{\frac{\beta\tau L(D-1)}{2}} - \frac{D}{2}\left(\beta + \sqrt{\frac{\beta\tau L}{2(D-1)}}\right).$$

*2.4. Simmen's algorithm*

In his paper, Simmen [6] proposed a new broadcasting algorithm which can be considered in our general setting as using two edge-disjoint spanning trees of depth $2D+1$. The first tree is depicted in *Fig. 4*, the other one being obtained by a rotation of $\pi/2$. The broadcast time is now:

$$b(L, 2D+1, 2) = \left(\sqrt{2D\beta} + \sqrt{\frac{\tau L}{2}}\right)^2.$$

Note that for large $L$ this algorithm is better than the first one.

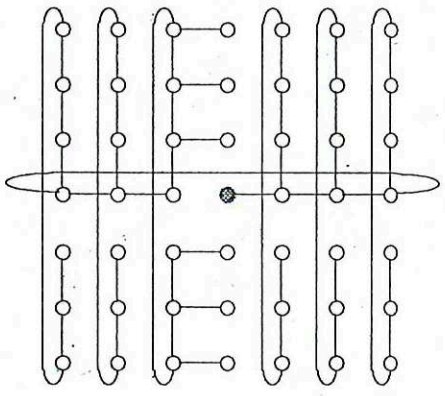

Fig. 4. One of the two spanning trees of Simmen.

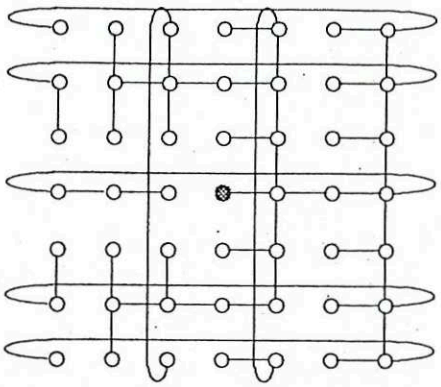

Fig. 5. One of the form arc disjoint spanning trees of depth $D+1$ for the bidirectional case.

## 3. New broadcasting algorithms

### *3.1. Improvement using alternatively bidirectional arc disjoint spanning trees*

In [4], Michallon et al. constructed four arc-disjoint spanning trees of depth $D+1$ under the hypothesis of bidirectional communication links (see *Fig. 5* for one of these trees, the others are obtained by sucessive rotations of $\pi/2$). The orientations of the arcs are omitted. So we have to use two steps to transmit a packet from one processor to a neighbour. Thus, we get a result similar to the proposition by replacing $h$ by $2h$ (in this case $2D+2$). We obtain the following time:

$$b(L, 2(D+1), 4) = \left(\sqrt{(2D+1)\beta} + \sqrt{\frac{\tau L}{4}}\right)^2,$$

which is better than Simmen's algorithm.

### *3.2. Improvement using edge-disjoint spanning trees*

We show in this section how to construct in a wraparound mesh two edge-disjoint spanning trees of depth better than those obtained by Simmen.

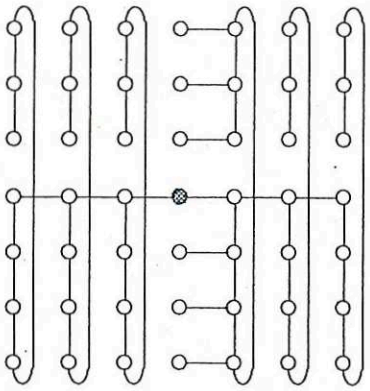

Fig. 6. One of the two edge disjoint spanning tree of depth $3D/2$.

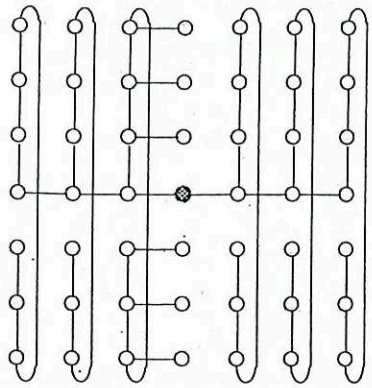

Fig. 7. One of the two edge disjoint spanning trees of depth $3D/2$ (modification of Simmen's trees).

A first easy improvement is obtained by modifying the trees of Saad and Schultz and Simmen (see *Figs 6* and *7*). We obtain two edge-disjoint spanning trees of depth $3D/2$ instead of $2D+1$ which leads to the time:

$$b\left(L,\frac{3D}{2},2\right)=\left(\sqrt{\left(\frac{3D}{2}-1\right)\beta}+\sqrt{\frac{\tau L}{2}}\right)^2.$$

The first tree is depicted in *Fig. 6*. The second one (*Fig. 7*) is obtained by rotation of $\pi/2$. Let us remark that these two constructions yield the same trees (symmetric with respect to the origin).

This can be improved as we have been able to find two edge-disjoint spanning trees of minimum depth as described in the following theorem.

**Theorem.** *There exist in an n by n wraparound mesh two edge-disjoint trees of minimum depth n.*

**Proof.** first note that the trees have a depth of at least $D$, so the result is optimal in the case $n$ even, as $D=n$. In the case of $n$ odd, the diameter is $n-1$ but it is impossible to build two edge-disjoint spanning trees of depth $D$. There are four vertices, namely $(\pm(n-1)/2, \pm(n-1)/2)$, which are at distance $D$ from the origin and furthermore there are four edges between these four vertices. These edges cannot be used in a spanning tree of depth $D$. So we can use at most $2n^2$-4 edges for the two spanning trees. But $n^2-1$ edges are needed in a spanning tree, so we get a contradiction.

To obtain the trees of optimum depth we use the following technique. Like in the preceding example we construct only one tree, the second one being obtained by a rotation of $\pi/2$. Furthermore we impose that this tree is symmetric with respect to the origin. Then we split the mesh into four parts which are as equal as possible. The first tree will contain all the vertices of the 0-row (vertices $(i, 0)$) and the second tree will contain the vertices of the 0-column (vertices $(0, j)$). For the first tree we take one column over two in the first domain and the other column in domain 2. Then we attain the other vertices by subpaths of the form $\lfloor$ or $\rceil$ except around the origin and on the borders of the figure where it depends on the congruence of $n$ modulo 4. *Figure 8* shows the general technique and *Figs. 9–12* give the construction for $n=10, 11, 12, 13$. This fills roughly one edge over two in the column not already used. All together the first tree uses around $\frac{3}{4}$ of the vertical and $\frac{1}{4}$ of the horizontal edges, and by rotation the second tree uses $\frac{3}{4}$ of the horizontal and $\frac{1}{4}$ of the vertical edges. These constructions can be easily extended to any value of $n$ by inserting four intermediate rows and four intermediate columns (two in each domain) of the following form:

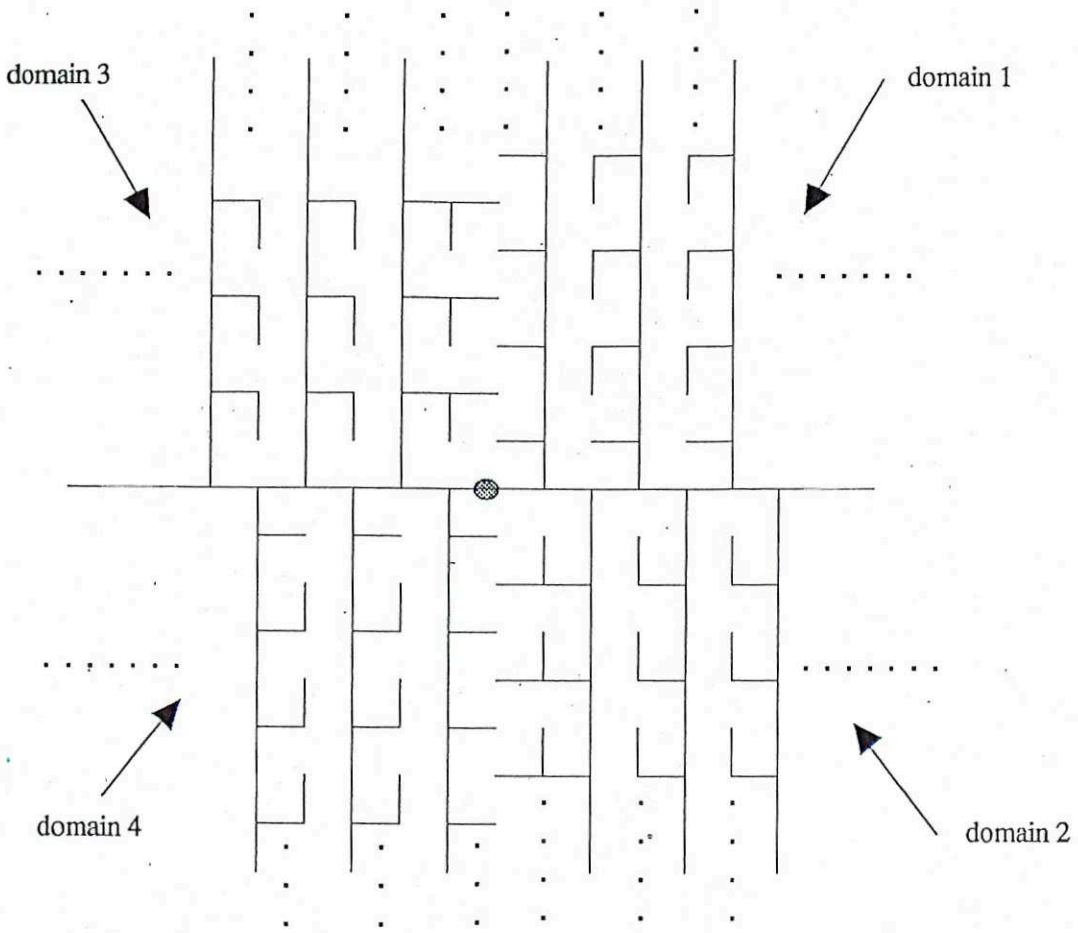


Fig. 8. Construction of the first tree.

The construction of these spanning trees can be extended to non-square wraparound meshes using the same principle.

The broadcast time is calculated using the basic formula:

$$b(L, n, 2) = \left(\sqrt{(n-1)\beta} + \sqrt{\frac{L}{2}\tau}\right)^2. \qquad \square$$

## 4. Experimental results

We report in this section some experiments on the broadcasting algorithm based on the use of our new family of edge-disjoint spanning trees. *Figure 13* gives the time related to the message length for various sizes of square wraparound meshes. The experiments have been done on a MegaNode which is a distributed-memory parallel machine with 128 transputers [8]. For such a machine, $\beta$ is equal to 628 $\mu$sec and $\tau = 2{,}2$ $\mu$sec/byte [4].

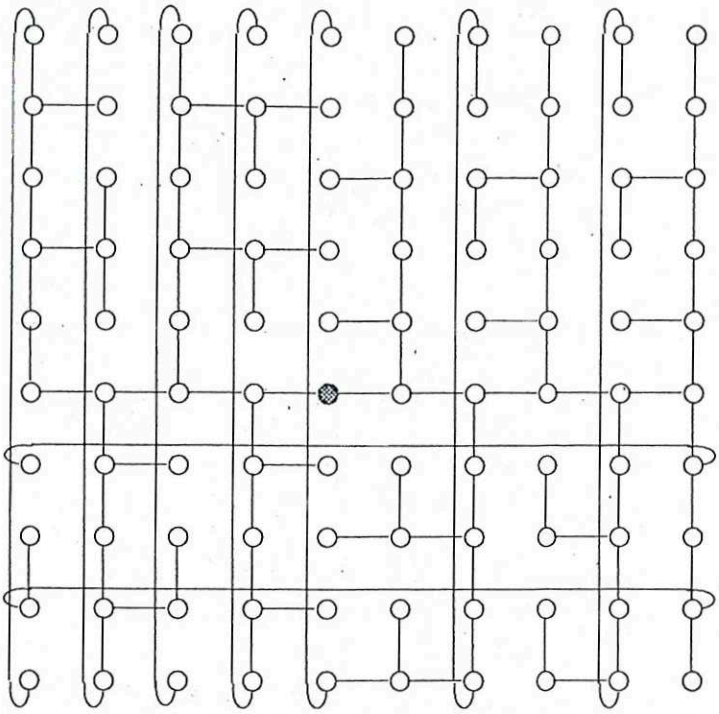

Fig. 9. Construction of the first tree for $n = 10$.

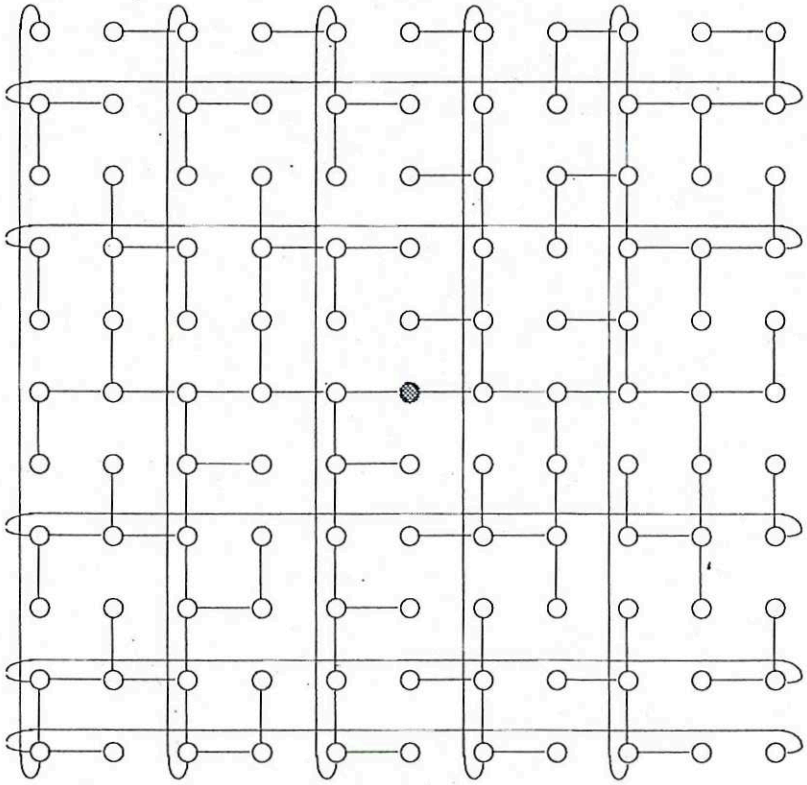

Fig. 10. Construction of the first tree for $n = 11$.

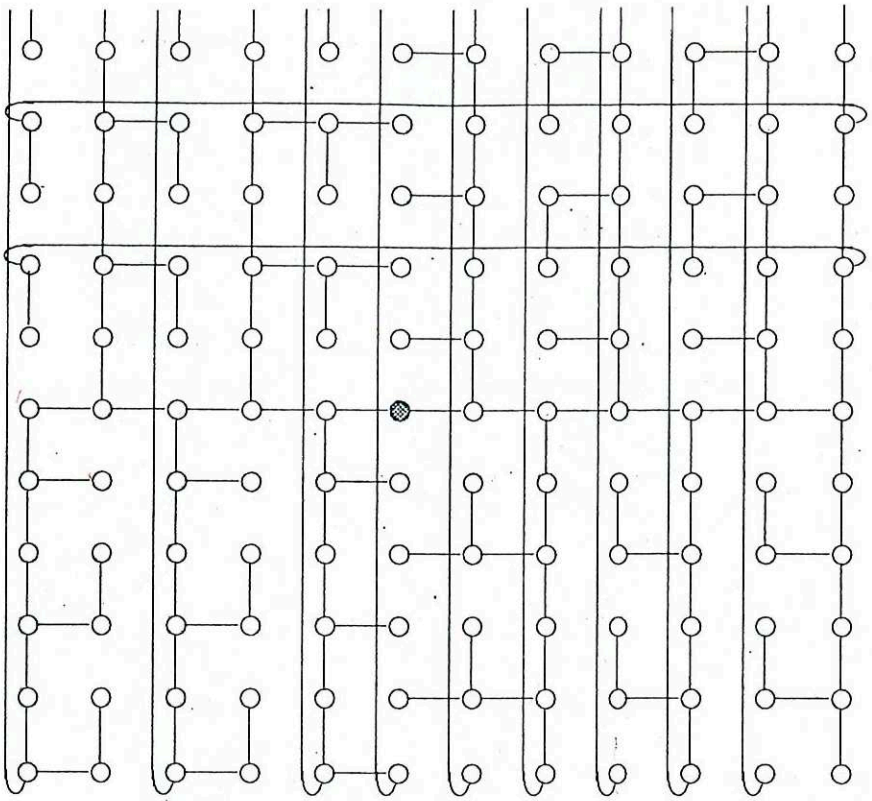

Fig. 11. Construction of the first tree for $n = 12$.

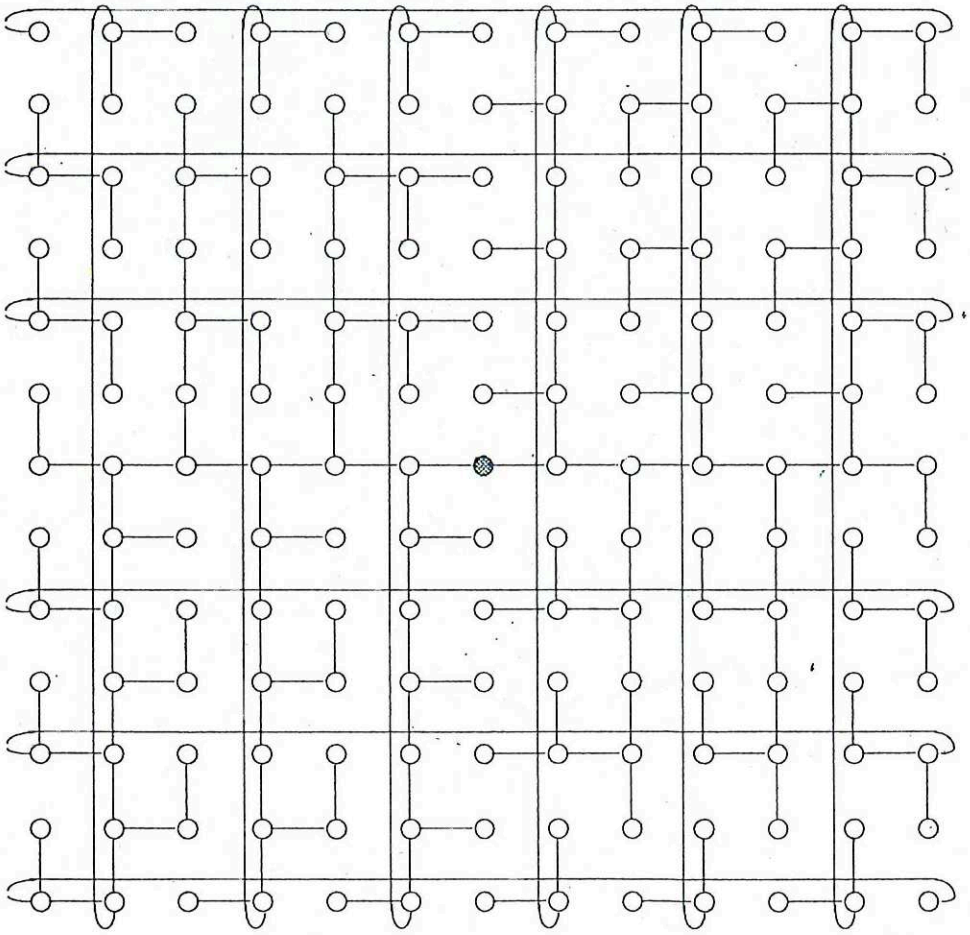

Fig. 12. Construction of the first tree for $n = 13$.

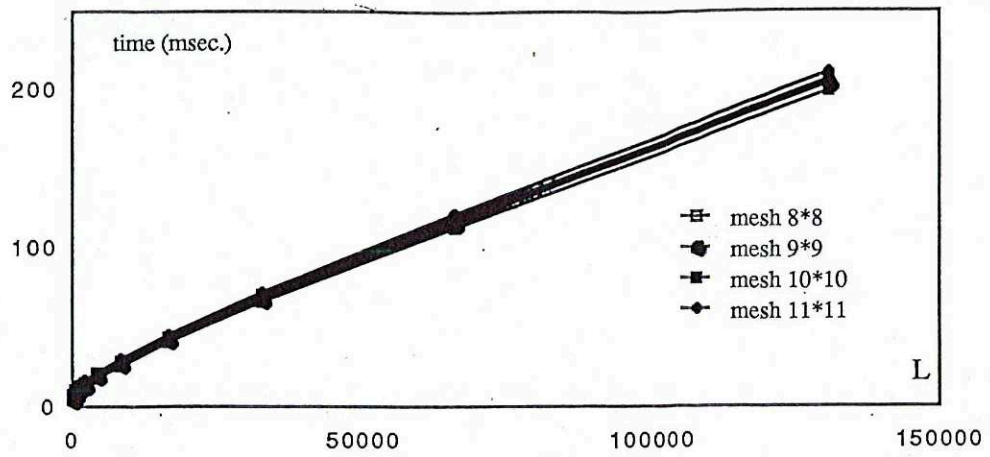

Fig. 13. Experimental results.

These experiments emphasize that there is a very good concordance between practical and theoretical results.

We can note that for large messages, on the theoretical point of view, the symmetric version of the Michallon–Trystram–Villard's algorithm is better than the proposed solution. However, from the practical point of view, it is not possible to implement this version because of synchronization reasons.

## 5. Conclusion

We have presented in this paper a general framework for the design and analysis of broadcasting algorithms on wraparound meshes of processors. All algorithm are based on pipelined sendings on edge-disjoint spanning trees. This analysis has first shown the link between the previous works of Saad and Schultz and Simmen and has allowed us to improve their results. The main result is the design of a new family of edge-disjoint spanning trees of optimal depth (equal to $n$ for an $n \times n$ square wraparound mesh).

## References

[1] JC. Bermond and P. Fraigniaud, Broadcasting and gossiping in de Bruijn networks, submitted to *SIAM J. Comp.*

[2] P. Fraigniaud and E. Lazard, Methods and problems of communication in usual networks, submitted to the special issue of *Discrete Applied Math.* on Broadcasting.

[3] H. Ho and L. Johnsson, Optimal broadcast on hypercubes, *IEEE TC* 38 (9) (1989).

[4] P. Michallon, D. Trystram and G. Villard, Optimal broadcasts on wraparound meshes, Technical Report #872I, LMC-IMAG, 1991.

[5] Y. Saad and M. Schultz, Data communication in parallel architectures, *Parallel Comput.* 11 (2) (1989) 131–150.

[6] M. Simmen, Comments on broadcast algorithms for 2-dimensional grids, *Parallel Comput.* 17 (1991) 109–112.

[7] Q. Stout and B. Wagar, Intensive hypercube communication; Prearranged communication in link-bound machines, *J. Parallel Distributed Comput.* 10 (2) (1990).

[8] La lettre du Transputer N.7, special issue on the TNode machine, Ed. Laboratoire d'Informatique de Besançon, 1990.

[9] E. Vargarigos and D. Bertsekas, Communication algorithms for isotropic tasks in hypercubes and wraparound meshes, Research Report CICS-P-231, 1990.