

# Broadcasting in Fully Connected Trees

Hovhannes Harutyunyan and Edward Maraachlian  
Department of Computer Science and Software Engineering,  
Concordia University,  
Montreal, Quebec, H3G 1M8, CANADA,  
haruty@cs.concordia.ca, e\_maraac@cs.concordia.ca

**Abstract**—*Broadcasting* is an information dissemination problem in a connected network, in which one node, called the *originator*, disseminates a message to all other nodes by placing a series of calls along the communication lines of the network. Once informed, the nodes aid the originator in distributing the message. Finding the minimum broadcast time of a vertex in an arbitrary graph is NP-complete. The problem is solved polynomially only for trees, unicyclic graphs, and tree of cycles. In this paper we will consider broadcasting in a new class called the Fully Connected Trees (FCT). We will present a  $O(n \log n)$  algorithm to find the broadcast time of any originator in an arbitrary FCT.

## I. INTRODUCTION

Computer networks have become essential in several aspects of modern society. Moreover single processor systems are becoming obsolete because they solve problems serially. Parallelism has become the direction of choice for most modern day computing systems. Even though having several processors on a single system board is possible and widely implemented, one cannot have thousands of processors all connected to a single board. For such a big number of processors there is usually an interconnection network connecting all the processing units. The performance of information dissemination in networks often determines their overall efficiency. One of the fundamental information dissemination problems is broadcasting. Broadcasting is a process in which a single message is sent from one member of a network to all other members. Inefficient broadcasting could degrade the performance of a network seriously. Therefore, it is of a major interest to improve the performance of a network by using efficient broadcasting algorithms.

*Broadcasting* is an information dissemination problem in a connected network, in which one node, called the *originator*, must distribute a message to all other nodes by placing a series of calls along the communication lines of the network. We assume that each call requires one unit of time and involves only one informed node and one of its uninformed neighbors. In one unit of time, many calls can be performed in parallel. A *broadcast scheme* of an originator  $u$  is a set of calls that completes the broadcasting in the network.

Formally, any communication network can be modelled as

a connected graph  $G = (V, E)$ , where  $V$  is the set of vertices (or nodes) and  $E$  is the set of edges (or communication lines) between the vertices in graph  $G$ .

Given a connected graph  $G = (V, E)$  and a message originator, vertex  $u$ , the *broadcast time* of vertex  $u$ ,  $b(u, G)$  or  $b(u)$ , is the minimum number of time units required to complete broadcasting from the vertex  $u$ . The broadcast time  $b(G)$  of the graph  $G$  is defined as  $\max\{b(u) | u \in V\}$ . For surveys of results on broadcasting and related problems, see [13], [6], [14], [15]. The problem of finding the broadcast time of an arbitrary vertex in an arbitrary graph is proved to be NP complete [20]. The problem stays NP-complete even in more restricted classes of graphs such as planar graphs [16] and bounded degree graphs [3] and [18].

Many papers have presented approximation and heuristic algorithms to determine the broadcast time of any vertex in  $G$  (see [1], [4], [5], [7], [8], [19], [21]). The best theoretical upper bound is presented in [5]. Their approximation algorithm generates a broadcast algorithm with broadcast time  $O(\frac{\log(|V|)}{\log \log(|V|)})b(G)$ . The heuristics in [1] and [12] are the best existing heuristics for broadcasting in practice with  $O(|V|^2 \cdot |E|)$  and  $O(|E|)$  respectively.

The broadcast problem has been solved with a linear algorithm only for trees [20], unicyclic graphs [10], [11] and tree of cycles [9]. Also there is an  $O(n^{4k+5})$  broadcast algorithm for partial k-trees [2]. In this paper we consider the broadcast problem in fully connected trees (FCT).

Assume that we have an unweighted complete graph where every vertex is the root of a tree. We will call the resulting graph fully connected trees. We will study the broadcast problem in this kind of graphs.

**Definition 1.** Consider  $n$  trees  $T_i = (V_i, E_i)$  rooted at  $r_i$  where  $1 \leq i \leq n$ . We define the fully connected tree,  $FCT_n = (V, E)$ , to be a graph such that  $V = V_1 \cup V_2 \cup \dots \cup V_n$  and  $E = E_1 \cup E_2 \cup \dots \cup E_n \cup E_{K_n}$  where  $E_{K_n} = \{(r_i, r_j) | 1 \leq i, j \leq n, i \neq j\}$ . The roots of the trees,  $r_i$ , will be called root vertices and the rest of the vertices will be called tree vertices.

The rest of this paper is organized as follows: in the next section we present a broadcast algorithm for fully connected trees where the originator is a root vertex and analyze

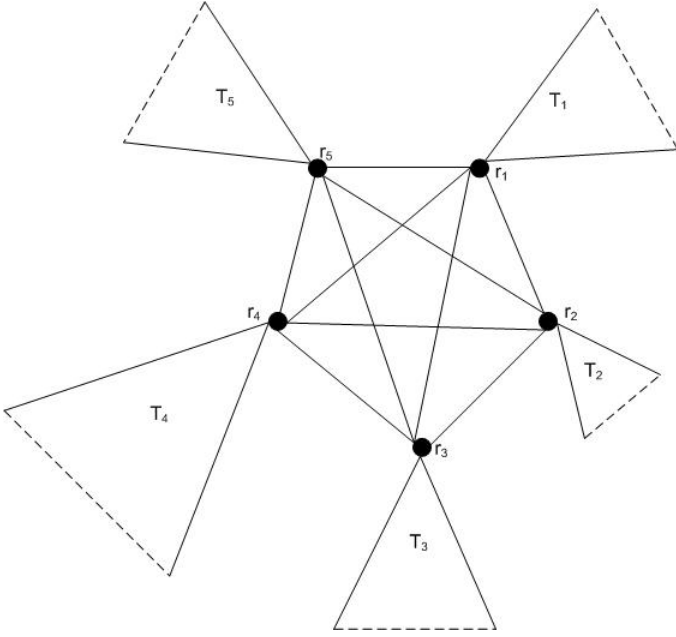


Figure 1. A fully connected tree  $FCT_5$  with 5 trees  $T_i$  rooted at  $r_i$ ,  $1 \leq i \leq 5$ . Note that the roots  $r_i$  induce a subgraph which is the complete graph  $K_5$ .

its complexity. In section 3 we consider the broadcast problem where the originator is a tree vertex. Section 4 is a conclusion.

## II. A BROADCAST ALGORITHM FOR ROOT VERTICES

In this section we consider the broadcast problem in fully connected trees. An upper bound on broadcast time can be obtained if the broadcast algorithm first informs all the vertices of the complete graph and afterwards each vertex informs the tree attached to it. In this case the time needed to inform all the vertices will be  $t_{max} = \lceil \log n \rceil + \max\{b(v_i, T_i)\}$ , where  $b(v_i, T_i)$  is the broadcast time of the vertex  $v_i$  in the tree  $T_i$ . Similarly one can see that a lower bound on the broadcast time is  $t_{min} = \min\{\lceil \log n \rceil, \max\{b(v_i, T_i)\}\}$ .

### A. The Broadcast Algorithm

First we will develop an algorithm that answers the following question: Given a time  $\tau$ , a graph  $FCT_n$ , and an originator  $v_o$ , is it possible to complete broadcasting in  $FCT_n$  in at most  $\tau$  time units? Obviously  $\tau$  should be greater than or equal to  $t_{min}$ . Also, if  $\tau$  is greater than  $t_{max}$  then the answer to the above question is trivially true. The interesting situation is when  $t_{min} \leq \tau \leq t_{max}$ .

The algorithm takes the graph  $FCT_n$ , originator  $v_o$  which is a root vertex, and the candidate broadcast time  $\tau$  as input parameters. The main idea of the algorithm is to do broadcasting assuming that the broadcast time is  $\tau$ . If  $\tau \geq b(v_o, FCT_n)$  then the algorithm will return TRUE and will inform all the vertices of the graph. Otherwise,

it will return FALSE, meaning that  $\tau$  is too small to be the broadcast time. If all the vertices of the graph are informed in  $\tau$  time units or less all we can conclude is that the broadcast time is less than or equal to  $\tau$ . In order to conclude is that  $\tau$  is the broadcast time we need to be able to inform all the vertices in  $\tau$  time units and our algorithm should also return FALSE when  $\tau - 1$  is given as input for the candidate broadcast time.

The first step of the algorithm will be to assign weights to every root vertex of the fully connected tree. The weight of each vertex  $r_i$  is initialized to the broadcast time  $b(r_i, T_i)$ . At every time unit  $t$ , where  $0 \leq t \leq \tau$ , each vertex will have to determine if  $\tau$  is enough to broadcast the information to all the vertices. The algorithm terminates if a vertex can decide that  $\tau$  will not be enough to inform all the vertices of the  $FCT_n$ . Otherwise, the algorithm continues and the informed vertices pass the message to uninformed neighbours. This process iterates until either all the vertices are informed or one of the informed vertices concludes that  $\tau$  cannot be the broadcast time.

When a tree vertex is informed there is not much it can do other than following the well known broadcast algorithm in trees. However, when a root vertex is informed it has the option of passing the information to another root vertex or pass it down to the tree attached to it. Deciding on how every informed root vertex should choose an uninformed neighbour to pass the information to is the main step of the algorithm.

In order to make the description of the algorithm simpler each root vertex  $r_i$  will be assigned a weight  $w(r_i, t)$ . These weightes will be equal to the broadcast time of  $r_i$  in the connected uninformed subtree of  $T_i$  rooted at  $r_i$ . In more details, the weight of each root vertex,  $r_i$ , at any time  $t$  will be calculated as follows: If  $r_i$  has no uninformed children which are tree vertices then its weight is zero because  $r_i$  can do nothing to speed up the process of informing the vertices of the subtree  $T_i$ . If  $r_i$  has one or more uninformed child, let the tree  $T_{i,t}$  rooted at  $r_i$  be the subtree of  $T_i$  which consists of only those vertices of  $T_i$  that are still not informed. The weight of  $r_i$  at time  $t$ ,  $w(r_i, t)$ , will be equal to the broadcast time of  $r_i$  in the tree  $T_{i,t}$ ,  $b(r_i, T_{i,t})$ . Next, we will describe how does an informed root vertex,  $r_i$ , at time  $t$  decide whether it has to inform another root vertex or a tree vertex. Two cases may arise:

- 1)  $r_i$  does not have uninformed children which are tree vertices, in this case  $w(r_i, t) = 0$ . In this case there is nothing  $r_i$  can do to speed up the broadcast process in the tree attached to it. Therefore,  $r_i$  should inform an uninformed root vertex with the highest weight among the vertices.
- 2)  $r_i$  has uninformed children which are tree vertices. In this situation  $r_i$  has 2 options. One is to inform an uninformed root vertex and the other is to inform an uninformed neighbouring tree vertex. The choice

---

**Algorithm 1:** The decision algorithm to decide if a given time  $\tau$  is enough to inform all the vertices of a graph  $FCT_n$ .

---

**Algorithm:** The Decision Algorithm:

*DecisionAlgo*( $G, v, t$ )

**Input:**  $FCT_n = (V, E)$ , originator  $v_o$ , candidate broadcast time  $\tau$

**Output:** FALSE if  $\tau$  cannot be the broadcast time, TRUE if broadcasting can be accomplished in time less or equal to  $\tau$

Initialize  $V_I$  such that  $V_I = v_o$  ;

**foreach**  $t$  such that  $0 \leq t \leq \tau - 1$  **do**

**foreach**  $v \in V_I$  **do**

**if**  $v$  is a root vertex **then**

**if**  $w(v, t) > \tau - t$  **then**

$v$  informs another uninformed root vertex at time  $t$  which has the highest weight among the uninformed root vertices ;

                Append the newly informed vertex to  $V_I$  ;

**else**

**if**  $w(v, t) = \tau - t$  **then**

$v$  informs one of its children which are tree vertices ;

                Append the newly informed vertex to  $V_I$  ;

**else**

            return FALSE ;

**end**

**end**

**else**

$v$  informs a tree vertex based on the tree broadcast algorithm in the uninformed subtree rooted at  $v$  ;

        Append the newly informed vertex to  $V_I$  ;

**end**

**end**

return TRUE ;

---

between informing a root vertex versus a tree vertex is done by comparing the time needed to inform the uninformed subtree attached to it,  $b(r_i, T_{i,t}) = w(r_i, t)$ , with the remaining time  $\tau - t$ . If  $\tau - t > w(r_i, t)$  then  $v$  informs another root vertex. If  $\tau - t = w(r_i, t)$  then  $v$  has to inform one of its children i.e. a tree vertex according to the broadcast algorithm in trees. The case where  $\tau - t < w(r_i, t)$  is not being considered here because as soon as that happens we conclude  $\tau$  cannot be the broadcast time and the algorithm return FALSE.

In Algorithm 1 we describe the decision algorithm that given a graph  $FCT_n$ , an originator  $v_o$ , and a candidate broadcast time  $\tau$ , decides if the broadcast time of the graph

is less than or equal that  $\tau$ . If the candidate time  $\tau$  happens to be the broadcast time, then this algorithm also generates the optimal broadcast scheme. Note that in the psuedocode  $V_I$  represents the set of informed vertices which at the start of the algorithm contains only  $v_o$ .

---

**Algorithm 2:** The broadcast algorithm of an root vertex originator  $v_o$  in a graph  $FCT_n$ .

---

**Algorithm:** The Broadcast Algorithm

*BroadcastAlgorithmFCT<sub>n</sub>*( $G, v_o, t_1, t_2$ )

**Input:**  $FCT_n = (V, E)$ , originator  $v_o$ , the minimum of a time range, and the maximum of a time range.

**Output:** Broadcast time  $\tau$  such that  $\tau = b(v_o, FCT_n)$

$t = t_1 + \lfloor \frac{t_2 - t_1}{2} \rfloor$  **if**  $t_1 = t_2$  **then**

**if** *DecisionAlgo*( $G, v_o, t_1$ ) **then**

        return  $t$

**else**

        return -1

**end**

**end**

**if**  $t_1 + 1 = t_2$  **then**

**if** *DecisionAlgo*( $G, v_o, t_1$ ) AND

    !*DecisionAlgo*( $G, v_o, t_2$ ) **then**

        return  $t_1$

**end**

**if** !*DecisionAlgo*( $G, v_o, t_1$ ) AND

*DecisionAlgo*( $G, v_o, t_2$ ) **then**

        return  $t_2$

**else**

        return -1

**end**

**end**

**if** *DecisionAlgo*( $G, v_o, t_1$ ) **then**

    return *BroadcastAlgorithmFCT<sub>n</sub>*( $G, v_o, t_1, t$ )

**else**

    return

*BroadcastAlgorithmFCT<sub>n</sub>*( $G, v_o, t + 1, t_2$ )

**end**

---

Now we describe a broadcast algorithm for completely connected trees using the decision algorithm presented above. The algorithm does a binary search for the broadcast time in the range of possible values. As mentioned above we already know the minimum and maximum of the range which are  $t_{min}$  and  $t_{max}$ . The binary search reduces the size of the range by applying the decision algorithm on the midpoint of the range. If the algorithm returns that broadcasting can be performed, then the lower half of the range is considered for the recursive call of the search algorithm. However, if the result of the algorithm is negative, meaning that the midpoint of the range cannot be the broadcast time, then the upper half of the range is considered and the algorithm is again applied recursively. The psuedocode of the algorithm is presented

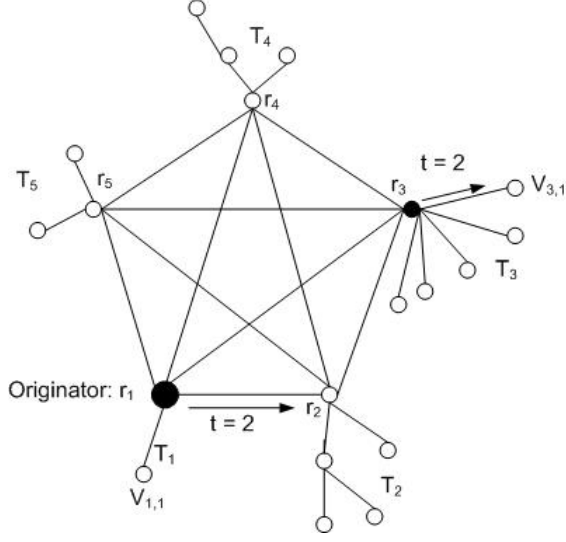


Figure 2. A fully connected tree  $FCT_5$  with 5 trees  $T_i$  rooted at  $r_i$ ,  $1 \leq i \leq 5$ . The originator is  $r_1$  and at time  $t = 1$  the informed vertices are  $r_1$  and  $r_3$ . This figure shows that at time  $t = 2$  the two vertices that get informed are  $r_2$  and  $v_{3,1}$ .

in Algorithm 2. The initial call of the algorithm will be  $BroadcastAlgorithmFCT_n(FCT_n, v_o, t_{min}, t_{max})$  where  $t_{min} = \min\{\lceil \log n \rceil, \max b(v_i, T_i)\}$  and  $t_{max} = \lceil \log n \rceil + \max\{b(v_i, T_i)\}$ .

We will not present the correctness proof of the broadcast algorithm because of space limitation. Finally in figure 2 we provide a simple example which depicts the different operations performed by the decision algorithm. We are given a graph which is a fully connected tree with 5 trees,  $T_i$  where  $1 \leq i \leq 5$ . The originator is vertex  $r_1$ , the root of tree  $T_1$ , and the candidate broadcast time  $\tau = 6$ . The figure represents a snapshot of the state of the graph at time  $t = 1$  where the informed vertices are  $r_1$  and  $r_3$ . In order to decide which vertices should be informed next at time  $t = 2$ , the weights  $w(v_i, t)$  should be calculated for  $t = 1$ . Since there are no tree vertices informed yet, the weights will be calculated as follows:  $w(v_i, 1) = b(v_i, T_i)$ . Hence, we obtain that  $w(r_1, 1) = 1$ ,  $w(r_2, 1) = 3$ ,  $w(r_3, 1) = 5$ ,  $w(r_4, 1) = 2$ , and  $w(r_5, 1) = 2$ . The remaining time  $t_{rem} = \tau - t$  is equal to  $t_{rem} = 6 - 1 = 5$ . None of the informed vertices has a weight greater than 5 so the algorithm does not return FALSE. The informed vertices  $r_1$  and  $r_3$  inform new vertices at time  $t = 2$  as follows: Vertex  $r_3$  has weight  $w(r_3, 1) = 5 = t_{rem}$ , therefore it informs a tree vertex based on the well known tree algorithm. However, vertex  $r_1$  informs another root vertex since  $w(r_1, 1) = 1 < t_{rem} = 5$ . It informs vertex  $r_2$  since it has the greatest weight among the uninformed root vertices. The weights of the vertices  $w(r_i, 2)$  remain the same for all vertices except  $r_3$ . Since  $r_3$  has one child tree vertex that is informed, its weight  $w(r_3, 2) = 4$ .

## B. Complexity Analysis

In this section we will calculate the complexity of the algorithm described above. First one can note that the  $BroadcastAlgorithm(G, v, t_1, t_2)$  does a binary search for the broadcast time in the range of possible values. The complexity of a binary search algorithm is  $O(\log N)$  where  $N$  is the number of values in the range that is being searched in. However, every time we need to verify if a certain value in the range is less than, greater than, or equal to what we are looking for, we are running the  $DecisionAlgo(G, v, t)$  which has a linear complexity in the number of vertices of the graph  $G$ . Assume that  $n$  is the number of vertices of the graph  $G$ . The  $DecisionAlgo(G, v, t)$  can calculate its decision in a linear time because every root vertex has to calculate its weight and compare with the remaining time. Once the weights are calculated once at the beginning of the algorithm, updating the weights at every new time unit can be done in a constant time. Also initializing the weights at the beginning of the algorithm is a linear operation in terms of the number of vertices of the graph because the tree broadcast algorithm has to run which is linear itself. Therefore, the complexity of the  $BroadcastAlgorithm(G, v, t_1, t_2)$  has a complexity of  $O(n \log(t_2 - t_1))$ . Also note that  $t_1$  and  $t_2$  can be also linear in  $n$  which implies that the complexity of the algorithm is  $O(n \log n)$ .

## III. BROADCASTING FROM ANY ORIGINATOR

In the previous sections we assumed that the originator is always one of the root vertices. In this section we will develop a broadcast algorithm for any originator in an arbitrary fully connected tree  $FCT$ . Assume we are given a fully connected tree  $G$  such that the originator  $v_o$  is in the tree  $T_i$  rooted at one of the root vertices  $r_i$ . There is a unique path  $P$  in  $T_i$  connecting  $r_i$  to the originator  $v_o$ . The vertex on the path  $P$  neighbouring  $r_i$  will be denoted by  $v_i$ . Let  $u_j$ ,  $1 \leq j \leq k$ , be the neighbors of  $v_o$  in the tree. One of these vertices falls on the path  $P$ , call this vertex  $u_i$ . The subtree of  $T_i$  rooted at the vertex  $v_i$  will be called  $T$  (see Fig 3a). The remaining subtree of  $T_i$ , rooted at  $r_i$ , after removing the edge  $(r_i, v_i)$  will be called  $T'_i$ . We will construct a new graph  $G'$  which is again a fully connected tree but the tree  $T'_i$  is attached at the root vertex  $r_i$  instead of the tree  $T_i$ . Figure 3a shows all the details described above. It is worth noticing that one can redraw the graph  $G$  differently by drawing the tree  $T$  rooted at the originator  $v_o$  and vertex  $v_i$  as one of its leaf vertices, this is shown in figure 3b. It can be observed that the graph  $G'$  is attached to the tree  $T$  by a bridge  $(v_i, r_i)$ . Since graph  $G'$  is connected to the tree  $T$  by a bridge, the broadcast algorithm in  $G'$  is independent of the broadcast algorithm in  $T$ . Once vertex  $r_i$  is informed, it can not inform any other vertex in  $T$  so its job is to inform the vertices of  $G'$  in the fastest possible way. However, since  $G'$  is a fully connected tree and  $r_i$  is a root vertex, we have a broadcast algorithm to solve the broadcast

problem of vertex  $r_i$  in  $G'$ . Therefore, the main remaining difficulty is finding the time at which the vertex  $r_i$  should be informed so that the broadcast time in the whole graph  $G$  is optimal.

In order to answer this question we will use the ideas from the algorithm for broadcasting in trees. Normally, vertex  $v_o$  informs a child vertex that has the highest broadcast time in the subtree rooted at it. The subtrees rooted at the children of  $v_o$  are labeled by  $H_j$ ,  $1 \leq j \leq k$ , as shown in figure 3b. The broadcast times  $b(u_j, H_j)$ ,  $1 \leq j \leq k$ , can be easily calculated except for the case where  $u_j = u_i$ . This is the case where there is the graph  $G'$  attached to  $v_i$  which might change the time needed by  $u_i$  to inform all of the vertices of  $H_i$  and  $G'$ . Since  $G'$  is a fully connected tree we can solve the broadcast problem for the originator  $r_i$  and obtain a broadcast tree  $T_{G'}$ . We construct a tree  $H'_i$  by attaching the tree  $H_i$  to  $T_{G'}$  by the edge  $(v_i, r_i)$ . The time needed for  $u_i$  to inform all the vertices of  $H_i$  and  $G'$  is equal to the broadcast time of the vertex  $u_i$  in the tree  $H'_i$ .

In conclusion, the broadcast problem of a non root vertex in an arbitrary fully connected tree can be solved by solving two problem: one is the broadcast problem in a fully connected tree with an originator that is a root vertex and the second one is the broadcast problem in a tree. The complexity of the problem still remains  $O(n \log n)$  because finding the broadcast time of a tree is linear and hence the complexity is determined by the algorithm that calculates the broadcast time of a root vertex in an arbitrary FCT.

#### IV. CONCLUSION

In this paper we present a  $O(n \log n)$  algorithm to find the broadcast time of any originator in an arbitrary Fully Connected Trees (FCT). This result can be used in designing polynomial algorithms for more complicated network topologies that involve cliques.

#### REFERENCES

- [1] R. Beier, J. F. Sibeyn, A powerful heuristic for telephone gossiping, *Proc. of the 7th International Colloquium on Structural Information & Communication Complexity, SIROCCO'00*, L'Aquila, Italy, 2000, pp. 17-36.
- [2] A. Dessmark, A. Linga, H. Olsson, and H. Yamamoto. Optimal Broadcasting in Almost Trees and Partial k-trees, *Proceedings of the Symposium on Theoretical Aspects of Computer Science, STACS'98*, Springer-Verlag, 1998, pp.432-443, UK.
- [3] M. J. Dinneen. The complexity of broadcasting in bounded-degree networks, *Tech. Report LACES-05C94 -31*, Los Alamos National Laboratory, 1994
- [4] M. Elkin, G. Kortsarz. A combinatorial logarithmic approximation algorithm for the directed telephone broadcast problem, *Proc. of ACM Symp. on Theory of Computing, STOC'02*, 2002, pp. 438-447.
- [5] M. Elkin and G. Kortsarz, Sublogarithmic approximation for telephone multicast: path out of jungle, *Proc. of Symposium on Discrete Algorithms, SODA'03*, Baltimore, Maryland, 2003, pp. 76-85.
- [6] P. Fraigniaud, E. Lazard, Methods and problems of communication in usual networks, *Discrete Appl. Math.* 53 (1994) 79-133.
- [7] P. Fraigniaud, S. Vial, Approximation algorithms for broadcasting and gossiping, *J. Parallel and Distrib. Comput.* 43(1) (1997) 47-55.
- [8] P. Fraigniaud, S. Vial, Comparison of Heuristics for One-to-All and All-to-All Communication in Partial Meshes, *Parallel Processing Letters*, 9(1), 1999, pp. 9-20.
- [9] H. A. Harutyunyan, E. Maraachlian, Linear Algorithm for Broadcasting in Networks with No Intersecting Cycles, *International Conference on Parallel and Distributed Processing Techniques and Applications, PDPTA'09*, 2009, pp. 296 - 301.
- [10] H. A. Harutyunyan, E. Maraachlian, On Broadcasting in Unicyclic Graphs, *Journal of Combinatorial Optimization* Vol 16,(2008), pp. 307 - 322.
- [11] H. A. Harutyunyan, E. Maraachlian, Linear Algorithm for Broadcasting in Unicyclic Graphs (Extended Abstract), *Proc. 13th Annual COCOON 2007 LNCS 4598* (2007), pp. 372 - 383.
- [12] H. A. Harutyunyan, B. Shao, An Efficient Heuristic for Broadcasting in Networks, *Journal of Parallel and Distributed Computing* 66(1), 2006, pp. 68-76.
- [13] S.M Hedetniemi, S.T. Hedetniemi, A.L. Liestman, A survey of gossiping and broadcasting in communication networks, *Networks*, 18 (1988) 319-349.
- [14] J. Hromkovic, R. Klasing, B. Monien, R. Peine, Dissemination of information in interconnection networks, *Combinatorial Network Theory, D.-Z. Du, D.F.Hsu(eds.)*, Kluwer Academic Publishers, 1996, pp. 125-212.
- [15] J. Hromkovic, R. Klasing, A. Pelc, P. Ruzicka, W. Unger, Dissemination of Information in Communication Networks - Broadcasting, Gossiping, Leader Election, and Fault-Tolerance, Springer 2005.
- [16] A. Jakoby, R. Reischuk, C. Schindelhauer, The Complexity of Broadcasting in Planar and Decomposable Graphs, *Discrete Applied Mathematics*, Vol. 83, 1998, 179-206.
- [17] D. Johnson and M. Garey, Computers and Intractability: A Guide to the Theory of NP-Completeness (Freeman, San Francisco, CA, 1979)
- [18] M. Middendorf, Minimum broadcast time is NP-complete for 3-regular planar graphs and deadline 2, *Information Processing Letters*, 46(6): 281-287, 1993.
- [19] R. Ravi, Rapid Rumor Ramification: Approximating the minimum broadcast time, *Proc. of 35th Symposium on Foundation of Computer Science, FOCS'94*, 1994, pp. 202-213.
- [20] P. J. Slater, E. J. Cockayne, S. T. Hedetniemi, Information dissemination in trees, *SIAM J. Comput.* 10(4) (1981) 692-701.

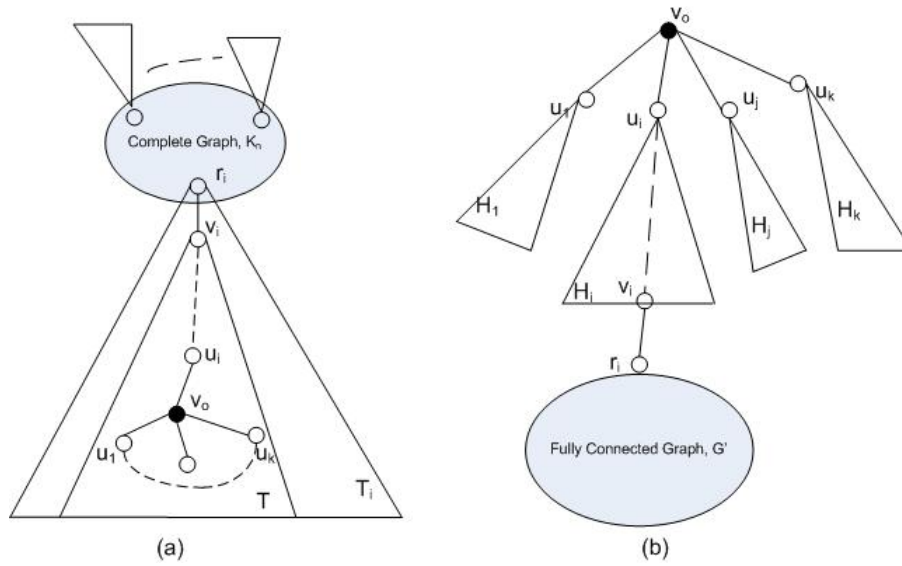


Figure 3. A fully connected tree  $G$  with an originator which is not a root vertex. (a) shows graph  $G$  and one of its trees  $T_i$  which contains the originator. (b) A different way of drawing the graph  $G$  by separating the tree  $T$  and the fully connected tree  $G'$ .

- [21] P. Scheuerman, G. Wu, Heuristic Algorithms for Broadcasting in Point-to-Point Computer Network, *IEEE Transactions on Computers* C-33(9) (1984) 804-811.