Broadcasting in Heterogeneous Tree Networks[*]

Yu-Hsuan Su[1], Ching-Chi Lin[2], and D.T. Lee[3]

[1] Graduate Institute of Electronic Engineering, National Taiwan University, Taiwan
d98943034@ntu.edu.tw
[2] Department of Computer Science and Engineering,
National Taiwan Ocean University, Taiwan
lincc@mail.ntou.edu.tw
[3] Institute of Information Science, Academia Sinica, Taiwan
dtlee@iis.sinica.edu.tw

Abstract. We consider the broadcasting problem in heterogeneous tree networks. A heterogeneous tree network is represented by a weighted tree $T = (V, E)$ such that the weight of each edge denotes the communication time between the two end vertices. The broadcasting problem is to find a broadcast center such that the maximum communication time from the broadcast center to all vertices is minimized. In this paper, we propose a linear time algorithm for the broadcasting problem in a heterogeneous tree network following the postal model. As a byproduct of the algorithm, we can compute in linear time the broadcasting time of any vertex in the tree, i.e., the maximum time required to transmit messages from the vertex to every other vertex in the tree. Furthermore, an optimal sequence by which the broadcast center broadcasts its messages to all vertices in T can also be determined in linear time.

Keywords: algorithm, broadcast center, heterogeneous network, weighted tree, postal model.

1 Introduction

A heterogeneous network is a network connecting workstations with different operating systems and communication protocols. Thus, the times to communicate between any pair of workstations may be different. A heterogeneous network is represented by a weighted graph $G = (V, E)$, in which $V(G)$ represents a set of workstations and each edge $\overline{u, v} \in E(G)$ represents a connection between two adjacent workstations. The weight of each edge represents the transmission time β required to transmit messages between two adjacent workstations. In heterogeneous networks, each communication link may have different message transmission time, and each workstation may need connection time α to set up connection between two adjacent workstations to complete the message transmission. In this paper, we consider the broadcasting problem in a heterogeneous tree

network $T = (V, E)$, where the weight $w(u, v)$ of each edge $\overline{u, v} \in E(G)$, denotes the transmission time required. The broadcasting problem has been extensively studied for several decades [1, 6–9, 11–13] due to the increasing demands of heterogeneous network of workstations [7, 9]. The broadcasting problem is to find a broadcast center such that the broadcasting time from the broadcast center to all vertices in T is minimized. We consider the problem following the *postal model* [1–4, 9] as described below.

The postal model makes distinction between *connection time* α and *transmission time* β, where $\alpha > 0$ is assumed to be a constant and $\beta \geq 0$ varies from edge to edge. More specifically, the postal model assumes that the sender requires α time to set up a connection. After the sender sets up the connection, the sender is allowed to set up another connection to the next receiver while the sender is still transmitting the messages to the current receiver. For example, the sender v first sets up the connection with the receiver u_1 in α time, and then v can set up another connection to the next receiver u_2 while still transmitting the messages to u_1. The message transmission between two vertices is referred as a "call". A sender can only set up one connection to make a call to a receiver per α time and a sender can only transmit the messages to adjacent vertices. A call is said to be completed only after the sender completes transmitting the messages to the receiver. That is, if the transmission time β between the sender v and receiver u is $w(u, v)$, and the receiver u is the ith vertex called by the sender, then the call from v to u will be completed after $\alpha i + w(u, v)$ time. The receiver u cannot forward the messages until the sender completes the transmission.

In contrast to the postal model is the *telephone model*, in which the sender is allowed to set up another connection to the next receiver only after completing the transmission of messages to the current receiver. That is, only after the sender completes the transmission, can the sender set up another connection to the next receiver. For example, the sender v first sets up a connection with the receiver u_1 after α time, then the sender v can only set up another connection to the next receiver u_2 after $\alpha + w(u_1, v)$ time. For both the postal and telephone models, the receiver cannot forward the messages until the receiver finishes the receipt of the messages from the sender. Moreover, we assume that the message transmission between two adjacent vertices is *full-duplex*, i.e., the sender and receiver can exchange the messages between them simultaneously.

Some notations are introduced in order to give a formal definition of the broadcasting problem. Given a weighted tree $T = (V, E)$, the *broadcasting time* of v, denoted as $b(v, T)$, is the minimum time required to broadcast a message from v to all vertices in T. The *broadcasting time* of T, denoted as $b(T)$, is the minimum broadcasting time of any vertex $v \in V(T)$, i.e., $b(T) = \min\{b(v, T) | v \in V(T)\}$. The *broadcast center* $BC(T)$ of T is the set of vertices with the minimum broadcasting time, i.e., $BC(T) = \{v \mid v \in V(T), \ b(v, T) = b(T)\}$.

Problem Definition: Given a weighted tree $T = (V, E)$ in which the weight $w(u, v) \geq 0$ of an edge $\overline{u, v}$ represents the transmission time between them, the broadcasting problem is to compute the set of broadcast centers $BC(T)$

and determine the broadcasting time $b(T)$, following the postal model with a constant connection time $\alpha > 0$. We use n to denote the size of the tree T.

1.1   Previous Work

The telephone model and the postal model are the two most popular communication models in the literature. In 1978, Slater *et al.* [13] showed that computing the broadcasting time of a given vertex in an arbitrary unweighted graph is NP-complete following the telephone model. Furthermore, they also provided an $O(n)$-time algorithm to compute the set of broadcast centers and determine the broadcasting time for the unweighted trees following the telephone model. By adopting Slater *et al.*'s algorithm, Koh and Tcha [10] extended the results to provide an algorithm for the weighted trees following the telephone model. The algorithm runs in $O(n \log n)$ time due to the use of a sorting procedure and a priority queue.

For the telephone model in an unweighted graph G, Farley [5] determined the lower and upper bounds of the time required to broadcast m messages from a given vertex v to all vertices in G. For the postal model in an unweighted complete graph G, Bar-Noy and Kipnis [2] presented an algorithm to determine the minimum broadcasting time from a given broadcast center to all vertices in G. Further, they showed that the algorithm is optimal which runs in $\Theta(\lambda \log n / \log(\lambda + 1))$ time, where λ is the communication latency, i.e., the time needed when the sender starts sending messages until the receiver completes the transmission.

Given a graph G and two nonempty subsets $A \subseteq V(G)$ and $B \subseteq V(G)$ of vertices in G, the multicasting problem is to determine the minimum time to broadcast messages from all vertices in A to all vertices in B. For the telephone model in an unweighted complete graph, Khuller *et al.* [8] presented polynomial-time approximation algorithms for the single-source multicasting, i.e., $|A| = 1$, multi-source broadcasting problems, i.e., $B = V$, and multi-source multicasting problems.

In this paper, we consider the broadcasting problem in heterogeneous tree networks following the postal model. Note that our problem becomes the broadcasting problem for unweighted trees following the telephone model when $\alpha = 1$ and $\beta = 0$ for all edges.

1.2   Contributions

In this paper, we propose an $O(n)$-time algorithm for the broadcasting problem in heterogeneous tree networks following the postal model. Similar to the algorithm by Koh and Tcha [10], our algorithm is based on the concept of Slater *et al.*'s algorithm. But unlike their algorithm which uses a priority queue and a sorting procedure, resulting in an $O(n \log n)$-time algorithm, we develop an $O(n)$-time algorithm by using a new observation and a non-sorting labeling method. The two major refinements lead to a time complexity improvement from $O(n \log n)$ to $O(n)$. We further show that an optimal sequence of calls can also be obtained in $O(n)$ time.

For determining the broadcast centers $BC(T)$ in T, Slater *et al.* [13] use a bottom-up approach to iteratively update the labels of vertices and remove leaf nodes with the smallest label. Since the edges are unweighted, the value of labels are integers in $\{0, \ldots, n-1\}$. They exploit this fact to create a useful data structure to avoid the use of a priority queue and a sorting procedure. Since the labels are real numbers in our problem, it is not clear how to avoid the use of priority queue, which takes $O(\log n)$ time per operation, if one needs to select a vertex with the smallest label. However, we prove in Lemma 4 that it is sufficient to select between any two labeled vertices the one with the smaller label. With this new observation, we can perform this task in $O(n)$ time. As for update of labels of vertices in T, we propose a brand-new non-sorting procedure, and show in Theorem 1 its details.

We summarize our contributions as follows.

- The set of broadcast centers $BC(T)$ can be computed and the broadcasting time $b(T)$ can be determined in $O(n)$ time.
- An optimal sequence of calls by which the broadcast center broadcasts to all vertices in T can be determined in $O(n)$ time.
- Given a vertex $v \in V(T)$, the broadcasting time $b(v, T)$ and an optimal sequence of calls from v to all vertices in T can be determined in $O(n)$ time.

1.3   Organization

The rest of the paper is organized as follows. Section 2 describes the algorithm BROADCAST which computes the set of broadcast centers $BC(T)$ and determines the broadcasting time $b(T)$ following the postal model. In Section 3, we provide the correctness proof and timing analysis of the algorithm. Finally, we give concluding remarks and suggest some direction for future work in Section 4.

2   Algorithm BROADCAST

The algorithm adopts a greedy strategy to process the vertices in T in a bottom-up manner. For an edge $\overline{u,v}$ in T, the removal of this edge will result in two subtrees, each of which contains v and u respectively. The subtree of T containing v is denoted as $T(v, u)$, and the subtree of T containing u is denoted as $T(u, v)$. Recall that $b(v, T)$ denotes the minimum time required to broadcast a message from v to all vertices in T. Suppose that $u_1, u_2, \ldots, u_k$ are the neighbors of v in T such that $b(u_1, T(u_1, v)) + w(u_1, v) \geq b(u_2, T(u_2, v)) + w(u_2, v) \geq \ldots \geq b(u_k, T(u_k, v)) + w(u_k, v)$. If we want to broadcast messages from v to all vertices in T, it is not difficult to see that an optimal sequence of calls from v to its neighbors would be ordered as $u_1, u_2, ..., u_k$. Hence, the broadcasting time from v to all vertices in T is $b(v, T) = \max\{b(u_i, T(u_i, v)) + w(u_i, v) + \alpha i | 1 \leq i \leq k\}$.

Based on the above observation, the concept of our algorithm BROADCAST is as below. The algorithm initially assigns a label $t(u) = 0$ to each leaf node u. It then removes a leaf u in the current tree T iteratively and assigns a label

$t(v)$ to the vertex v, which becomes a leaf in the tree $T - \{u\}$. We always keep the leaf vertex with the largest label when removing a leaf node. It is one of the most significant differences of our algorithm from the algorithm by Slater *et al.* [13]. We will show that the tree $T - \{u\}$ still contains a broadcast center in T after removing the vertex u. Each vertex in T would be labeled exactly once except the last remaining vertex, which gets labeled twice. Let κ denote the last remaining vertex in the rest of this paper. For a vertex $v \neq \kappa$ in T, let v' be the neighbor of v such that v' is on the path from v to κ. We will show that $t(v) = b(v, T(v, v'))$, which is the minimum broadcasting time from v to all vertices in $T(v, v')$. The algorithm is detailed below.

Algorithm 1. Algorithm BROADCAST

Input: A weighted tree $T = (V, E)$ with weight $w(u, v) \geq 0$ for $\overline{u, v}$ in $E(T)$.
Output: The broadcasting time $b(T)$ and the set of broadcast centers $BC(T)$.
 1: **for** each leaf $\ell \in T$ **do** $t(\ell) \leftarrow 0$;
 2: let $BC(T) \leftarrow \emptyset, W \leftarrow \emptyset$ and $T' \leftarrow T$;
 3: **while** $|V(T')| \geq 2$ **do**
 4:     select two leaves u_x and u_y in T' arbitrarily;
 5:     let $t(u) \leftarrow \min\{t(u_x), t(u_y)\}$, $W \leftarrow W \cup \{u\}$ and $T' \leftarrow T' - \{u\}$;
        let v be the vertex adjacent to u in T';
 6:     **if** v is a leaf in T' **then**
 7:         suppose that v is adjacent to labeled vertices $u_1, u_2, \ldots, u_k$ in W such that
            $t(u_1) + w(u_1, v) \geq t(u_2) + w(u_2, v) \geq \ldots \geq t(u_k) + w(u_k, v)$;
            let $t(v) \leftarrow \max\{t(u_i) + w(u_i, v) + \alpha i | \ 1 \leq i \leq k\}$;
 8:     **end if**
 9: **end while**
10: let κ be the only vertex left in T', $b(T) \leftarrow t(\kappa)$ and $BC(T) \leftarrow \{\kappa\}$;
11: let the neighbors of κ in T be $u_1, u_2, \ldots, u_k$ such that
    $t(u_1) + w(u_1, \kappa) \geq t(u_2) + w(u_2, \kappa) \geq \ldots \geq t(u_k) + w(u_k, \kappa)$;
12: let h be the smallest integer such that $t(u_h) + w(u_h, \kappa) + \alpha h + \alpha > b(T)$;
13: $BC(T) \leftarrow BC(T) \cup \{u_i | \ w(u_i, \kappa) = 0 \text{ and } i \leq h\}$.

In the beginning of the algorithm, we set $t(\ell) = 0$ for each leaf ℓ in T. Next, in each iteration of the while loop, to keep the vertex with the largest label, we arbitrarily select two leaves u_x and u_y in the current tree T. If $t(u) = \min\{t(u_x), t(u_y)\}$, then we remove u from the current tree T. Suppose that v is the vertex adjacent to u in the current tree T. If v becomes a leaf in $T - \{u\}$, we set $t(v) = \max\{t(u_i) + w(u_i, v) + \alpha i | \ 1 \leq i \leq k\}$, where $u_1, u_2, \ldots, u_k \in W$ are the neighbors of v with $t(u_i) + w(u_i, v) \geq t(u_{i+1}) + w(u_{i+1}, v)$ for $1 \leq i \leq k - 1$. After the execution of the while loop, we have only one vertex left, denoted as κ, which is one of the broadcast centers in T. Moreover, the neighbors of κ are the only candidates for being a broadcast center of T. Suppose that h is the smallest integer such that $t(u_h) + w(u_h, \kappa) + \alpha h + \alpha > b(T)$. Let $N(\kappa)$ be the neighbors of κ in T. We will prove that for each vertex $u_i \in N(\kappa)$, u_i is a broadcast center in T if and only if $w(u_i, \kappa) = 0$ and $i \leq h$.

3   Correctness and Complexity Analysis

Let v be an arbitrary vertex in T and $u_1, ..., u_k$ be the neighbors of v. Let $\bar{T} = \bigcup_{1 \leq i \leq k} \{(v, u_i) + T(u_i, v)\}$ be a rooted tree of T with v as the root, and the neighbors of v are ordered so that $b(u_i, T(u_i, v)) + w(u_i, v) \geq b(u_{i+1}, T(u_{i+1}, v)) + w(u_{i+1}, v)$ for $1 \leq i \leq k - 1$.

Intuitively, in order to shorten the broadcasting time from v to all vertices in $\bar{T}$, we will transmit the message to u_1 first, then u_2, u_3, and so on. The following lemma shows that $u_1, ..., u_k$ is in fact an optimal sequence of calls to broadcast messages from v to its neighboring vertices in $\bar{T}$. Then, it follows that the minimum time required to broadcast from v to all vertices in $\bar{T}$ is $b(v, \bar{T}) = \max\{b(u_i, T(u_i, v)) + w(u_i, v) + \alpha i \mid 1 \leq i \leq k\}$.

Lemma 1. *Let v be an arbitrary vertex in T and $u_1, ..., u_k$ be the neighbors of v. Let $\bar{T} = \bigcup_{1 \leq i \leq k} \{(v, u_i) + T(u_i, v)\}$ be a rooted tree of T with v as the root, and the neighbors of v are ordered so that $b(u_i, T(u_i, v)) + w(u_i, v) \geq b(u_{i+1}, T(u_{i+1}, v)) + w(u_{i+1}, v)$ for $1 \leq i \leq k - 1$. Then, $u_1, u_2, \ldots, u_k$ is an optimal sequence of calls to broadcast messages from v to neighboring vertices in $\bar{T}$. Consequently, $b(v, \bar{T}) = \max\{b(u_i, T(u_i, v)) + w(u_i, v) + \alpha i \mid 1 \leq i \leq k\}$.*

Then we show an important result, on which the correctness proof of our algorithm is based, and also some useful properties for finding a broadcast center.

Lemma 2. *For each vertex v in T we have $t(v) = b(v, T(v, v'))$.*

Proof. Let $v_1, v_2, \ldots, v_n$ be the elimination order of vertices in T such that v_i is removed from T' before v_j if $i < j$. We prove the statement by induction on the number of vertices. Let vertex v_1' be the neighbor of v_1 on the path from v_1 to $\kappa = v_n$. Clearly, v_1 is a leaf in T and so we have $t(v_1) = b(v_1, T(v_1, v_1')) = 0$. Suppose that the statement holds for $i = k$. We consider $i = k + 1$ below.

We first consider the case when v_{k+1} is a leaf in T. Clearly, $t(v_{k+1}) = b(v_{k+1}, T(v_{k+1}, v_{k+1}')) = 0$, where v_{k+1}' is the neighbor of v_{k+1} on the path from v_{k+1} to κ. Next, we consider the case when v_{k+1} is an internal vertex in T. Suppose that $v_{k+1}', u_1, ..., u_\ell$ are the neighbors of v_{k+1} in T, and without loss of generality, let we assume that $t(u_i) + w(u_i, v) \geq t(u_{i+1}) + w(u_{i+1}, v)$, for $i = 1, ..., \ell - 1$. Notice that $u_i \in \{v_1, ..., v_k\}$ for $1 \leq i \leq \ell$. By induction hypothesis, we have $t(u_i) = b(u_i, T(u_i, v_{k+1}))$ for $1 \leq i \leq \ell$. Therefore, by Lemma 1, it holds that $b(v_{k+1}, T(v_{k+1}, v_{k+1}')) = \max\{t(u_i) + w(u_i, v_{k+1}) + \alpha i \mid 1 \leq i \leq \ell\}$. Meanwhile, according to the execution of Step 7 of the algorithm BROADCAST, we have $t(v_{k+1}) = \max\{t(u_i) + w(u_i, v_{k+1}) + \alpha i \mid 1 \leq i \leq \ell\}$. It follows that $t(v_{k+1}) = b(v_{k+1}, T(v_{k+1}, v_{k+1}'))$.

Lemma 3. *If $b(x_1, T(x_1, x_2)) \leq b(x_2, T(x_2, x_1))$ with $\overline{x_1, x_2} \in E(T)$, then the following two statements hold :*

1. *$b(x_1, T) = \alpha + w(x_1, x_2) + b(x_2, T(x_2, x_1))$; and*
2. *$b(x_2, T) \leq b(x_1, T)$.*

In the following, we show that after removing a leaf u in the current tree T, $T - \{u\}$ still contains a broadcast center by keeping the vertex with the largest label. It leads to the fact that the last remaining vertex κ is a broadcast center.

Lemma 4. *Suppose that a leaf u is deleted in the current tree T in the ith iteration of the while loop. Then we have the following results:*

1. *$BC(T) \cap V(T') \neq \emptyset$, where $T' = T - \{u\}$; and*
2. *the last remaining vertex $\kappa \in BC(T)$ and $b(\kappa, T) = b(T)$.*

Proof. Suppose that the leaf u is deleted in the ith iteration and v is the vertex adjacent to u in the current tree T. To prove $BC(T) \cap V(T') \neq \emptyset$, it suffices to show that $b(v, T) \leq b(u, T)$. We first consider the case when the current tree T contains exactly two vertices u and v. Note that by the choice of u and Lemma 2, $b(v, T(v, u)) = t(v) \geq t(u) = b(u, T(u, v))$. According to Lemma 3, since $b(v, T(v, u)) \geq b(u, T(u, v))$ with $\overline{u, v} \in E(T)$, we have $b(v, T) \leq b(u, T)$.

Next, we consider the case when the current tree T contains at least three vertices u, v, and y. Suppose that u and y are the two leaves selected in the ith iteration of the while loop and $t(u) \leq t(y)$. Let y' be the neighbor of y on the path from y to v. Similarly, it suffices to show that $b(v, T) \leq b(u, T)$. Once again, we prove that by showing $b(v, T(v, u)) \geq b(u, T(u, v))$ according to Lemma 3. Suppose to the contrary that $b(v, T(v, u)) < b(u, T(u, v))$. Since $T(y, y')$ is a subtree of $T(v, u)$, we have $b(y, T(y, y')) < b(v, T(v, u))$. By $t(u) \leq t(y)$, we have $b(u, T(u, v)) = t(u) \leq t(y) = b(y, T(y, y'))$. This implies that $b(y, T(y, y')) < b(v, T(v, u)) < b(u, T(u, v)) \leq b(y, T(y, y'))$, a contradiction. Therefore, it is in fact that $b(v, T(v, u)) \geq b(u, T(u, v))$. Hence according to Lemma 3, since $b(v, T(v, u)) \geq b(u, T(u, v))$ with $\overline{u, v} \in E(T)$, we have $b(v, T) \leq b(u, T)$.

Intuitively, a tree may contain more than one broadcast center. Below we show that the only candidates for being a broadcast center of T are the neighbors of κ. Moreover, for each vertex $u_i \in N(\kappa)$, u_i is a broadcast center of T if and only if $w(u_i, \kappa) = 0$ and $i \leq h$.

Lemma 5. *If v is the broadcast center in T, then $v \in N(\kappa) \cup \{\kappa\}$.*

Lemma 6. *For each vertex $u_i \in N(\kappa)$, if $w(u_i, \kappa) = 0$ and $i \leq h$, then we have $u_i \in BC(T)$.*

Lemma 7. *For each vertex $u_i \in N(\kappa)$, if $w(u_i, \kappa) > 0$ or $i > h$, then we have $u_i \notin BC(T)$.*

Since the only candidates for being a broadcast center of T are the neighbors of κ, the set of broadcast centers $BC(T)$ is a star. Then we have the following corollary.

Corollary 1. *The set of broadcast centers $BC(T)$ is a star.*

Below we provide the correctness proof and the timing analysis of the algorithm.

Theorem 1. *Given a weighted tree $T(V, E)$, the algorithm BROADCAST computes the set of broadcast centers and determines the broadcasting time $b(T)$ in $O(n)$ time.*

Proof. Combining Lemmas 4 to 7, we obtain the correctness proof of the algorithm. In the following, we will show that the algorithm can be implemented in $O(n)$ time. We first observe that Steps 1 and 2 take $O(n)$ time. For the while loop, there is an intuitive implementation which takes $O(k \log k)$ time to sort the values $t(u_1) + w(u_1, v), t(u_2) + w(u_2, v), \ldots, t(u_k) + w(u_k, v)$ for each vertex v in T. However, we can see that one vertex is removed from T in each iteration of the while loop. Therefore, if Step 7 takes $O(k)$ time to assign the label $t(v)$ for each vertex v in T, then the while loop can also be completed in $O(n)$ time. In the following, we introduce a useful data structure which enables us to compute labels for n vertices in $O(n)$ time without sorting.

Suppose that $u_1, u_2, \ldots, u_k$ are the neighbors of v in T such that $t(u_1) + w(u_1, v) \geq t(u_2) + w(u_2, v) \geq \ldots \geq t(u_k) + w(u_k, v)$. Note that $k = |N(v)|$. If a vertex u_p satisfies $t(u_1) + w(u_1, v) \geq t(u_p) + w(u_p, v) + \alpha k$ with $k \geq p$, then $t(u_1) + w(u_1, v) + \alpha > t(u_p) + w(u_p, v) + \alpha p$. Note that the vertex u_p would have no influence on the label of v. Therefore, in order to determine $t(v)$, we only need to consider the neighbor u_i of v such that $t(u_i) + w(u_i, v) > t(u_1) + w(u_1, v) - \alpha k$. Based on the above observation, we construct the following k linked lists.

For each list$[i]$, with $0 \leq i \leq k - 1$, the list$[i]$ contains the vertices u_j such that $\alpha i \leq (t(u_1) + w(u_1, v)) - (t(u_j) + w(u_j, v)) < \alpha(i + 1)$. Note that the vertex u_1 with $t(u_1) + w(u_1, v) = \max\{t(u_i) + w(u_i, v)\}$ can be determined by simply visiting the neighbors of v once. For each neighbor u_i of v in $N(v) - \{u_1\}$, if $t(u_i) + w(u_i, v) \leq t(u_1) + w(u_1, v) - \alpha k$, then we discard it. Otherwise, we insert the vertex into the front of its corresponding linked list. Therefore, it takes $O(k)$ time to construct these k linked lists.

We use num$[i]$ to denote the number of vertices in the list$[i]$ and let acc$[i] = \sum_{j=0}^{i}$ num$[j]$ for $0 \leq i \leq k - 1$. Further, let u_{i^*} be the vertex in the list$[i]$ such that $t(u_{i^*}) + w(u_{i^*}, v) = \min\{t(u_j) + w(u_j, v) \mid u_j$ belongs to the list$[i]\}$. Clearly, for $0 \leq i \leq k - 1$, the values num$[i]$, acc$[i]$, and the vertex u_{i^*} can be determined in $O(k)$ time. For any given vertices u_x and u_y belonging to the same linked list with $x < y$, since $y - x \geq 1$ and $(t(u_x) + w(u_x, v)) - (t(u_y) + w(u_y, v)) < \alpha$, we have $t(u_y) + w(u_y, v) + \alpha y > t(u_x) + w(u_x, v) + \alpha x$. Therefore,

$$t(v) = \max\{t(u_i) + w(u_i, v) + \alpha i \mid 1 \leq i \leq k\}$$
$$= \max\{t(u_{i^*}) + w(u_{i^*}, v) + \alpha \text{acc}[i] \mid 0 \leq i \leq k - 1\}.$$

So the label of v, $t(v)$, can be determined in $O(k)$ time. By the above labeling method, the while loop can be completed in $O(n)$ time for n vertices.

Next, we show that the smallest integer h such that $t(u_h) + w(u_h, \kappa) + \alpha h + \alpha > b(T)$ can also be determined in $O(n)$ time in Step 12. Let q be the smallest integer such that $t(u_{q^*}) + w(u_{q^*}, \kappa) + \alpha \text{acc}[q] + \alpha > b(T)$. Clearly, the list$[q]$ contains the vertex u_h. We will show that either $h = \text{acc}[q]$ or $h = \text{acc}[q] - 1$. For any given vertices u_x and u_y belonging to the same linked list with $y \geq x + 2$, since

$(t(u_x)+w(u_x,v))-(t(u_y)+w(u_y,v)) < \alpha$, we have $(t(u_y)+w(u_y,\kappa)+y)-(t(u_x)+w(u_x,\kappa)+x) > \alpha$. Note that $t(u_{q^*}) + w(u_{q^*},\kappa) + \alpha\mathrm{acc}[q] \le b(T)$. Therefore, we have $h = \mathrm{acc}[q]$ or $h = \mathrm{acc}[q] - 1$. Hence, h can be determined in $O(n)$ time. These prove the correctness and the time complexity of the algorithm.

In the following, we will show that an optimal sequence of calls by which the broadcast center broadcasts its messages to all vertices in T can be determined in $O(n)$ time.

Theorem 2. *An optimal sequence of calls by which the broadcast center broadcasts its messages to all vertices in T can be determined in $O(n)$ time.*

Proof. Suppose that $u_1, u_2, \ldots, u_k$ are the neighbors of v in T such that $t(u_1) + w(u_1,v) \ge t(u_2)+w(u_2,v) \ge \ldots \ge t(u_k)+w(u_k,v)$. For each list$[i]$, with $0 \le i \le k-1$, the list$[i]$ contains the vertices u_j such that $\alpha i \le (t(u_1)+w(u_1,v))-(t(u_j)+w(u_j,v)) < \alpha(i+1)$. Suppose further that the list$[k]$ contains the vertices u_j such that $t(u_1) + w(u_1,v) \ge t(u_j) + w(u_j,v) + \alpha k$. Let $\mathrm{num}[i]$ denote the number of vertices in the list$[i]$ and $\mathrm{acc}[i] = \sum_{j=0}^{i} \mathrm{num}[j]$ for $0 \le i \le k-1$. Further, let u_{i^*} be the vertex in the list$[i]$ such that $t(u_{i^*}) + w(u_{i^*},v) = \min\{t(u_j) + w(u_j,v) \mid u_j$ belongs to the list$[i]\}$. We place the vertex u_{i^*} at the end of the list$[i]$ for $0 \le i \le k$.

Then, we assume that $u_{\pi(1)}, u_{\pi(2)}, \ldots, u_{\pi(k)}$ is a traversal ordering of the lists such that the list$[p]$ is traversed before the list$[q]$ if $p < q$. When traversing a list, we traverse the list from the beginning to the end of the list sequentially. Clearly, the ordering $u_{\pi(1)}, u_{\pi(2)}, \ldots, u_{\pi(k)}$ can be determined in $O(k)$ time. Since

$$t(v) = \max\{t(u_i) + w(u_i,v) + \alpha i \mid 1 \le i \le k\}$$
$$= \max\{t(u_{i^*}) + w(u_{i^*},v) + \alpha\mathrm{acc}[i] \mid 0 \le i \le k-1\}$$
$$= \max\{t(u_{\pi(i)}) + w(u_{\pi(i)},v) + \alpha\pi(i) \mid 1 \le i \le k\},$$

$u_{\pi(1)}, u_{\pi(2)}, \ldots, u_{\pi(k)}$ is an optimal sequence of calls by which v broadcasts messages to its neighboring vertices in T, which completes the proof.

Using a similar method of the above arguments, one can show that the following result is true.

Theorem 3. *Given a vertex $v \in V(T)$, the broadcasting time $b(v,T)$ and an optimal sequence of calls from v to all vertices in T can be determined in $O(n)$ time.*

4   Conclusion and Future Work

We have proposed a non-sorting linear time algorithm for the broadcasting problem in a weighted tree following the postal model. The algorithm BROADCAST computes the set of broadcast centers, determines the broadcasting time $b(T)$, and an optimal sequence of calls from the broadcast center to all vertices in T. Also, given a vertex $v \in V(T)$, the broadcasting time $b(v,T)$ and an optimal sequence of calls from v to all vertices in T can be determined in linear time.

Below we present some open problems related to the broadcasting problem in heterogeneous networks following the postal model, i.e., the broadcasting p-center and the broadcasting p-median problems.

The broadcasting p-center problem is a generalization of the broadcasting problem. In the broadcasting p-center problem, we want to locate p centers on a network and partition the set of n vertices in p subsets, such that the maximum communication time from the centers to the associated subsets of vertices is minimized. On the other hand, in the broadcasting p-median problem, instead of minimizing the maximum communication time, we minimize the sum of communication times from the centers to the associated subsets of vertices.

References

1. Bar-Noy, A., Guha, S., Naor, J., Schieber, B.: Message multicasting in heterogeneous networks. SIAM J. Comput. 30(2), 347–358 (2000)
2. Bar-Noy, A., Kipnis, S.: Designing broadcasting algorithms in the postal model for message-passing systems. Math. Systems Theory 27(5), 431–452 (1994)
3. Bar-Noy, A., Kipnis, S.: Multiple message broadcasting in the postal model. Networks 29(1), 1–10 (1997)
4. Elkin, M., Kortsarz, G.: An approximation algorithm for the directed telephone multicast problem. Algorithmica 45(4), 569–583 (2006)
5. Farley, A.M.: Broadcast time in communication networks. SIAM J. Appl. Math. 39(2), 385–390 (1980)
6. Hedetniemi, S.M., Hedetniemi, S.T., Liestman, A.L.: A survey of gossiping and broadcasting in communication networks. Networks 18(4), 319–349 (1988)
7. Khuller, S., Kim, Y.-A.: Broadcasting in heterogeneous networks. Algorithmica 48(1), 1–21 (2007)
8. Khuller, S., Kim, Y.-A., Wan, Y.-C.: On generalized gossiping and broadcasting. J. Algorithms 59(2), 81–106 (2006)
9. Khuller, S., Kim, Y.-A., Wan, Y.-C.: Broadcasting on networks of workstations. Algorithmica 52(3), 1–21 (2008)
10. Koh, J.-M., Tcha, D.-W.: Information dissemination in trees with nonuniform edge transmission times. IEEE Trans. Comput. 40(10), 1174–1177 (1991)
11. Lee, H.-M., Chang, G.J.: Set to set broadcasting in communication networks. Discrete Appl. Math. 40(3), 411–421 (1992)
12. Richards, D., Liestman, A.L.: Generalizations of broadcasting and gossiping. Networks 18(2), 125–138 (1988)
13. Slater, P.J., Cockayne, E.J., Hedetniemi, S.T.: Information dissemination in trees. SIAM J. Comput. 10(4), 692–701 (1981)