Efficient line broadcasting in trees

A. Averbuch * I. Gaber † Y. Roditty ‡

Abstract

Efficient algorithms for line broadcasting in Complete Binary and Binomial trees, are presented. They have linear in n(the number of vertices), complexity. This answers a question raised by Fujita and Farley [5]. An insight on some other trees is given, as well.

1 Introduction

Let $G = G(V, E)$ be a connected graph. A communication network is defined as a connected graph. Given a set $A \subset V(G), A \neq \emptyset$, we define *broadcasting (multi-casting)* from A to be the process of transmitting one (many) unit(s) of information from A to all the vertices in $G \setminus A$ ($B \subset (G \setminus A)$). The vertices in A are called *originators*. This is accomplished by a series of transitions over G. The messages are distributed over G and spread through the edges, where each vertex transmits a message to its neighbors upon receiving it regardless of the activities in other vertices (besides the vertex that receives the message that has to be idle). This procedure is called *local*.

In *line* broadcasting an informed vertex may call any other vertex using the communication links of any simple path between the two vertices with the restriction that no link is used in more than one call at a given time unit. It is obvious that if local broadcasting is in action then the number of informed vertices can be at most double at each time unit, therefor, at least $\lceil log_2 n \rceil$ time units are needed for broadcasting in a network of n vertices. In general it is not possible to inform all vertices of a network in that time. However, Farley [3] showed that there is a minimum-time line broadcast scheme for any originator in any connected network.

The question we deal with here is:
How much line broadcasting is needed to complete a minimum-time broadcasting from any arbitrary originator in a given connected graph?

*Department of Computer Science, School of Mathematical Sciences, Tel Aviv University, Tel Aviv 69978, Israel.

†Department of Computer Science, School of Mathematical Sciences, Tel Aviv University, Tel Aviv 69978, Israel and Department of Computer Science, The Academic College of Tel-Aviv-Yaffo, Tel-Aviv 61161, Israel.

‡Department of Computer Science, School of Mathematical Sciences, Tel Aviv University, Tel Aviv 69978, Israel and Department of Computer Science, The Academic College of Tel-Aviv-Yaffo, Tel-Aviv 61161, Israel. email: jr@math.tau.ac.il

Farley [3] ,which introduced the topic, gave a constructive proof that minimum-time line broadcasting is possible in any tree ,and thus in any connected network. His construction gives an upper bound of $(n-1)\lceil log_2 n\rceil$ on the total length (total number of edges) needed for line broadcasting in a minimum time in any network of n vertices. Kane and Peters [6] determined the value of the minimum total length for minimum-time line broadcasting in any cycle on n vertices. When $n = 2^k$ they gave an exact value, while for other choices of n an upper bound was presented, which is about $\frac{1}{3}$ of Farley's upper bound. Not much work has been done along this line since then (for further information see [6]). Recently A. Averbuch, Y. Roditty and B. Shoham [1] obtained efficient line broadcasting algorithms in a $d-$ dimensional grid from certain originators (in some cases from any originator) which produce linear , in n, cost.

In this article we establish efficient line broadcasting algorithm for the complete binary tree and optimal algorithm for the binomial tree and give insight on some other trees.

Our algorithms demonstrate broadcasting scheme of linear ,in n, cost.

For other graph theory concepts consult Harary [7] or West [8].

2 Line-broadcasting in particular trees

2.1 Definitions and preliminary results

Denote by $F(j, v)$ the total cost of line broadcasting from an originator v, after j time units.

A vertex $v \in V(G)$ is called *optimal originator* if the cost of broadcasting (line-broadcasting) from it is minimal with respect to other vertices in G. (It is possible that v is not unique).

To the sequel we deal with trees only.

Let T be a tree and $v \in V(T)$. By T_v we denote the subtree of T rooted at v.

If a vertex w receives the message from one of the vertices, say w_1, we call the path between w_1 and w the *transmission path* between w_1 and w.

Let T be a tree with $|V(T)| = n$. Denote by $W(T, j)$ the number of vertices in T which know the message after j time units, and $\overline{W(T,j)} = n - w(T, j)$.

Let $T(k)(T^*(k))$ be a tree with $f(k)$ $(g(k))$ vertices such that there is an edge (a vertex v) in it say,(x, y), so that $T(k) \setminus \{(x,y)\}$ $(T^*(k) \setminus \{v\})$, results in a forest consisting of two trees $T(k-1)(T^*(k-1))$. Denote by T_x, T_y these two isomorphic parts of $T(k)$ $(T^*(k))$. The subtree $T(k-1)(T^*(k))$ has the same property up to a tree which is a single edge (vertex).

Observe that the definition yields $f(k) = 2^k$ and $g(k) = 2^k - 1$.

We start with two useful lemmas.

Lemma 2.1 *Let $T(k)$ $(T^*(k))$ be the tree defined above. Then in any efficient line broadcasting scheme,*

$$|W(T_x, t) - W(T_y, t)| \leq 1 \ , \tag{1}$$

and $t = 1, 2, \cdots, \lceil log_2 n\rceil$.

Proof: Assume the contrary. That is at some time unit, say $t = s$,

$$|W(T_x, s) - W(T_y, s)| \geq 2, \ .$$

Put $W(T_y, s) = a$ so that, $W(T_x, s) \geq a + 2$. At the best case a vertex from T_x transmits at $t = s + 1$ to a vertex at T_y. (Observe that during this transmission the edges $(v, x), (v, y)$ or the edge (x, y) are occupied and no other transmission can pass). It follows that,

$$W(T_x, s) \geq 2(a + 1) + 1 = 2a + 3 \ , \quad W(T_y, s) = 2a + 1.$$

Since each vertex transmits at each time unit in a minimum time broadcasting scheme , until the last time unit and since $n = 2^r$or $n = 2^r - 1$, the "gap" still remains and broadcasting will not end at $t = \lceil log_2 n \rceil$. Otherwise, if either T_y does not receive a message from outer vertices or T_x does not send a message outside, the "gap" gets larger. ∎

Lemma 2.2 *Let $T(k)$ ($T^*(k)$) be the tree defined above. Assume w is an optimal originator located ,say in T_x. Then,there exists an efficient line-broadcasting algorithm such that:*
(i) v transmits the message at $t = 1$ to the optimal originator in T_y.
(ii) For $t = 2$ to $\lceil log_2 n \rceil$ we continue with the same broadcasting scheme recursively.

Proof: The proof of (i) follows from lemma 2.1. Indeed, by lemma 2.1 w must transmit the message at $t = 1$ to a vertex in T_y. So a "good" vertex is obviously the optimal one.

Now from $t = 2$ both T_x and T_y should continue broadcasting according to the same procedure autonomically (so that (1) is not violated). ∎

2.2 Line broadcasting in a Complete Binary Tree

The Complete Binary Tree is viewed as a rooted tree in u_1 with $N[v] = \{x, u_2\}$ and levels $L_1, L_2, \cdots, L_r$, where in level L_j we have 2^{j-1} vertices, and $|V(T)| = n = 2^r - 1$ so that $r = log_2(n + 1)$. We shall denote it $r-$CBT.

By RP_r we denote the right most path in T of length $r - 1$, that is: $(u_1, u_2, u_3, \cdots, u_r)$ where, u_j is at level L_j. Let y be any vertex in T. We denote by T_y the complete binary subtree of T, rooted in y and by T_y^* we denote the complete subtree rooted at y with his father. By $rs(w), ls(w)$ we denote the right and left sons of a vertex w, respectively. $f^i(w)$ is the $i's$ ancestor of a vertex w, such that $f(w)$ is its father , $f^2(w)$, is its grandfather , and so on.

By the symmetry of the $r-$CBT we may assume that any originator of it belongs to RP_r.

By $F(k, w)$ we denote the cost to line broadcasting from w, after k time units.

Our main result in this section is:

Theorem 2.1 *Let T be a $r-CBT$. Then, for any $u_j \in RP_r$,*

$$\frac{6}{4}n - log_2(n + 1) \leq F(r, u_j) \leq \frac{7}{4}n - j \times log_2(n + 1) + j^2 - \frac{17}{4} \ . \qquad (1)$$

The proof of Theorem 2.1 is divided into two major parts: The proof of the lower bound (theorem 2.2) and the proof of the upper bound. The later case is accomplished as follows: first we prove it for u_{r-3} (theorem 2.3), later for u_1 (theorem 2.4) and finally for u_j , $j \neq 1, r - 3$ (theorem 2.5).

Theorem 2.2 *Let $u_j \in RP_r$. Then,*

$$\frac{6}{4}n - log_2(n+1) \leq F(r, u_j) \ .$$

Proof: Observe first that at $t = r - 1$, $\lceil \frac{n}{2} \rceil$ of the vertices know the message and have to transmit it to the rest $\lfloor \frac{n}{2} \rfloor$ vertices, at time $t = r$. So that at least $\frac{n}{2}$ edges has been used already. Now we shall prove that the cost of line broadcasting in any efficient line broadcasting algorithm from from $t = r - 1$ to $t = r$ is a least $2^r - r - 1 = n - log_2(n+1)$.

Let w be any vertex. Suppose the edge $(w, f(w))$ is not used at $t = r$, Then it follows that $|T \setminus T_w| = n_1 \equiv 0(mod\ 2)$,

$$W(T \setminus T_w, r-1) = \overline{W(T \setminus T_w, r-1)} \ , \tag{2}$$

and, $|T_w| = n_2 \equiv 1(mod\ 2)$,

$$W(T_w, r-1) \geq \overline{W(T_w, r-1)} \ . \tag{3}$$

Now since n_1 is even all edges in $T \setminus T_w$, must participate at the last stage. Indeed, if there is any edge , say e, which is not used at $t = r$, then $(T \setminus T_w) \setminus \{e\}$ results in two odd sub-trees of $T \setminus T_w$, which is impossible because of (2).

On the other hand, since T_w satisfies (3) we can follow the arguments for $(w, f(w))$, and obtain that at most r edges might not been used during the last time unit.

Since there is a bijection between the vertices that know the message in $t = r - 1$ and those that have to be informed , we use at least $2^r - 1$ edges except at most the r mentioned edges.

So the theorem is proved. ∎

Before establishing the proof of the upper bound in (1) we need the following simple but essential lemma.

Lemma 2.3 *Let T be a $4-CBT$. Then,*

$$F(4, u_1) = 19.$$

Proof: To prove the assertion of the lemma we present the following algorithm:

Algorithm Binary :

1. At $t = 1$, u_1 transmits the message to u_2.
2. At $t = 2$, u_1 transmits the message to x and u_2 transmits to u_3.
3. At $t = 3$, x transmits the message to $rs(x)$, u_2 transmits to $ls(u_2)$, u_3 transmits to u_4 and u_1 transmits the message to $ls(x)$

4. At $t = 4$, u_1 transmits to $ls(ls(x))$, $ls(x)$ transmits to $rs(ls(x))$ x transmits to $ls(rs(x))$, $rs(x)$ transmits to $rs(rs(x))$ u_3 transmits to $ls(u_3)$, u_2 transmits to $ls(ls(u_2))$ and $ls(u_2)$ transmits to $rs(ls(u_2))$.

Easy computation show the value obtained in the lemma.

In Fig. 1 we demonstrate Algorithm Binary. (The doted vertices are those who know the message after each time unit). ■

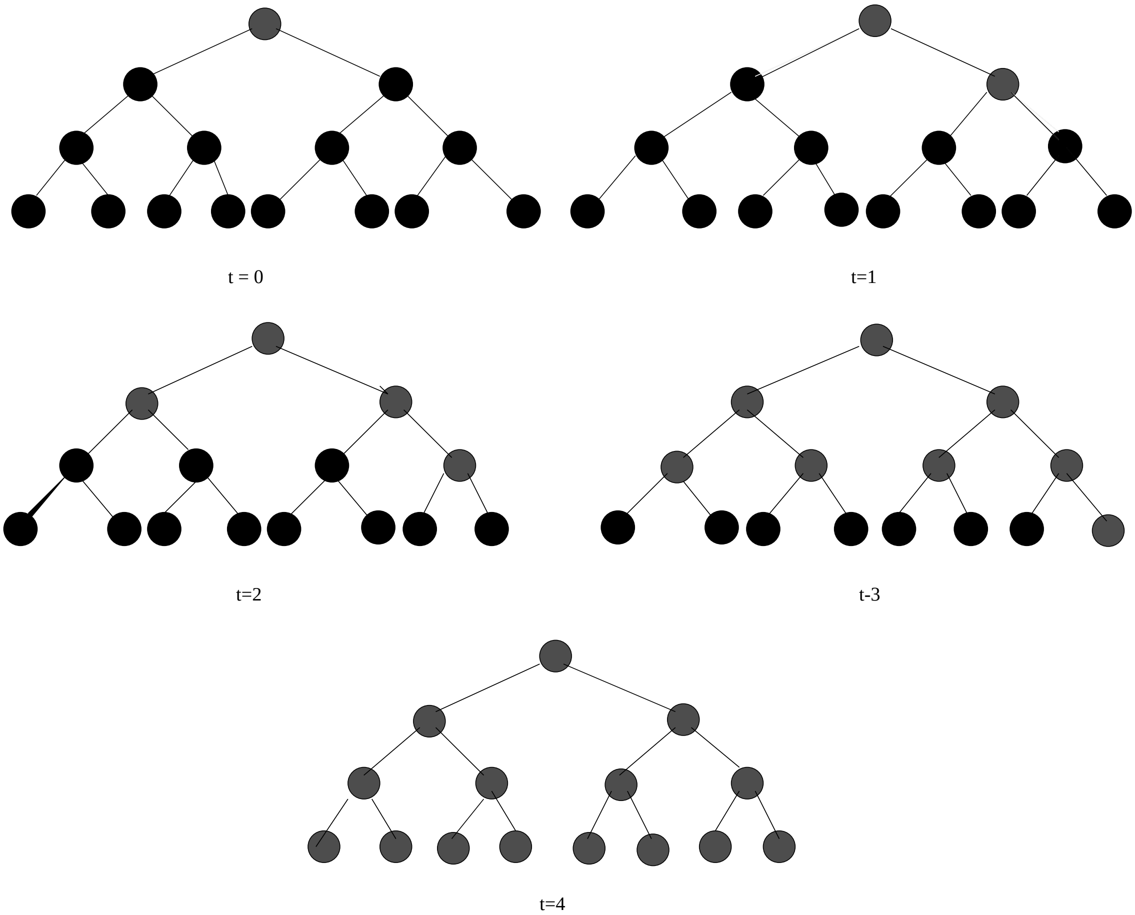

Figure 1: Algorithm Binary for a tree with $n = 2^4 - 1$.

Theorem 2.3 *Let T be a $r-CBT$. Then,*

$$F(r, u_{r-3}) \leq \frac{7}{4}2^r - 3r + 3 = \frac{7}{4}n - 3log_2(n+1) + \frac{19}{4} \; . \tag{4}$$

Proof: To prove the theorem we present Algorithm Binary 1:

1. At $t = 1$,u_{r-3} will transmit the message to the corresponding u_{r-3} originator of the subtree $T_{ls(u_1)}$,say, v_{r-3}.

2. From $t = 2$ to $\lceil log_2 n \rceil$, both vertices, u_{r-3} and v_{r-3}, will continue with the same argument so that all vertices at the $r-3-$level are informed.

3. Then, each of the vertices in L_{r-3} is a root of a $4-$CBT and they continue transmitting according to Algorithm Binary.

4. At $t = r$ one of the end-vertices in each $4-$CBT ,which was informed already at $t = r - 1$, will inform the "roots" of all complete binary sub-trees.

The cost of Algorithm Binary 1 is:
The first stage:

$$\sum_{i=0}^{r-5} 2^{i+1}(r-i-4) \ .$$

The second stage:

$$19 \times 2^{r-4} \ .$$

The third stage:

$$\sum_{i=0}^{r-5} 2^{i}(r-i-1) \ .$$

One can verify that the sum of all stages is:

$$\frac{7}{4}2^r - 3r + 3 = \frac{7}{4}n - 3log_2(n+1) + \frac{19}{4} \ .$$

■

Notation: When Algorithm Binary 1 is applied upon a complete binary sub-tree of order $n_k = 2^k - 1$ we write $F_1(k)$, to address the corresponding cost obtained in (4).

Theorem 2.4 *Let T be a $r-CBT$. Then,*

$$F(r, u_1) \leq \frac{7}{4}2^r - r - 5 = \frac{7}{4}n - log_2(n+1) - \frac{13}{4} \ . \tag{5}$$

Proof: In order to prove (5) we present the Algorithm Binary 2:

1. At $t = 1$, u_1 transmits the message to v_{r-3}, and from $t = 2$, v_{r-3} continues according to Algorithm Binary 1.

2. From $t = 2$ to $t = r - 4$, u_1 transmits to the corresponding " u_{r-3} -vertex" in each $T^*_{u_i}$, $2 \leq i \leq r - 4$. Upon receiving the message each of these vertices will continue according to Algorithm Binary 1 (except the $ls(u_{r-4})$ that is the root of a $4-$CBT, and will proceed according to lemma 2.3).

3. At $t = r - 3, r - 2$, u_1 transmits the message to $ls(u_{r-3})$ and $ls(u_{r-2})$, respectively. They will continue by Algorithm Binary.

4. At $t = r - 1$ and $t = r$ u_1 transmits to $ls(u_{r-1})$ and to $rs(u_{r-1})$, respectively.

5. At $t = r$ all end-vertices that received the message at $t = r - 1$ will transmit to the vertices u_i , $i = 2, \cdots, r - 1$.

The cost of Algorithm Binary 2 is:
Stages 1 and 2:

$$\sum_{k=4}^{r-1} F_1(k) + \ r - 4 + (r-4)^2.$$

Stages 3+4:

$$2(r-1) + (r-3) + (r-2).$$

Stage 5:

$$\sum_{k=1}^{r-2} k \ = \ \frac{(r-1)(r-2)}{2}.$$

So that,

$$F(r, u_1) \leq \frac{7}{4}2^r - r - 5 = \frac{7}{4}n - log_2(n+1) - \frac{13}{4} \ .$$

■

Notation:
When Algorithm Binary 2 is applied on a complete binary sub-tree of order $n_k = 2^k - 1$, we write $F_2(k)$ to address the corresponding cost obtained in (5). ■

Theorem 2.5 *Let T be a $r-CBT$ and $u_j \in RP_r$, $j \neq 1, r-3$. Then,*

$$F(r, u_j) \leq \frac{7}{4}2^r - rj + j^2 - 6 = \frac{7}{4}n - j \times log_2(n+1) + j^2 - \frac{17}{4} \ . \qquad (6)$$

Proof: The proof is based upon the previous algorithms. At $t = 1, 2, \cdots, j-1$ u_j will transmit he message to the corresponding " u_{r-3} -vertex" in each of the sub-trees $T^*_{u_k}$, $1 \leq k \leq j-1$.

Upon receiving the message each of these vertices will continue as in the previous algorithms , so that at $t = r$ all vertices $u_1, u_2, \cdots u_{j-1}$, will be informed by end-vertices that received the message at $t = r - 1$.

From $t = j$ to $t = r$ u_j will continue by Algorithm Binary 2.

Then,
$F(r, u_j) \leq \sum_{k=r-j+1}^{r-1} F_1(k) \ + \ F_2(r-j+1) + (r-3)(j-1) + r(j-1) - \frac{j(j-1)}{2} =$
$\frac{7}{4}2^r - rj + j^2 - 6 = \frac{7}{4}n - j \times log_2(n+1) + j^2 - \frac{17}{4}$. ■

With this theorem we completed the proof of the upper bound in our main theorem , theorem 2.1 .

2.3 Line broadcasting in the Binomial Tree(BT)

In order to establish line broadcasting in the Binomial Tree we introduce it first.

Given an originator and broadcast time r, an optimal broadcasting tree, called hereinafter Binomial Tree(BT for short) , is the largest possible tree for which the originator can broadcast a message in time at most r, in a local broadcasting process. This tree has $n = 2^r$ vertices. The tree was introduced first in [4] and for recent treatment one may see also [2]. The BT has actually two originators from which broadcasting can be accomplished within r time units using only local broadcasting. One of the main properties of the BT is that each vertex ,say u , which is not a leaf, in BT, can be an originator of a sub-tree of the same form as the BT, rooted in u. It is well known that $d(BT) = 2r - 1$, where, $d(BT)$ is the diameter of the BT. Let u_1, w_1 be the two originators of the BT. The longest path in the BT ,that is the path which admits the diameter, is denoted by $OP = (u_r, u_{r-1}, \cdots u_1, w_1, w_2, \cdots w_r)$. In Fig. 1 there is an example of a BT with $n = 16$ vertices with a possible sub-division of it into 4 sub-trees , each of which is a BT on 4 vertices.

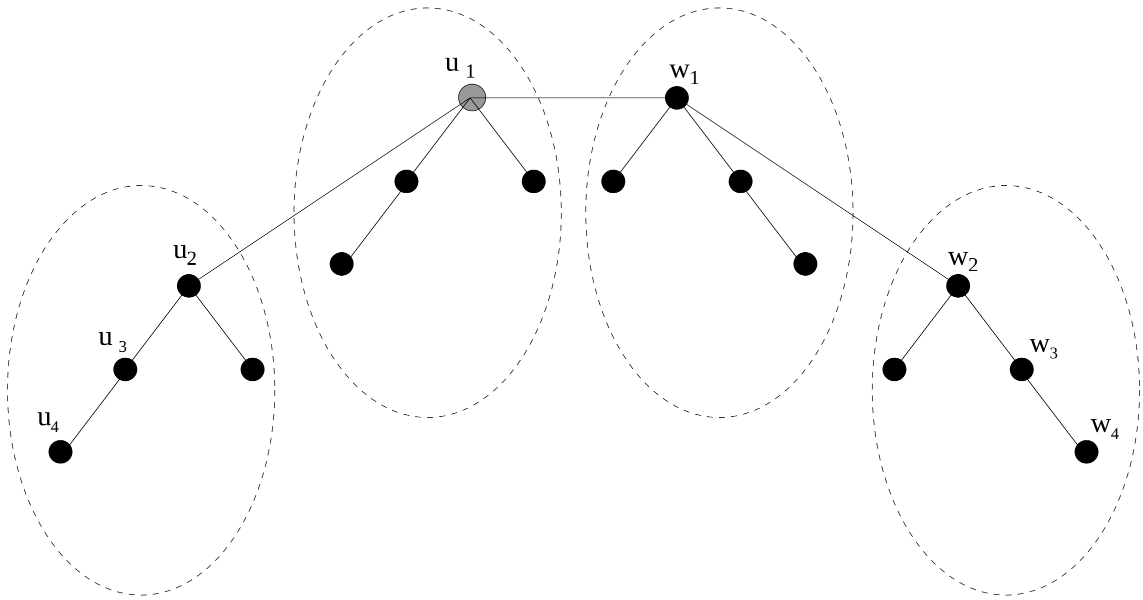

Figure 2: A Binomial Tree for $n = 16$.

We shall view BT with levels L_i , $1 \le i \le r$, where $u_i, w_i \in L_i$, $1 \le i \le r$.

In the next theorem we establish line broadcasting in the BT from any vertex. Observe that while broadcasting from either u_1 or from w_1, it is a local broadcasting where:

$$F(r, u_1) = F(r, w_1) = 2^r - 1 \ .$$

Theorem 2.6 *Given the tree $T = BT$ and $x_j \in L_j$, $1 \leq j \leq r$, be any originator. Then,*

$$F(r, x_j) = 2^r - 1 + \frac{(j-1)j}{2} \ .$$

Proof: W.l.o.g assume x_j is in the "half" tree rooted at w_1. At $t = 1$, x_j transmits the message to u_1, and u_1 will continue broadcasting in T_{u_1}, from $t = 2$ to r, using only local broadcasting. At $t = 2$ x_j will transmit the message to either w_1 or to w_2, depending upon to which binomial sub-tree x_j belongs. In each time unit x_j transmits the message to its furthest originator in the binomial sub-tree results after the previous transmission. The cost is $j + 1$ where, j to reach the root of $x_j's$ sub-tree and 1 to its neighbor. However the cost of the later edge is already counted in the local broadcasting algorithm.

Hence, we obtain,

$$F(r, x_j) \leq 2^r - 1 + \sum_{k=1}^{j-1} k = 2^r - 1 + \frac{(j-1)j}{2} \ . \qquad (5)$$

Now to obtain the lower bound recall that by lemma 2.2 it is easily seen that if an originator is in say, T_{w_1}, he has to transmit first to a vertex in T_{u_1}, and hence to u_1 which is the optimal originator there. So that from $t = 2$ broadcasting will continue in each of the binomial sub-trees independently, which yields exactly the algorithm presented at the beginning of the proof. Thus, the lower bound is that obtained in the R.H.S. of (5), and the proof is completed. ∎

3 Final Remarks

A *caterpillar* is a tree $T(a_1, \ldots, a_t)$ which consists of a main path on t vertices, where a star with a_i edges is attached to vertex number i in the path. The vertices on the main path are the roots of the stars. Thus, T has $t + a_1 + \ldots + a_t$ vertices. Note that the family of caterpillars includes stars (the case $t = 1$), paths (the case where all $a_i = 0$), double stars (the case $t = 2$), etc.

We note that if $t = 2^k$ and the optimal originator is given, then by lemma 2.1 and lemma 2.2 we can establish an efficient line-broadcasting algorithm for caterpillars. Observe that in that case all vertices in the main path of the caterpillar finish broadcasting in the path at the same time, namely, after k time units, so that they start line-broadcasting in the stars attached to them at the same time. Hence, The cost will be the cost of line -broadcasting in the path (see [5]) plus 2^k times the maximum cost to broadcast in a the largest star.

In fact the same holds for a unicyclic graph with 2^k vertices on the cycle. Here for example we can use the cost appeared in [6].

References

[1] A.Averbuch, Y. Roditty, B. Shoham, Efficient line broadcasting in a $d-$ dimensional grid, *(Submitted).*

[2] A. Averbuch, Y. Roditty, B. Shoham, On Time-Relaxed Broadcasting in Communication Networks, *(Submitted).* .

[3] A.M. Farley, Minimum-time line broadcast networks, *Networks 10(1980), 59-70* .

[4] A. Farley, S. Hedetniemi, S. Mitchell and A. Proskurowski, Minimum broadcast graphs, *Discrete Mathematics 25(1979), 189-193.*

[5] S. Fujita and A. Farley, Minimum-cost line broadcasting in paths, *Discrete Applied Mathematics 75(1997), 255-268.*

[6] Jave O. Kane and Joseph G. Peters, Line broadcasting in cycles, *Discrete Applied Mathematics 83(1998), 207-228.*

[7] F. Harary, Graph Theory, *Addison-Wesley, 1969.*

[8] D. West, Introduction to Graph Theory, *Simon @ Schuster A Viacom Company, Upper Saddle River, NJ 07458, (1996).*