Optimal broadcasting in all-port meshes of trees with distance-insensitive routing[*]

Petr Salinger and Pavel Tvrdík

Department of Computer Science and Engineering
Czech Technical University, Karlovo nám. 13
121 35 Prague, Czech Republic
{salinger,tvrdik}@sun.felk.cvut.cz

Abstract

The mesh-of-trees topology has very attractive properties if a distance-sensitive routing, such as store-and-forward, is used. How its properties compare to meshes if distance-insensitive routing, such as wormhole, is used, was not previously understood. In this paper, we show that meshes of trees allow an elegant and optimal one-to-all broadcast algorithm supposing that routers implement distance-insensitive routing and have all-port capability.

1 Introduction

1.1 Collective communication operations

Given a connected network G and a designated node s possessing a packet, the task of a *one-to-all broadcast* (OAB) in G with the *source* s is to deliver the packet from s to every other node in G, either directly by s or by employing previously informed nodes. An *all-to-all broadcast* (AAB), also called *gossip*, is a collective communication operation in which every node is the source of one OAB. At the end of an AAB in G, *every* node knows the accumulated information of G.

The complexity and total latency of these communication operations depend on the network topology and switching and buffering capabilities of routers. In this paper, we consider only the *complete trees* and *meshes of trees*.

1.2 Complete binary trees and meshes of trees

Let $\mathcal{B} = \{0, 1\}$ be the *binary* alphabet and let $\mathcal{B}^n = \{x_{n-1} \ldots x_0 \, ; x_i \in \mathcal{B}\}$, $n \geq 1$, be the set of all n-bit strings.

For integer $i \geq 1$, the *i-fold concatenation* of bit α is denoted by α^i. The *empty* string is $\alpha^0 = \varepsilon$. Let $\mathcal{B}^0 = \{\varepsilon\}$. If $x \in \mathcal{B}^n$, $n \geq 1$ and $0 \leq i \leq n - 1$, then $x \, \text{XOR} \, 2^i$ is denoted by $\text{neg}_i(x)$. If $x \in \mathcal{B}^i$, then $\text{len}(x) = i$ denotes its *length*.

An undirected graph G consists of *nodes* $N(G)$ and *edges* $E(G)$. An *edge joining* two *adjacent* nodes u and v is denoted by $\langle u, v \rangle$.

Given integer $n \geq 1$, *the complete binary tree* of *height* n, CBT_n, is defined as follows:
$N(CBT_n) = \cup_{i=0}^n \mathcal{B}^i$
$E(CBT_n) = \{\langle x, xa \rangle \, ; \text{len}(x) < n \quad \text{and} \quad a \in \mathcal{B}\}$.

Nodes at *level i* are labeled with $(n - i)$-bit strings and are ordered lexicographically from left to right. The *root* of CBT_n, labeled with ε, is at *level n*. The *leaves* of CBT_n are nodes at level 0 (see Figure 1). For any node $x \in N(CBT_n)$, $x \neq \varepsilon$, we define $parent(x) = x'$, where $x = x'a, a \in \mathcal{B}$. An edge joining a parent with its left (right) child at level i is a left (right, respectively) *edge at level i*.
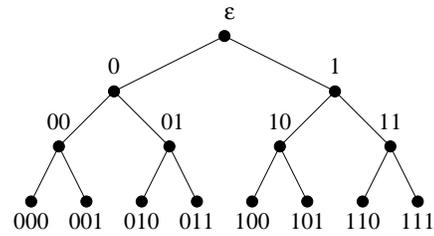

Figure 1. Node labeling in complete binary tree CBT_3.

Given integer $n \geq 1$, the 2-D *mesh of trees* of height n, MT_n, is defined as follows:
$N(MT_n) = \left(\mathcal{B}^n \times \cup_{i=0}^n \mathcal{B}^i \right) \cup \left(\cup_{i=0}^n \mathcal{B}^i \times \mathcal{B}^n \right)$
$E(MT_n) = \{\langle (x, y), (xa, y) \rangle \, ; \text{len}(x) < n \quad \text{and} \quad a \in \mathcal{B}\}$
$\quad \cup \, \{\langle (x, y), (x, ya) \rangle \, ; \text{len}(y) < n \quad \text{and} \quad a \in \mathcal{B}\}$.
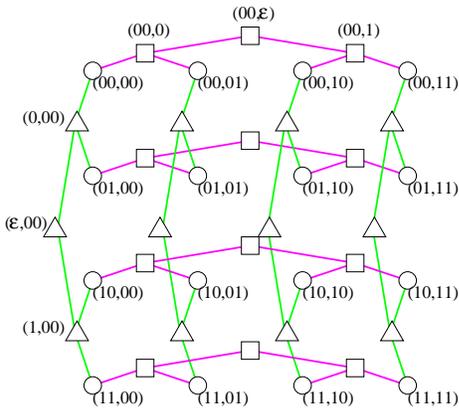
Figure 2. Node labeling in mesh of trees MT_2.

Hence, $|N(MT_n)| = 3 \cdot 2^{2n} - 2^{n+1}$ and $|E(MT_n)| = 2^{n+2}(2^n - 1)$. MT_n is a subgraph of the cartesian product $CBT_n \times CBT_n$. It can be viewed as the 2-D mesh $2^n \times 2^n$ whose nodes are leaves of *row* and *column* CBT_n's. Therefore, each mesh node is a leaf of exactly one row tree and one column tree. Figure 2 shows MT_2. The mesh (=level-0) nodes are depicted as circles ($\circ$), column tree roots and internal nodes as triangles ($\triangle$), and row tree roots and internal nodes as squares ($\square$). The mesh of trees is a hierarchically recursive topology: MT_n decomposes into grid 2×2 of MT_{n-1}'s by deleting roots of all the row and column trees. This is called the *canonical decomposition* of MT_n. By iteration, MT_n can be canonically decomposed into 2^{2i} MT_{n-i}'s arranged in square grid $2^i \times 2^i$. Each of these sub-meshes of trees is uniquely specified by pair (x, y), where $x, y \in \mathcal{B}^i$, and is denoted by sub$MT_{n-i}(x, y)$.

The mesh of trees is not regular. The mesh and root nodes have degree 2 and the internal tree nodes have degree 3. The mesh of trees combines good properties of trees and meshes, while eliminating their weaknesses [3]. It has the diameter of the tree topology (i.e., logarithmic) and the bisection width of the 2-D mesh topology (i.e., the square root of the size).

The *shortest path routing* is a combination of the tree and mesh routing. The shortest path from a row-tree node to a column-tree node is defined uniquely and it passes through the single mesh node shared by both trees. If source node u is in one row tree at level $l(u)$ and destination node v in another row tree at level $l(v)$ and if $l(u) \geq l(v)$, then there exist $2^{l(v)}$ different shortest paths from u to v. A column-tree case is similar [3].

MT_1 is isomorphic to a ring with 8 nodes. Therefore, we consider only MT_n with $n \geq 2$ here. The mesh-of-tree topology can be generalized in at least three different ways: by using k-ary trees, rectangular meshes, and/or higher-dimensional meshes. Here, we consider only the standard binary 2-D square topology.

1.3 Communication models

We assume that each node has a router capable of *distance-insensitive* switching, such as wormhole or circuit switching. We also assume *half-duplex* links and *all-port* capability, i.e., each router can use all its input and output ports simultaneously. A OAB algorithm consists of *rounds*, corresponding to levels of the broadcast tree. The packets are moved along edges of the broadcast tree, mapped on paths in the mesh of tree, level by level. In each round, all paths are pairwise edge-disjoint. The complexity of a OAB is defined to be the number of its rounds.

2 Previous and related work

Previous results apply either to 1-port meshes of tree or to all-port meshes and tori.

2.1 1-port meshes of trees

Algorithms for OAB and AAB in MT_n in 1-port store-and-forward full-duplex combining communication model have been presented in [1]. The lower bound on the number of rounds of both OAB and AAB is the diameter of MT_n, i.e., $4n$. The algorithm in [1] for OAB needs $5n + \lfloor \frac{n}{2} \rfloor + 1$ rounds and for AAB $7n + 2$ rounds. In both cases, there is a gap between the lower and upper bounds.

Optimal algorithms for OAB and AAB in 1-port wormhole MT_n with packet combining capability are also known. If routers allow link-disjoint routing, an optimal OAB algorithm from [2] can be used. It is shown in [6] that even under the constraints of node-disjoint routing, there is an optimal algorithm in MT_n for OAB. It needs $2n + 2$ rounds, which is exactly the lower bound. Paper [6] also describes asymptotically optimal algorithms for AAB, in both half-duplex and full-duplex models. Their number of rounds differs from the known lower bounds by additive lower-order terms.

2.2 All-port wormhole meshes and tori

Many OAB algorithms have been developed for all-port wormhole 2-D meshes and tori. The lower bound in the i-port model is $\lceil \log_{i+1}(|N(G)|) \rceil$ rounds, i.e., $\lceil \log_5(|N(G)|) \rceil$ for a 2-D all-port torus G.

An optimal algorithm for OAB in 2-D torus $5^k \times 5^k$, $k \geq 1$, is given in [5]. It is based on recursive tiling of the torus. The same problem for n-dimensional square torus with $(2n + 1)^k$ nodes in each side is studied in [4]. The main advantage over the algorithm in [5] is that it can be applied to higher-dimensional cases. The number of rounds

is nk, which is optimal. This algorithm can be simulated on a n-dimensional square mesh with $(2n+1)^k$ nodes in each side in $nk + n + k - 1$ rounds. The routing used in these algorithms is not dimension-ordered.

An approach conforming with dimension-ordered routing is described in [7]. It is based on an extension of the graph-theoretical concept of dominating sets. These OAB algorithms take $2k$ and $2k + 1$ rounds in square tori $4^k \times 4^k$ and $2.4^k \times 2.4^k$, respectively, which is about $\log_4 5$ times more than the lower bounds. They work only for 2-D tori of special size and are hard to generalize to higher dimensional tori. The best known all-port OAB algorithm has been proposed in [8]. It is based on so called "dilated diagonals". The number of rounds required by this scheme is at most 2 rounds (5 rounds) more than the optimal number of rounds when the torus is square (rectangular, resp.). The routing used in the scheme is always dimension ordered.

3 Lower bound

The minimal number of rounds of an all-port OAB in MT_n depends on the degree of the source node. As noted above, MT_n contains nodes with degree 2 and 3. Let $b_2(n)$ ($b_3(n)$) denote the lower bound on the number of rounds of an all-port OAB in MT_n if the source node has degree 2 (3, respectively).

Lemma 1 $b_3(n) = n + 1$ and $b_2(n) = n + 2$ for all $n \geq 2$.

Proof:
The case of $b_3(n)$ is trivial, since $b_3(n) = \lceil \log_4 |N(MT_n)| \rceil = \lceil \log_4(3.2^{2n} - 2^{n+1}) \rceil = n + 1$.

Consider an OAB from a node with degree 2. Let x_i denote the number of informed nodes after round i. Then

$$x_0 = 1,$$

$$x_i \leq x_{i-1} + 2x_0 + 3(x_{i-1} - x_0) = 4x_{i-1} - 1, \qquad i \geq 1.$$

The solution of this recurrent equation is

$$x_i \leq 4^i - \frac{4^i - 1}{3}.$$

We have to find the smallest i such that $x_i \geq |N(MT_n)|$, i.e.,

$$4^i - \frac{4^i - 1}{3} \geq 3.2^{2n} - 2^{n+1},$$

which is equivalent to

$$2.4^i \geq 9.4^n - 6.2^n - 1. \tag{1}$$

If $n \geq 3$, then (1) holds if and only if $i \geq n + 2$.

It remains to show that $b_2(2) = 4$, even though for $n = 2$ (1) holds for $i \geq n + 1 = 3$. The proof is by contradiction.

Assume that $b_2(2) = 3$. MT_2 has 40 nodes. Clearly, $x_1 = 3$ and $x_2 \leq 4.3 - 1 = 11$. A 3-round OAB in MT_2 cannot have $x_2 = 10$, since then $x_3 \leq 4.10 - 1 = 39$. Assume $x_1 = 3$ and $x_2 = 11$. Let $x_{i,2}$ ($x_{i,3}$) denote the number of degree-2 (degree-3, respectively) nodes informed after round i. Then $x_{2,2} \geq 1$, since the source itself is a degree-2 node. Also $x_{2,2} \leq 4$, since for $x_{2,2} \geq 5$, we get $x_3 \leq x_2 + 2.x_{2,2} + 3(x_2 - x_{2,2}) = 4x_2 - x_{2,2} \leq 4.11 - 5 = 39$.
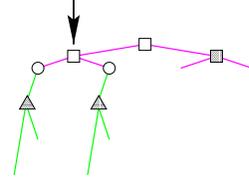
Figure 3. The neighborhood of an uninformed level-1 node in MT_2.

MT_2 has 24 degree-2 nodes and 16 degree-3 nodes. It forms a bipartite graph, degree-2 nodes have neighbors of degree 3 and vice versa. Hence, any path in MT_2 alternates degree-2 and degree-3 nodes. Each uninformed degree-2 node has 2 degree-3 neighbours. Therefore, the number of uninformed degree-3 nodes such that each of them has all its degree-2 neighbours uninformed (Figure 3 shows the situation in case of a row tree) is at least $16 - 2.x_{2,2} - x_{2,3} = 16 - x_2 - x_{2,2} = 5 - x_{2,2} > 0$ uninformed degree-3 node(s) facing the situation on Figure 3. Even if all the 3 degree-3 nodes (shaded on the figure) are informed, all the 4 uninformed nodes cannot be informed in one single round, since each link can carry at most one packet in one direction in one round. □

4 The OAB algorithm

Our approach to designing an optimal OAB in the all-port MT_n is the following: For any $n \geq 3$, we give an optimal, i.e., $(n+1)$-round algorithm for OAB if the source node is an internal tree node at level 1. This algorithm is called Algorithm A. If the source node is not at level 1, then, in the first round, the source sends its packet to any level-1 tree node and then Algorithm A is applied. Hence, our algorithm achieves the lower bound if the source of OAB is a mesh node or a root or a level-1 tree node and it needs 1 round more if the source is an internal tree node at level i, $2 \leq i \leq n - 1$.

The algorithm for MT_2 is a special case, it can be obtained by modification of the second round of Phase 2 and Phase 3 of Algorithm A (see Section 4.1).

Since MT_n is node symmetric with respect to the tree nodes at level 1, we describe without loss of generality Algorithm A with the source node $(0^n, 0^{n-1})$.

For the sake of brevity, we use the following auxiliary functions on binary strings.

Definition 1 *The auxiliary functions $\lambda_X, \lambda_Y : \mathcal{B}^3 \to \mathcal{B}^2$ are defined by $\lambda_X(x_2 x_1 x_0) = y_1 \overline{y_0}$ and $\lambda_Y(x_2 x_1 x_0) = y_1 y_0$, where $y_1 = x_2 \oplus x_1 \oplus x_0$ and $y_0 = x_1$. For easier reading, these functions are evaluated in the following table.*

x	000	001	010	011	100	101	110	111
$\lambda_X(x)$	01	11	10	00	11	01	00	10
$\lambda_Y(x)$	00	10	11	01	10	00	01	11

For any $a \in \cup_{i=0}^{\infty} \mathcal{B}^i$ and $b \in \mathcal{B}$, define

$$del\_trailing(a, b) = \begin{cases} del\_trailing(a', b) & \text{if } a = a'b, \\ a & \text{otherwise;} \end{cases}$$

$$\kappa(ab) = del\_trailing(a, b).$$

First, we give the pseudocode of Algorithm A and then we prove its correctness and complexity. In Figures 4–8, the following convention is used: the already informed nodes are *black*, the nodes being informed in the given round are *grey*, and uninformed nodes are *white*. Algorithm A uses the standard shortest-path routing.

Algorithm A from source node $(0^n, 0^{n-1})$**:**

/* Phase 1: $n - 3$ rounds, see Figure 4 */
for $i = 1$ **to** $n - 3$ **do**
   for all $x, y \in \mathcal{B}^{i-1}$ **do_in_parallel**
     $(x00^{n-i}, y00^{n-1-i})$ sends the packet
        to $(x00^{n-i}, y10^{n-1-i})$ and
        to $(x10^{n-i}, y00^{n-1-i})$ via $(x, y00^{n-1-i}0)$ and
        to $(x10^{n-i}, y10^{n-1-i})$ via $(x, y00^{n-1-i}1)$;

/* Phase 2: 2 rounds */
/* the first round, see Figure 5 */
for all $x, y \in \mathcal{B}^{n-3}$ **do_in_parallel**
   $(x000, y\lambda_Y(000))$ sends the packet
      to $(x\lambda_X(001), y001)$ and
      to $(x\lambda_X(101), y101)$ and
      to $(x100, y\lambda_Y(100))$ via $(x, y000)$;

/* the second round, see Figure 6 */
for all $x, y \in \mathcal{B}^{n-3}$ and $z \in \mathcal{B}$ **do_in_parallel**
   parbegin
     $(xz00, y\lambda_Y(z00))$ sends the packet /* row-tree source */
        to $(x\lambda_X(z11), yz11)$ and
        to $(x\lambda_X(z00), yz00)$ and
        to $(xz11, y\lambda_Y(z11))$ via $(xz, y\lambda_Y(z00)1)$;

     $(x\lambda_X(z01), yz01)$ sends the packet /* column-tree source */
        to $(xz01, y\lambda_Y(z01))$ and
        to $(xz10, y\lambda_Y(z10))$ and

        to $(x\lambda_X(z10), yz10)$ via $(x\lambda_X(z01)1, yz)$;
   parend

/* Phase 3: 2 rounds */
/* the first round */
for all $x, y \in \mathcal{B}^{n-3}$ and $z \in \mathcal{B}^3$ **do_in_parallel**
   parbegin
   /* see Figure 7(a) */
     $(xz, y\lambda_Y(z))$ sends the packet /* row-tree source */
        to (xz, y) and /* grandparent */
        to $(neg_0(xz), y\lambda_Y(z)0)$ and
        to $(neg_0(xz), y\lambda_Y(z)1)$;

     $(x\lambda_X(z), yz)$ sends the packet /* column-tree source */
        to (x, yz) and /* grandparent */
        to $(x\lambda_X(z)0, neg_0(yz))$ and
        to $(x\lambda_X(z)1, neg_0(yz))$;
   parend

/* the second round */
for all $x, y \in \mathcal{B}^{n-3}$ and $z \in \mathcal{B}^3$ **do_in_parallel**
   parbegin
     /* row-tree pattern, see Figure 7(b) */
     $(xz, y\lambda_Y(z))$ sends the packet /* level-1 source */
        to $(xz, y\lambda_Y(z)0)$ and
        to $(xz, y\lambda_Y(z)1)$;

     $(neg_0(xz), y\lambda_Y(z)0)$ sends the packet /* level-0 source */
        to $(parent(neg_0(xz)), y\lambda_Y(z)0)$ and
        to $(neg_0(xz), y\lambda_Y(z))$;

     $(neg_0(xz), y\lambda_Y(z)1)$ sends the packet /* level-0 source */
        to $(parent(neg_0(xz)), y\lambda_Y(z)1)$;

     /* column-tree pattern */
     $(x\lambda_X(z), yz)$ sends the packet /* level-1 source */
        to $(x\lambda_X(z)0, yz)$ and
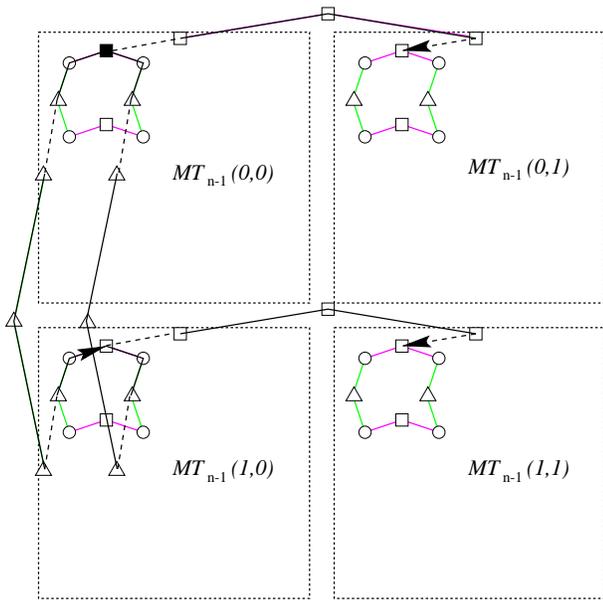   to $(x\lambda_X(z)1, yz)$;

     $(x\lambda_X(z)0, neg_0(yz))$ sends the packet /* level-0 source */
        to $(x\lambda_X(z)0, parent(neg_0(yz)))$ and
        to $(x\lambda_X(z), neg_0(yz))$;

     $(x\lambda_X(z)1, neg_0(yz))$ sends the packet /* level-0 source */
        to $(x\lambda_X(z)1, parent(neg_0(yz)))$;

     /* see Figure 8 */
     (xz, y) sends the packet /* level-3 row-tree source */
        to $(xz, y0)$ and
        to $(xz, y1)$ and
        to $(xz, \kappa(y))$; /* only for $y \neq 0^{n-3}$ */

     (x, yz) sends the packet /* level-3 column-tree source */
        to $(x0, yz)$ and
        to $(x1, yz)$ and
        to $(\kappa(x), yz)$; /* only for $x \neq 0^{n-3}$ */

   parend

4

Figure 4. The canonical decomposition of MT_n and the first round of Phase 1.

Theorem 1 *Algorithm A performs the OAB from source node $(0^n, 0^{n-1})$ in the all-port wormhole MT_n in $n + 1$ rounds using the standard shortest-path routing.*

Sketch of the proof:

Phase 1

In the first round of Phase 1, MT_n is canonically decomposed into 4 sub$MT_{n-1}(x, y)$, where $x, y \in \mathcal{B}$, and the source node $(0^n, 0^{n-1})$ in sub$MT_{n-1}(0, 0)$ sends the packet to its homothetic images in the remaining three sub$MT_{n-1}(x, y)$, for $(x, y) \in \{(0, 1), (1, 0), (1, 1)\}$. See Figure 4. Recursively, the same is done within each sub$MT_{n-1}(x, y)$. After $n - 3$ rounds, every sub$MT_3(x, y)$, $x, y \in \mathcal{B}^{n-3}$, contains exactly one informed node $(x0^3, y0^2)$.

Phase 2

The task of Phase 2 is to disseminate the packet possessed by node $(x0^3, y0^2)$ within each sub$MT_3(x, y)$, $x, y \in \mathcal{B}^{n-3}$, so that the following conditions hold:

(i) In each sub$MT_1(x', y')$, $x', y' \in \mathcal{B}^{n-1}$, exactly one level-1 node is informed.

(ii) In each row and in each column of level-1 nodes of any sub$MT_3(x, y)$, $x, y \in \mathcal{B}^{n-3}$, exactly one node is informed.
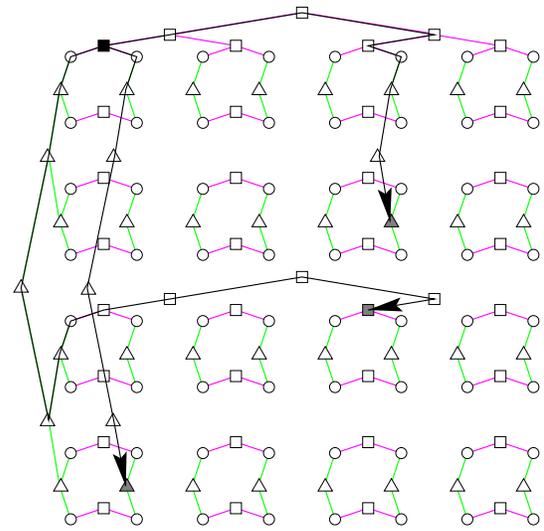
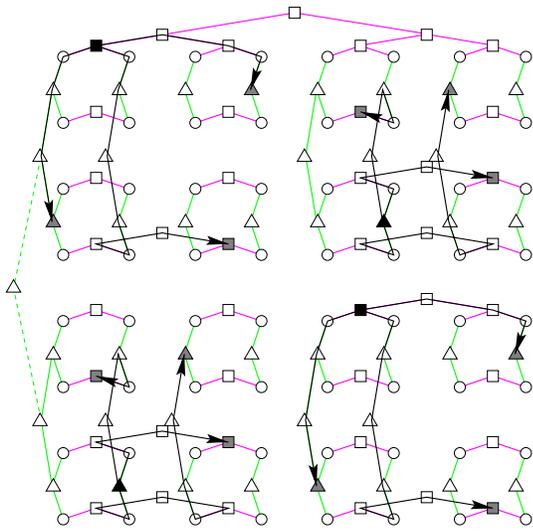**Figure 5. Phase 2, the first round – the packet dissemination in one sub$MT_3(x, y)$.**

This is achieved by using functions λ_X and λ_Y for computing the destinations in Phase 2. More precisely, the set of level-1 informed nodes after Phase 2 is $(xz, y\lambda_Y(z))$ and $(x\lambda_X(z), yz)$ for all $x, y \in \mathcal{B}^{n-3}$ and $z \in \mathcal{B}^3$.

The 2 rounds of Phase 2 within one sub$MT_3(x, y)$ are illustrated in Figures 5, 6, where all the level-1 and level-0 nodes are depicted and most of the level-2 and level-3 nodes are omitted.
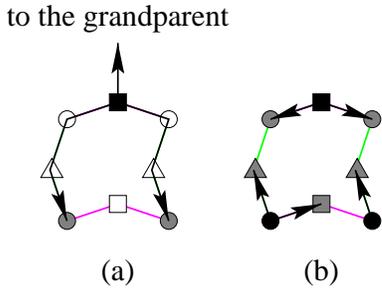
Phase 3

In the first round of Phase 3, each informed level-1 node in each sub$MT_1(x, y)$, $x, y \in \mathcal{B}^{n-1}$, sends the packet to its grand parent and to two level-0 nodes. The situation at one row-tree level-1 node is depicted in Figure 7(a), the other situations are symmetric. After this round, all level-3 nodes are informed.

The last round, i.e., the second round of Phase 3, is a bit more difficult to describe, since there are 3 kinds of informed nodes at levels 0, 1, and 3, respectively, but the communication pattern is very simple. Every level-1 source sends the packet to its two children. Level-0 sources inform one or two level-1 neighbors, Figure 7(b) shows one kind of the situation in the row-tree case, the others are symmetric. Finally, the level-3 nodes take care about the rest. Each of them informs its two children (all level-2 nodes are then informed) and one of upper level nodes. This upper level destination is computed by function κ, which guarantees that different level-3 nodes inform different predecessors in their trees. Nodes $(0^{n-3}, y)$ and $(x, 0^{n-3})$, $x, y \in \mathcal{B}^n$, send the packet to its two children only. Figure 8 illustrates the communication pattern in a row tree in MT_6. □

Figure 6. Phase 2, the second round – the packet dissemination in one sub $MT_3(x, y)$.

to the grandparent

(a) (b)

Figure 7. Phase 3 – the packet dissemination in one sub $MT_1(x, y)$ between nodes at level $0, 1$ and 3. (a) The first round. (b) The second round.

Theorem 2 *The number of rounds to perform a OAB in the all-port wormhole MT_n is $n + 2$ for all $n \geq 3$.*

Proof:

The lower bound has been proven in Lemma 1. If the source node is at level 1, Algorithm A is used. Otherwise, the source node sends the packet to a closest level-1 node and algorithm A is applied. □

4.1 OAB in MT_2

If the source is one of degree-3 nodes of MT_2, then a 3-round OAB is obtained by a trivial modification of the last 3 rounds of Algorithm A. This is optimal, since $b_3(2) = 3$. If the source is a degree-2 node, then $b_2(2) = 4$ by Lemma 1 and we can apply the same idea as in Theorem 2 to get an optimal broadcasting algorithm.
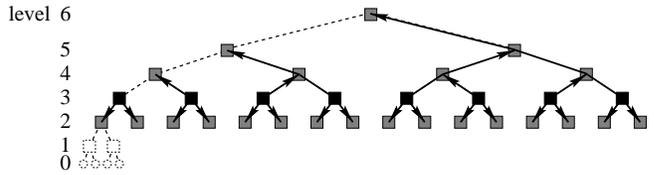
Figure 8. The second round of Phase 3 at row-tree nodes at levels $2 \ldots n$.

5 Conclusions and further work

We have proposed a OAB algorithm for all-port wormhole meshes of trees. The algorithm achieves the optimal number of rounds if the source of the OAB is a mesh node, level-1 tree node, or a root. This represents 67% of nodes. In the remaining cases, our algorithm needs one additional round. To remove this gap seems to be hard.

This basic result can be extended in several ways. We have generalized Algorithm A for 2-D binary rectangular meshes of trees, i.e., if the mesh has size $2^n \times 2^m$, $n \neq m$. We have also designed a similar optimal algorithm for 3-D binary cube meshes of trees. Since the degree of mesh nodes is 3 in this case, the main idea is to inform mesh nodes by splitting the mesh uniformly into $1/64$-th parts. The splitting pattern is recursive with the period of 3 rounds. However, it remains an open problem to generalize the main result of this paper to k-ary meshes of trees, $k \geq 3$.

References

[1] D. Barth. An algorithm of broadcasting in the mesh of trees. In *CONPAR 92–VAPP*, volume 634 of *LNCS*, pages 843–844. Springer-Verlag, 1992.

[2] A. M. Farley. Minimum-time broadcast networks. *Networks*, 10:59–70, 1980.

[3] F. T. Leighton. *Introduction to Parallel Algorithms and Architectures: Arrays • Trees • Hypercubes*. Morgan Kaufmann, 1992.

[4] J.-Y. L. Park and H.-A. Choi. Circuit-switched broadcasting in torus and mesh networks. *IEEE Transactions on Parallel and Distributed Systems*, 7(2):184–190, Feb. 1996.

[5] J. G. Peters and M. Syska. Circuit-switched broadcasting in torus networks. *IEEE Transactions on Parallel and Distributed Systems*, 7(3):246–255, Mar. 1996.

[6] P. Salinger and P. Tvrdík. Optimal broadcasting and gossiping in one-port wormhole meshes of trees. In *Proc. of PDCS'99*, volume 2, pages 713–718. Acta Press, 1999.

[7] Y.-J. Tsai and P. K. McKinley. A broadcast algorithm for all-port wormhole-routed torus networks. *IEEE Transactions on Parallel and Distributed Systems*, 7(8):876–885, Aug. 1996.

[8] Y.-C. Tseng. A dilated-diagonal-based scheme for broadcast in a wormhole-routed 2D torus. *IEEE Transactions on Computers*, 46(8):947–952, 1997.