Contents

Set to Set Broadcasting in Heterogeneous Networks

Student: An-Chen Hsiao

Advisor: Der-Tsai Lee, Ph.D.

June, 2008

Abstract

In a *heterogeneous network* the *Broadcasting Problem*, which is to find the *Broadcasting Center* from which to broadcast messages to all the vertices in the network takes the least time, is an interesting problem. In this thesis, we will focus on heterogeneous tree networks under the telephone communication model. Given a tree $G = (V, E)$ in which each vertex $v \in V$ has a weight $W(v)$ and each edge $e(u, w) \in E$ has a weight $W(u, w)$, the weight $W(v)$ of a vertex v represents the amount of time (in units) needed for v to process the information before it can be passed on to other vertices u which are connected to v by an edge and the weight $W(u, w)$ of an edge $e(u, w)$ represents the amount (in units) of time needed to transmit messages between two end vertices of e. Under the telephone communication model in which each vertex can be engaged in the transmission of messages to exactly one adjacent vertex at a given time, we present an algorithm that solves the *Broadcasting Problem* in $O(n \log n)$ time, where n is the number of vertices in the tree. A similar

problem, called *Gathering Problem*, in which the messages of all the vertices are to be *gathered* at the *Gathering Center* in the least amount of time, is also investigated. An $O(n \log n)$ time algorithm for finding the *Gathering Center* in a tree network of n vertices is also presented. We also study the *Set to Set Broadcasting Problem*, where the messages in one set A of vertices are to be transmitted to all the vertices in another set B of vertices in the least amount of time under the same model. Based on the two algorithms obtained for the broadcasting problem and for the gathering problem, we present an $O(n \log n)$ time algorithm to solve the *Set to Set Broadcasting Problem* in a tree network.

1 Introduction

In a heterogeneous communication network, messages are transmitted between vertices that are connected by an edge. The transmission takes time which depends on the bandwidth available and communication delay between the two end vertices. If we use point-to-point transmission to transmit messages when we want to do broadcasting, i.e., the sender wishes to send messages to all the vertices, the same message may get transmitted along an edge more than once. This point-to-point communication method is in fact very inefficient, in terms of bandwidth usage and transmission time. A point-to-point transmission between two end vertices requires that we find a transmission path connecting these two points. Broadcasting from the sender, called the *source*, denoted s, to all the other vertices, $v_1,\ v_2,\ \ldots,\ v_{n-1}$, in the communication network, will require establishment of $n-1$ transmission paths from s to v_i, $i = 1,\ 2,\ \ldots,\ n-1$. If we ignore the fact that some of edges in the network may belong to more than one path, the *union* of these $n-1$ transmission paths will result in a *tree*, rooted at the source s. In fact, this tree is a *shortest path tree*, so that for each node v_i, the path from the root s to v_i is the *shortest* in the sense that the point-to-point transmission time between them is the least. This *shortest path tree* or any spanning tree of the heterogeneous communication network is also referred to as a *broadcasting tree*. Given a connected network, there exists more than one broadcasting tree. The broadcasting time of a broadcast tree is defined to be the duration of communication time when the last vertex receives the message sent from the source. Depending on the model of communication and transmission cost along each edge, we will obtain different broadcasting tree in order to optimize the broadcasting time. Multicasting from the sender, or source, denoted

s, is on the other hand to send messages from the source to a subset of vertices in the communication network. In other words, broadcasting is a special case of multicasting in which the destinations are the entire set of vertices. The broadcasting and multicasting are important communication primitives and often used in many applications for message-passing systems (see [3]).

Consider a connected communication network, modeled as a graph, $G = (V, E)$ in which each vertex $v \in V$ has a weight $W(v)$ and each edge $e(u, w) \in E$ has a weight $W(u, w)$. The weight $W(v)$ of a vertex v represents the amount of time (in units) needed for v to process the information before it can be passed on to other vertices u which are connected to v by an edge and the weight $W(u, w)$ of an edge $e(u, w)$ represents the amount (in units) of time needed to transmit messages between two end vertices of e. From here on we shall use *vertex* and *node* interchangeably. We shall consider the famous *telephone model*, proposed by [2], as our model of communication in this thesis. In this model, a node sends messages to at most one adjacent node at a time, called a *round*. The character of the telephone model is that the sender and receiver are busy when they are engaged in the transmission of messages. Only after the receiver gets the messages completely can both end vertices be able to communicate with other nodes. Transmissions of messages along two edges can take place simultaneously as long as there are no common end vertices. In our heterogeneous network model it is further assumed that the receiver node v, upon receipt of the messages for the first time may take $W(v)$ units of time processing the messages before it can start transmitting the messages to other nodes. The transmission time $W(u, w)$ along an edge $e(u, w)$ is assumed to be constant. There are other models of communication, such as the postal model[1] and the LogP model[5]. Given a source v in a connected graph G, and a spanning tree T of G, the broadcasting time $b(v, T)$ is defined to be the *minimum* amount of time needed to have the messages sent to *all* the nodes. Given a source v, the *Broadcasting Problem* in G is to find a spanning tree T such that $b(v, T)$ is minimized. The vertex, v, for which $b(v, T)$ is minimum among all possible spanning trees T, is called the *Broadcasting Center*. The *Broadcasting Problem* in G is to find such a broadcasting center v, a spanning tree in G and a specification of the *rounds* or the *sequence of calls*.

It has been known that finding an optimal broadcasting center under the telephone communication model for an arbitrary graph is NP-hard [4]. It is NP-hard even for 3-regular planar graphs [9]. There are some known approximation algorithms for this problem. Ravi [11] gave an algorithm for finding the minimum multicasting time required to multicast messages from one source to a subset of size k of the destination nodes with an approximation factor of $O(\frac{\log n \log k}{\log \log k})$ under the basic telephone model in which a node may send a message to at most one other node in each round.

We shall consider a simpler case where the communication network is a tree, denoted $T = (V, E)$ As mentioned earlier, telephone calls along two disjoint edges can take place simultaneously. Let $b(v, T)$ denote the minimum time in which node v can send messages to every other node in T. The broadcasting time in tree T, denoted $b(T)$, is defined as $b(T) = \min\{b(v, T) | v \in V(T)\}$. The *Broadcasting Center* of T, denoted $BC(T)$, is the set of all vertices having minimum broadcasting time, i.e., $BC(T) = \{v | b(v, T) = b(T)\}$. Slater et al. [8] gave a linear-time algorithm to find the minimum broadcasting time and broadcasting center of an arbitrary tree

T in which no processing time $W(v)$ at each node v and no transmission delay $W(u, v)$ along each link $e(u, v)$ were considered. Slater et al. [8] also proved that this problem in general graphs is NP-complete.

Given a tree $T = (V, E)$, the *Gathering Problem* is to have every vertex in tree T pass messages to a destination vertex v. Let $g(v, T)$ denote the minimum time in which vertex v can receive messages from every other node in T. The gathering time in tree T, denoted $g(T)$, is defined as $g(T) = \min\{g(v, T)|v \in V(T)\}$. A node v for which the gathering time $g(v, T) = g(T)$ is called a *Gathering Center*. We find that the minimum gathering time of tree T is not always equal to the minimum broadcasting time. But we find the gathering center problem is similar to the broadcasting center problem. We can use similar idea to solve this problem.

Given a graph $G = (V, E)$, each vertex has a unique message and is ignorant of the messages of other vertices at the beginning. The *Gossiping Problem* is to determine the minimum time for all vertices to know all messages under the basic telephone model. Let A and B be two nonempty vertex sets in a connected graph $G = (V, E)$. Each vertex in A has a unique message. By a sequence of calls, all vertices in B are supposed to get all the messages from A. We denote by $t(A, B, G)$ the minimum time that vertex set A broadcasts all its messages to vertex set B. The *Set to Set Broadcasting Problem* is a generalization of the Gossiping Problem, in which $A = B = V$. Richards and Liestman [7], and Lee and Chang [6] proposed an idea to solve the set to set broadcasting problem in linear time under the telephone model.

In Chapter 2 of this thesis, we consider the *Broadcasting Problem* on a tree under the telephone model. Given a tree $T = (V, E)$ with non-negative integer vertex weight $W(v)$ for all $v \in V$, we present an algorithm for determining the broadcasting center and minimum broadcasting time with an optimal sequence of calls in a heterogeneous model. In Chapter 3, we provide an algorithm for determining the gathering center and minimum gathering time with an optimal sequence of calls in the same model of Chapter 2. In Chapter 4, we give an algorithm to compute the minimum broadcasting time $t(A, B, T)$ for the set to set broadcasting problem with an optimal sequence of calls in the same model of Chapter 2. We make use of the results obtained in Chapter 2 and Chapter 3 to solve this problem. In Chapter 5, we consider the case that the edge weights can be non-negative integers as a contrast to the model of Chapter 2. And we provide an algorithm for determining the broadcasting center and minimum broadcasting time with an optimal sequence of calls. The algorithms in Chapters 2, 3, 4 and 5 all have time complexity $O(n \log n)$,where n is the size of the heterogeneous network.

2 Broadcasting Center for a Tree with Vertex Weight

In this chapter, we will provide an algorithm to calculate the minimum broadcasting time and find the broadcasting center of a heterogeneous tree network under the telephone model. We will get an optimal sequence of telephone calls in $O(n \log n)$ time, where n is the number of nodes in the network.

We recall our communication model in the following. Given a connected graph $G = (V, E)$ with each vertex $v_i \in V$ having weight $W(v_i)$, where the weight is a non-negative integer. Transmitting messages in G has the following constraints:

1. A vertex can only send messages to its adjacent vertex, and receiver can receive messages immediately when the sender is ready to transmit.

2. A vertex can only communicate (send or receive) with one other vertex at a time.

3. After receiving a message, a vertex v needs $W(v)$ units of time to process that message. Only after the processing can it send the message to other vertices.

4. When finishing the processing of received messages or a transmission of messages to an adjacent vertex, a vertex can send messages to another vertex at the next unit of time. We call it a *switching delay*.

We call the model "Heterogeneous Networks" and this model differs only from basic communication networks in that we have weights associated with vertices. In this thesis we assume graph G is a tree $T = (V, E)$, and we call it a heterogeneous tree.

2.1 Problem Definition and the Algorithm

In this section, we will present an algorithm for finding the broadcasting center of a heterogeneous tree with vertex weights. In 1981 Slater et al. [8] provided some good properties of broadcasting problem for trees which are useful in heterogeneous trees.

We define a few terminology and introduce some notation below. For an edge $e(v, u)$ in a tree 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 that containing u is denoted $T(u, v)$, as shown in Figure 1. We define $b(v, G)$ to be the broadcasting time from v to all vertices in graph G. Therefore, our goal is to find a vertex v such that $b(v, T)$ is minimum. This vertex is called *broadcasting center* of T. Note that the center may not be unique. A heterogeneous tree may have more than one broadcasting center. All the broadcasting centers of a tree T, written as $BC(T)$, will form a *star*, which is a tree itself that all vertices but one are leaves.

Our algorithm will adopt a greedy strategy and process the nodes in a bottom up manner. The key idea is described below. Since a broadcasting center has minimum broadcasting time, using bottom-up method is a very reasonable idea. Suppose a vertex v has neighbors $u_1,\ u_2,\ \ldots,\ u_k$ in T. Without loss of generality we suppose that $b(u_1, T(u_1, v)) \geq b(u_2, T(u_2, v)) \geq \ldots \geq b(u_k, T(u_k, v))$. If v needs to broadcast messages, we will send messages to u_i from 1 to k in that order. Since u_1 has the largest transmission time to $T(u_1, v)$, sending messages from v to it at first is the best strategy. If v would send messages at any other time, the broadcasting time won't be better, since $b(v, T(v, u)) > b(u_1, T(u_1, v)) + 1$. This is what we call *message broadcasting sequence protocol*, i.e., v will make the first *telephone call* to u_1 and then to $u_2,\ u_3,\ \ldots,\ u_k$ subsequently in that order. The broadcasting time from v will be $b(v) = \max\{b(u_i, T(u_i, v)) + i | 1 \leq i \leq k\}$.

The **Algorithm Broadcasting** uses $t(u)$ to label every vertex u in tree T. Let v be a broadcasting center of tree T and u a vertex other than v. Let vertex u' be the vertex on the path from u to v that is adjacent to u. The **Algorithm Broadcasting** computes $t(u) = b(u, T(u, u'))$, which is equal to the minimum broadcasting time from u to every vertex in tree $T(u, u')$. Finally we can get that $t(v) = b(v, T)$ where v is a broadcasting center. We use bottom-up method to label vertices in a certain order to be described below. At first, we label every leaf $\ell \neq v$ with 0, i.e., $t(\ell) = 0$. We will process the vertex in non-decreasing order of its associated label. Each time a leaf node is selected for processing, it will be *removed* from the current tree T,

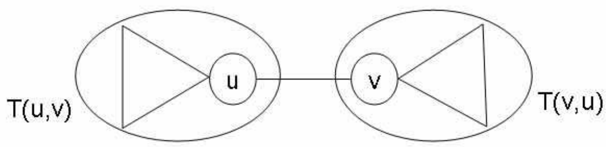

Figure 1: Subtrees of T as a result of the removal of edge (u, v) are T(u,v) and T(v,u)

which is being updated as some of its leaves are processed, and the removed nodes will be added to another set, denoted U, of *processed* nodes, as described below. If the selected node w is the last leaf of another node v, i.e., its removal will make its adjacent node v a new leaf node, we will begin to compute the label $t(v)$ of node v according to the *messages broadcasting sequence protocol* as described earlier. In fact, this is how the label of a non-leaf node of T gets computed before it becomes a leaf node in the subsequent iteration. As we recall, all the leaf nodes of T get the label 0 initially. We will *always* select leaf nodes for processing and move them from the current tree T to a newly created set U as described in the algorithm below. Let T' denote the tree T being processed as its leaves are being removed. $T' = T$ initially. Let $\mathcal{L}$ denote the set of leaves of T'.

Algorithm Broadcasting

$U \leftarrow \emptyset$;

$T' \leftarrow T$;

For each vertex $u \in \mathcal{L}$ **do** $t(u) \leftarrow 0$;

while $|V(T')| > 2$ **do**

let w be a leaf of T' and $t(w) = \min\{t(w_i) : w_i \in \mathcal{L}\}$;

let v be the vertex adjacent to w in T';

$T' \leftarrow T' - \{w\}$;

$U \leftarrow U \cup \{w\}$;

if v is a leaf of T'

then [let v be adjacent to some vertices which are in U. Suppose that

they are $u_1,\ u_2,\ \ldots,\ u_k$ and $t(u_1) \geq t(u_2) \geq \ldots \geq t(u_k)$;

$t(v) \leftarrow \max\{t(u_i) + i | 1 \leq i \leq k\} + W(v)$;]

end;

Let u and v be the two remaining vertices in T' and $W(v) + t(u) \leq W(u) + t(v)$;

$T' \leftarrow T' - \{u\}$;

$U \leftarrow U \cup \{u\}$;

Let the neighbors of v in T be $u_1,\ u_2,\ \ldots,\ u_k$ and $t(u_1) \geq t(u_2) \geq \ldots, \geq t(u_k)$;

Let j be the *smallest* integer such that $t(u_j) + j = \max\{t(u_i) + i | 1 \leq i \leq k\}$;

$b(T) \leftarrow W(v) + t(u_j) + j$;

if $j > 1$

if $W(u_i) = 0$ for $1 \leq i \leq j$

$BC(T) \leftarrow \{v, u_i\}$;

else

if $W(v) + t(u_1) = W(u_1) + W(v) + \max\{t(u_i) + (i-1) | 2 \leq i \leq k\}$

$BC(T) \leftarrow \{v, u_1\}$;

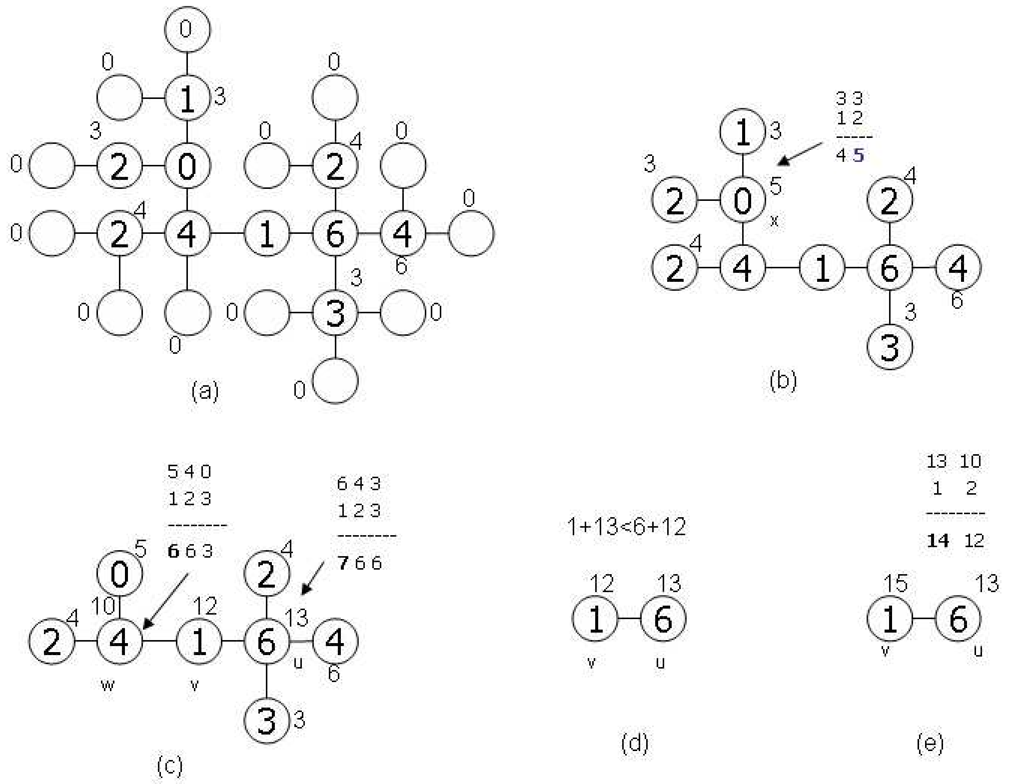

Figure 2: Algorithm **Broadcasting** applied to tree T

Figure 2 illustrates the **Algorithm Broadcasting** for tree T. In Figure 2(a) each leaf is labeled 0. Some vertices are labeled by **Algorithm Broadcasting**. In Figure 2(b) we see that sending messages from node x to its children needs 5 units of time and we need $W(x)$ units of time for processing. Thus, $t(x) = 5 + 0 = 5$. In Figure 2(c) we compute $t(w)$ and $t(u)$. There is only one vertex w in U, when we compute $t(v)$. So $t(v)$ is 12. In Figure 2(d) there are two vertices v and u remaining. We find $W(v) + t(u) \le W(u) + t(v)$. By **Algorithm Broadcasting**, vertex v is the remaining vertex and v will be the broadcasting center of T. In Figure 2(e) we compute that $t(v)$ is 15 and it is the broadcasting time of T. By **Algorithm Broadcasting** we know that the broadcasting center of the tree shown in Figure 2(a) consists of just one vertex v.

2.2 Proof of Correctness

In this section we will prove the correctness of **Algorithm Broadcasting**.

Theorem 2.1 *Suppose that* $u_1,\ u_2,\ \ldots,\ u_k$ *are the neighbors of vertex* v *in tree* T*. Without loss of generality, we suppose* $b(u_1, T(u_1, v)) \geq b(u_2, T(u_2, v)) \geq \ldots \geq b(u_k, T(u_k, v))$*. Then*

$$b(v, T) = max\{b(u_i, T(u_i, v)) + i\}\ + W(v),\ for\ 1 \leq i \leq k$$

Proof: Suppose vertex v passes information to u_i at time $\pi(i)$, $i = 1,\ 2,\ \ldots,\ k$, where π is a permutation of $\{1,\ 2,\ \ldots,\ k\}$. Thus vertices in subtree $T(u_i, v)$ will get the messages after $b(u_i, T(u_i, v)) + \pi(i)$ units of time. Therefore, we derive that

$$b(v, T) = min_{\pi}\{max_i\{b(u_i, T(u_i, v)) + \pi(i)\} + W(v)\}.$$

We claim that the permutation $\pi(i) = i$ yields an optimal calling sequence. Let us assume that $\pi'(i) \neq \pi(i)$ for some $i \in \{1,\ 2,\ \ldots,\ k\}$ yields a solution $b'(v, T)$ which is strictly less than $b(v, T)$ obtained with π. Let j be the smallest index such that $\pi'(j) \neq \pi(j)$. Without loss of generality, we further assume that among all permutations which yield a better broadcasting time than π, π' is the one with the largest j. Let us assume $\pi'(j) = \ell > \pi(j) = j$. In other words, $\pi' = (1,\ 2,\ \ldots,\ j-1,\ \ell,\ \ldots,\ j,\ \ldots)$ where the suffix $(\ell,\ \ldots,\ j,\ \ldots)$ is a permutation of $\{j,\ j+1,\ \ldots,\ k\}$. Let us construct a new permutation π'' from π' by exchanging ℓ and j. That is, $\pi'' = (1,\ 2,\ \ldots,\ j-1,\ j,\ \ldots,\ \ell,\ \ldots)$. Since $b(u_j, T(u_j, v)) + \ell > b(u_j, T(u_j, v)) + j$, it is not difficult to see that the broadcasting time $b''(v, T)$ associated with π'' is no greater than $b'(v, T)$. In other words, permutation π'' would also yield a broadcasting time better than π does. This contradicts the assumption that permutation π' is the one with the largest j. Therefore, the permutation $\pi(i) = i$ would minimize $b(v, T)$. According to our model, we know vertex v needs $W(v)$ units of time to process information, and then passes the information to its neighbors. Therefore, we have $b(v, T) = W(v) + max\{b(u_i, T(u_i, v)) + i\}$, for $1 \leq i \leq k$. □

Let v be the vertex obtained by **Algorithm Broadcasting**. We now prove that the vertex v by definition is the broadcasting center of tree T.

Lemma 2.1 *Let u be any vertex other than the broadcasting center v, and let u' be the first vertex (adjacent to u) on the path from u to v. We have $t(u) = b(u, T(u, u'))$.*

Proof: If vertex u is leaf, we can get that $t(u) = 0 = b(u, T(u, u'))$. We use bottom-up method to compute the labels of the vertices. Let us assume inductively that u is a non-leaf vertex such that $t(u) = b(u, T(u, u'))$ is true, where u' is the *parent* of u. We consider the vertex u'. Let the parent of u' be r. We shall show $t(u') = b(u', T(u', r))$. The label $t(u')$ of vertex u' is calculated when all the labels of its children $w_1,\ w_2,\ \ldots,\ w_j$ for some $j \geq 1$, but one, say w_j, have been computed, and vertex w_j is the current vertex whose label $t(w_j)$ is the smallest among all those labels associated with the leaf nodes of the current tree. In other words, we have $t(w_1) \leq t(w_2) \leq \ldots \leq t(w_j)$. According to the *messages broadcasting sequence protocol* we compute $t(u')$ as the maximum of $\{t(w_i)+j-i+1 | 1 \leq i \leq j\}$ plus $W(u')$. By Theorem 2.1 and the definition of broadcasting time, we have $t(u') = b(u', T(u', r))$. This completes the proof. □

Lemma 2.2 *Assume that u and v are adjacent vertices in tree T and have vertex weight $W(u)$ and $W(v)$ respectively. Let $b(u,T(u,v))+W(v) \leq b(v,T(v,u))+W(u)$.*

1. $b(u,T) = W(u) + 1 + b(v,T(v,u))$

2. $b(v,T(v,u)) \leq b(v,T) \leq max\{b(v,T(v,u)) + 1, W(v) + 1 + b(u,T(u,v))\}$

Proof:

1. If vertex u is the source of tree T, it has two ways to pass the information.

   Sending messages to $T(v,u)$ first, and then to $T(u,v)$. The vertex v will get the messages at time $W(u) + 1$, and it takes $W(u) + 1 + b(v,T(v,u))$ units of time to finish broadcasting messages to all vertices in T(v,u). In this situation, vertex u will send messages to its neighbors in $T(u,v)$ at time $W(u) + 2$. Thus, vertices in $T(u,v)$ will receive the broadcast messages in $b(u,T(u,v)) + 1$ units of time. By assumption since $W(u) + 1 + b(v,T(v,u)) \geq b(u,T(u,v))+1$, we have $b(u,T) = W(u)+1+b(v,T(v,u))$.

   Sending messages to $T(u,v)$ first, and then to $T(v,u)$. In this situation, the best strategy is to send messages to vertex v as soon as possible. Thus vertex u will take additional one unit time to transmit messages to vertex v during the transmission time when u transmits messages to its neighbors in $T(u,v)$. Thus, finishing broadcasting messages to all vertices in T(u,v) will take at most $b(u,T(u,v)) + 1$ units of time. Since vertex u transmits messages to its neighbors in $T(u,v)$ first, transmission of messages from vertex u to vertex v will delay at least one unit time. Thus, the vertex v will get the messages in at least at time $W(u) + 1 + 1$. In the other words the subtree $T(v,u)$ will be done broadcasting in at least $W(u) + 2 + b(v, T(v, u)$ units of time. Therefore $b(u, T) \geq W(u) + 2 + b(v, T(v, u))$.

Following the arguments above, we have $b(u, T) = b(v, T(v, u)) + 1 + W(u)$.

2. $T(v, u)$ is a subtree of tree T, and therefore $b(v, T(v, u)) \leq b(v, T)$. In our model, the vertex v has two ways to send messages. One way is to send messages to vertices in $T(v, u)$ first, and it takes $b(v, T(v, u))$ units of time. Another is to send messages to vertex u first, and then u sends messages to vertices in $T(u, v)$. It takes $b(u, T(u, v)) + 1 + W(v)$ units of time. So we select the maximum between $b(v, T(v, u))$ and $b(u, T(u, v)) + 1 + W(v)$.

Case 1: $b(v, T(v, u)) \geq W(v) + 1 + b(u, T(u, v))$**.** In this situation, the best strategy is to send messages to vertex u as soon as possible. By the main idea of Theorem 2.1, vertex v will take additional one unit time to transmit messages to vertex u during the transmission when v transmit messages to its neighbors in $T(v, u)$. Thus, the subtree $T(v, u)$ will take at most $b(v, T(v, u)) + 1$ units of time. The vertex u will get the messages in at least $W(v) + 1 + 1$ units of time. Thus, T(u,v) will take $W(v) + 1 + b(u, T(u, v)) + 1$ and $b(v, T(v, u)) + 1 \geq W(v) + 1 + b(u, T(u, v)) + 1$. Therefore $b(v, T) \leq b(v, T(v, u)) + 1$.

Case 2: $b(v, T(v, u)) < W(v) + 1 + b(u, T(u, v))$**.** In this case, vertex v must send messages to vertex u first. Thus, it will take $W(v) + 1 + b(u, T(u, v))$ units of time to finish broadcasting of subtree $T(u, v)$. Vertex v sends messages to u at the beginning, which means that the children of vertex v in $T(v, u)$ will have a delay of one unit of time to get messages. Thus, finishing broadcasting of subtree $T(v,u)$ will take at least $b(v,T(v,u))+1$ units of time. Since $b(v,T(v,u)) < W(v)+1+b(u,T(u,v))$, $b(v,T) = \max\{b(v,T(v,u))+1,\ W(v)+1+b(u,T(u,v))\}$. Therefore $b(v,T) = W(v)+1+b(u,T(u,v))$.

We therefore conclude that, $b(u,T) \leq max\{b(v,T(v,u))+1,\ W(v)+1+b(u,T(u,v))\}$. □

This lemma illustrates some properties of the **Algorithm Broadcasting** when the conditions of Lemma 2.2 hold. The following lemma is useful.

Lemma 2.3 *If $w_1,\ w_2,\ \ldots,\ w_{n-1}$ is the sequence of vertices selected by **Algorithm Broadcasting**, and vertex w_n is the one remaining. Then $t(w_1) \leq t(w_2) \leq \ldots \leq t(w_n)$.*

Proof: Suppose $1 \leq k < h \leq n$. It suffices to show that $t(w_k) \leq t(w_h)$. Suppose vertex w_k is chosen during the k^{th} iteration of **Algorithm Broadcasting**. If w_h has received the label $t(w_h)$ during one of the first k-1 iterations, then $t(w_k) \leq t(w_h)$ by definition of how w_k is selected by **Algorithm Broadcasting**. If w_h has received label$t(w_h)$ during the j^{th} iteration, where $j \geq k$, then by Lemma 2.1 and Theorem 2.1 we have $t(w_h) > t(w_j)$. This completes the proof. □

Lemma 2.4 *Let vertices v and u be adjacent vertices in T', and u be selected by **Algorithm Broadcasting** in the while loop when $|V(T') > 2|$. Then $b(v,T) \leq b(u,T)$*

Proof: Suppose that there is a leaf $x \neq u$ and $x \neq v$ at the time when vertex u is selected. Let x' be the vertex in T' adjacent to x. By Lemma 2.3 $b(x, T(x, x')) \geq b(u, T(u, v))$, and we know that $T(x, x')$ is a subtree of $T(v, u)$. It implies that $b(v, T(v, u)) \geq W(v)+1+b(x, T(x, x'))$, which is no less than $b(u, T(u, v))+1+W(v)$, and so $b(v, T(v, u)) + W(u) > b(u, T(u, v)) + W(v)$.

By Lemma 2.2 we have $b(u, T) = W(u)+1+b(v, T(v, u))$ and $b(v, T) \leq b(v, T(v, u))+1$. Therefore $b(u, T) \geq b(v, T)$. □

Supposeu and v are the two remaining vertices after the while loop in **Algorithm Broadcasting** and $t(u) \leq t(v)$, i.e., $b(u, T(u, v)) \leq b(v, T(v, u))$. If we follow the same selection idea of **Algorithm Broadcasting**, we would select vertex u to delete and vertex v is the broadcasting center of T. But if $W(u) + b(v, T(v, u))$ is smaller than $W(v) + b(u, T(u, v))$, then u should have been the broadcasting center. Thus, we need an additional step to compare $W(u) + b(v, T(v, u))$ and $W(v) + b(u, T(u, v))$ to decide between u and v which is the broadcasting center of T. If $W(v) + b(u, T(u, v) \leq W(u) + b(v, T(v, u))$, then by Lemma 2.2 we have $b(v, T) \leq b(u, T)$ and thus the vertex v is the broadcasting center. Conversely, the vertex u is the broadcasting center.

Lemma 2.5 *Let u and v be adjacent vertices in tree T with weights $W(u)$ and $W(v)$ respectively. Without loss of generality, we suppose that $b(u, T(u, v)) + W(v) \leq b(v, (T(v, u)) + W(u)$. For each vertex s in $T(u, v) \setminus \{u\}$, $b(s, T) > b(v, T)$*

Proof: By Lemma 2.2 we have two cases: one is $b(u, T(u, v)) + 1 + W(v) \leq b(v, T(v, u))$, and the other is $b(v, T(v, u)) < b(u, T(u, v)) + 1 + W(v)$. At first we know that $b(s, T) \geq W(s) + 1 + W(u) + 1 + b(v, T(v, u))$.

Case 1: $b(u, T(u, v)) + 1 + W(v) \leq b(v, T(v, u))$**.** Then by Lemma 2.2 $b(v, T) \leq b(v, T(v, u)) + 1$. Therefore we have $b(s, T) > b(v, T)$.

Case 2: $b(u, T(u, v)) + 1 + W(v) > b(v, T(v, u))$**.** Thus by Lemma 2.2 we have $b(v, T) = W(v) + 1 + b(u, T(u, v))$. Since $b(u, T(u, v)) + W(v) \leq b(v, (T(v, u)) + W(u)$, we have $b(s, T) \geq W(s) + 1 + W(u) + 1 + b(v, T(v, u)) \geq W(s) + 1 + W(v) + 1 + b(u, T(u, v)) > W(v) + 1 + b(u, T(u, v)) = b(v, T)$. □

Lemma 2.6 *Let x and v be the two remaining vertices in **Algorithm Broadcasting** after the while loop and $W(v) + b(x, T(x, v)) \leq W(x) + b(v, T(v, x))$. If there exists some vertex u such that $b(u, T)$ is equal to $b(v, T)$, then u and v must be adjacent.*

Proof: Suppose that vertex s is not adjacent to v in tree T, and vertex u is adjacent to v in the path from s to v. We shall prove that if s and v are not adjacent, then $b(s, T) > b(v, T)$.

By Lemma 2.3, we know $b(u, T(u, v)) \leq b(v, T(v, u))$.

Then by Lemma 2.4 we have $b(u, T(u, v)) + W(v) \leq b(v, T(v, u)) + W(u)$.

This shows that the constraint of Lemma 2.5 holds and vertex s is in subtree $T(u, v) \setminus \{u\}$.

By Lemma 2.5 we have $b(s, T) > b(v, T)$. □

This lemma shows that if the Broadcasting Center of tree T is not unique, then it must be a star. Thus, we only need to check the vertices adjacent to the remaining vertex v to find out if any of them is also a Broadcast Center. The following theorem shows that the method of finding the broadcasting center is correct.

Theorem 2.2 *Suppose v is the last selected vertex after the while loop in **Algorithm Broadcasting** and let $u_1,\ u_2,\ \ldots,\ u_k$ be the neighbors of v. Without loss of generality we let $b(u_1, T(u_1, v)) \geq b(u_2, T(u_2, v)) \geq \ldots \geq b(u_k, T(u_k, v))$. Let j be the smallest integer such that*

$$t(u_j) + j = max\ \{t(u_i) + i\},\ for\ 1 \leq i \leq k$$

1. *If $j = 1$, and $W(v) + b(u_1, T(u_1, v)) = W(u_1) + b(v, T(v, u_1))$, then u_1 is also a broadcasting center.*

2. *If $j \geq 2$, and $W(u_i) = 0$, u_i is also a broadcasting center, for $i \leq j$.*

Proof: We know the broadcasting time of tree T is $b(v, T) = W(v) + t(u_j) + j$. We shall first prove that the vertex u_i, for $j < i \leq k$, cannot be a broadcasting center of tree T. Suppose we choose a vertex u_i, for $j < i \leq k$, to be a new source of tree T. The vertex v will receive messages from u_i at $W(u_i) + 1$ units of time and then pass messages to other vertices in subtree $T(v, u_i)$. It will take $W(u_i)+1+W(v)+t(u_j)+j$ units of time. It is larger than $b(v, T)$. Thus, the vertex u_i, for $j < i \leq k$, cannot be a broadcasting center of tree T.

Next we will prove the method of choosing $BC(T)$ in **Algorithm Broadcasting** is correct.

1. For $j = 1$, if $W(v) + b(u_1, T(u_1, v)) = W(u_1) + b(v, T(v, u_1)$.

   By Lemma 2.2 $\quad b(u_1, T) = W(u_1)+1+b(v, T(v, u_1)) = W(v)+1+b(u_1, T(u_1, v)) = b(v, T)$

2. For $j > 2$, if $W(u_i) = 0$, for $1 \leq i \leq j$ then u_i is also a broadcasting center of tree T.

Note that vertex v passes messages to its neighbors in the following order $u_1,\ u_2,\ \ldots,\ u_k$. Suppose that we choose a vertex u_h, for $1 \leq h \leq j$, to be a new source vertex. Vertex u_h will transmit messages to vertex v first. Then the vertex v passes the messages to its other neighbors in the following order $u_1,\ u_2,\ \ldots,\ u_{h-1},\ u_{h+1},\ u_{h+2},\ \ldots,\ u_k$. Therefore, all the vertices in T_i, for $1 \leq i \leq h-1 < j$, may be called at time $W(u_h)+1+W(v)+t(u_i)+i$. If $W(u_h)=0$, $W(v)+t(u_i)+i+1 \leq W(v)+t(u_j)+j$. All vertices in T_i, for $h+1 \leq i \leq k$, may be called at time $W(u_h)+1+W(v)+t(u_i)+(i-1)$. If $W(u_h)=0$, $W(u_h)+1+W(v)+t(u_i)+(i-1) = W(v)+t(u_i)+i$ which is no greater than $W(v)+t(u_j)+j$ by definition of j. All vertices in $T(u_h,v)$ may be called at time $t(u_h)+1 \leq t(u_h)+(h-1) \leq W(v)+t(u_j)+j$. Thus u_h, for $1 \leq h \leq j$, can also be a broadcasting center if $W(u_h)=0$. □

Property 2.1 *Let x be a vertex in tree T and x is not in $BC(T)$. There exists a path with the smallest distance from x to a vertex $v \in BC(T)$ containing vertex u, where u is adjacent v such that $b(x,T) = k + b(u,T)$, where k is the distance from x to u.*

Proof: That vertex x transmits messages to vertex u and then broadcasts to every vertex in tree T is a solution of $b(x,T)$. It is clearly that $b(x,T) \leq k + b(u,T)$.

Let us compute $b(u,T)$. If $b(u,T(u,v)) \geq W(u)+1+b(v,T(v,u))$, then we have $b(u,T(u,v))+W(v) > W(u)+b(v,T(v,u))$. And by Lemma 2.2 we have $b(u,T) \leq b(v,T)$. It shows that u is also broadcasting center, which is a contradiction. Thus, $W(u)+b(v,T(v,u)) \geq b(u,T(u,v))+W(v)$ so we have $b(u,T) = W(u)+1+b(v,T(v,u))$. It implies that $b(x,T) \geq k+W(u)+1+b(v,T(v,u)) = k+b(u,T)$. □

Lemma 2.7 *The time complexity of* ***Algorithm Broadcasting*** *for solving the broadcasting problem in heterogeneous tree networks for is* $O(n \log n)$.

Proof: Our algorithm has a main loop iterated n times. In each iteration, we remove a vertex with the smallest label from tree T'. We use a priority queue to implement it, so it needs $O(\log n)$ in each iteration and $O(n \log n)$ time totally. Other operations are all constant time operations, and take totally linear time. Outside the loop, except constant time operations we have a small loop iterated m times, where m is the number of nodes adjacent to v with labels larger than the maximum smallest integer j which $t(u_j) + j =$ max $\{t(u_i) + i\}$, for $1 \leq i \leq k$. It take totally $O(m)$ time. Since $m < n$, the total time complexity of the algorithm is $O(n \log n)$. □

Finally to obtain an optimal sequence of calls, we use the labels $t(v)$ of each vertex v in tree T. By Lemma 2.1 we know that $t(v)$ is equal to the minimum broadcasting time of vertex v passing messages to its children. By Theorem 2.1 we know the optimal strategy of broadcasting in heterogeneous trees. Each vertex having finished the processing, will pass messages to its children in nonascending order of their labels.

3 Gathering Center for a Tree with Vertex Weight

In this chapter we will provide an algorithm to calculate the minimum gathering time and find the gathering center of a heterogeneous tree network in the telephone model. In our heterogeneous tree communication model the reception and transmission operations are asymmetric. Therefore, the time needed to gather information from all points in tree T may not be equal to the time to broadcast information. Nevertheless this problem is similar to the broadcasting problem and many similar properties hold. So we can use the same ideas to solve this problem.

3.1 Problem Definition and Algorithm

In this section, we will show an algorithm for finding the gathering center of a heterogeneous tree with vertex weights. This problem is similar to the broadcasting problem studied in Chapter 2. Many good properties of the broadcasting problem for tree are useful for this problem as well.

We define a few terminology and introduce some notation below. Let $g(v, T)$ denote the gathering time from all vertices in tree T to vertex v. Therefore, our goal is to find a vertex v such that $g(v, T)$ is minimum. This vertex is called the *gathering center* of T. Similar to broadcasting center, the gathering center may not be unique. A heterogeneous tree may have more than one gathering center. Let $GC(T)$ denote all the gathering centers of a tree T. $GC(T)$ is also a star which is a tree itself that all vertices but one are leaves.

The **Algorithm Gathering** uses $t(u)$ to label every vertex u in tree T. Let v be a gathering center of tree T and u a vertex other than v. Let vertex u' be the vertex on the path from u to v that is adjacent to u. The **Algorithm Gathering** computes $t(u) = g(u, T(u, u')) + W(u)$ which is equal to the minimum gathering time from every vertex in tree $T(u, u')$ through u to pass to u'. In other words, $t(u)$ denotes the time at which all messages from every vertex in $T(u, u')$ can be gathered and ready for transmission. Finally we can get that $t(v) = g(v, T)$ where v is a gathering

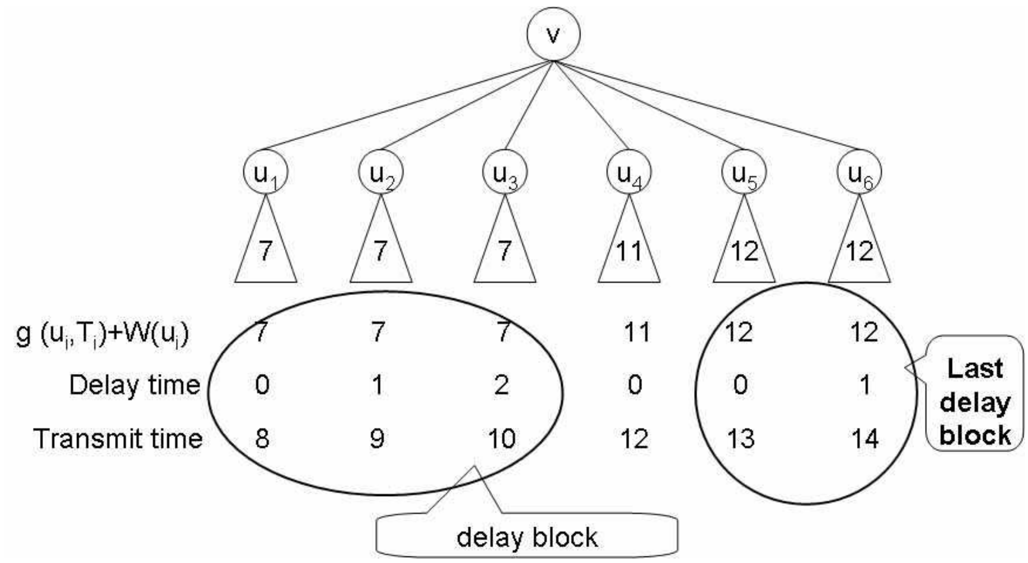

Figure 3: Gathering

center. We use bottom-up method to label vertices iteratively. Initially we label every leaf u with $W(u)$. We will process the vertex in non-increasing order of its associated labels.

But by our model, a vertex can only communicate (send or receive) with one other vertex at a time. Thus, it means that some vertex will be delayed by several units of time until other vertices having higher priority have finished transmission. We use $D(u)$ to label every vertex u in tree T and it is a positive integer. It means vertex u will have a delay of $D(u)$ units of time. Thus the actual time instant at which u is ready to transmit messages to its parent is $t(u) + D(u)$. It means that u can transmit messages to its parents at time no earlier than $t(u) + D(u)$. And vertices u_i, for $1 \leq i < k$ are said to be in the same j^{th} *delay block* when $D(u_i) = 0$ for $1 \leq i_{j-1} \leq i < i_j$, $D(u_i) > 0$ for $i_j \leq i \leq i_{j+1}$ and $D(u_i) = 0$ for $i_{j+1} \leq i < i_{j+2} < k$ (see Figure 3).

Our algorithm will adopt a greedy strategy and process the nodes in a bottom up manner. The key idea is described below. Since a gathering center has minimum gathering time, using bottom-up method is a very reasonable idea. Suppose a vertex v having neighbors $u_1,\ u_2,\ \ldots,\ u_k$ in T. Without loss of generality we suppose that $g(u_1, T(u_1, v)) + W(u_1) \leq g(u_2, T(u_2, v)) + W(u_2) \leq \ldots \leq g(u_k, T(u_k, v)) + W(u_k)$. The transmission strategy between v and its children that we adopt is that as soon as a child node is ready to send messages, we let it do so whenever possible. At first we compute the delay time of all vertices, $D(u_i) = 0$ or $D(u_i) = g(u_{i-1}, T(u_{i-1}, v)) + W(u_{i-1}) + D(u_{i-1}) - g(u_i, T(u_i, v)) + W(u_i)$, for $1 \leq i \leq k$. Since u_1 has the smallest $g(u_1, T(u_1, v)) + W(u_1)$ among all of v's children, sending messages from u_1 to v first, i.e., as soon as the messages have been gathered, is the best strategy. If v would gather messages at any other time instant by holding or delaying u_1 from transmission, the overall gathering time at v won't be better. Thus vertex v will establish the first *telephone call* with u_1 and then with $u_2,\ u_3,\ \ldots,\ u_k$ subsequently in that order. This is referred to as the *message gathering sequence protocol*. The gathering task at vertex v is finished, when vertex v gets messages from u_k. Thus, the gathering time of v will be $g(v, T) = g(u_k, T(u_k, v)) + W(u_k) + D(u_k) + 1$.

Let $\mathcal{L}(\mathcal{T})$ denote the set of leaves of T.

Algorithm Gathering

For each vertex $v \in \mathcal{L}(\mathcal{T})$ **do** $t(v) \leftarrow W(v)$;

For each vertex $v \in T$ **do** $D(v) \leftarrow 0$;

$U \leftarrow \emptyset$;

$T' \leftarrow T$;

while $|V(T')| \geq 2$ **do**

Let w be a leaf of T' and $t(w) = \min\ \{t(w_i) | w_i \in \mathcal{L}(\mathcal{T}')\}$;

Let v be the vertex adjacent to w in T';

$T' \leftarrow T' - \{w\}$;

$U \leftarrow U \cup \{w\}$;

if v is a leaf of T'

then Let v be adjacent to some vertices which are in U. Suppose that they are $u_1,\ u_2,\ \ldots,\ u_k$ and $t(u_1) \leq t(u_2) \leq \ldots \leq t(u_k)$;

for $m = 2$ to k

if $t(u_{m-1}) + D(u_{m-1}) - t(u_m) \geq 0$

$D(u_m)$=$t(u_{m-1}) + D(u_{m-1}) - t(u_m) + 1$;

$t(v) \leftarrow t(u_k) + D(u_k) + 1 + W(v)$;

end;

Let v be the remaining vertex left in T';

Let the neighbors of v in T be $u_1,\ u_2,\ \ldots,\ u_k$ and

$t(u_1) + D(u_1) \leq t(u_2) + D(u_2) \leq \ldots \leq t(u_k) + D(u_k)$;

if $W(v) = 0$

if vertex u_i is in *final delay block*

$GC(T) \leftarrow \{v, u_i\}$;

else

if $t(u_k) - t(u_{k-1}) - 1 = W(v)$

$GC(T) \leftarrow \{v, u_k\}$;

$g(T) \leftarrow t(u_k) + 1$;.

3.2 Proof of Correctness

Theorem 3.1 *Suppose that $u_1,\ u_2,\ \ldots,\ u_k$ are the neighbors of vertex v in tree T. Without loss of generality, we suppose $g(u_1, T(u_1, v)) + W(u_1) \leq g(u_2, T(u_2, v)) +$*

$W(u_2) \leq \ldots \leq g(u_k, T(u_k, v)) + W(u_k)$. *Then*

$$g(v, T) = g(u_k, T(u_k, v)) + W(u_k) + D(u_k) + 1$$

Proof: We use greedy method like First-Come-First-Served (FCFS) to solve this situation. We let vertex u_i pass messages to vertex v whenever u_i is ready to do so. But if there are two or more vertices which are ready to pass messages to vertex v, the delays of transmission of messages from these vertices will occur. We know that vertex v can only communicate with one vertex at a time. So the other vertex will wait to transmit messages to v in order. Thus, we can compute the delay time of every vertex easily. The first vertex u_1 can transmit messages to v at time $g(u_1, T(u_1, v)) + W(u_1)$ without any delay, and thus $D(u_1) = 0$. When $g(u_{i-1}, T(u_{i-1}, v)) + W(u_{i-1}) + D(u_{i-1}) \geq g(u_i, T(u_i, v)) + W(u_i)$, it means that u_i will delay $D(u_i) = g(u_{i-1}, T(u_{i-1}, v)) + W(u_{i-1}) + D(u_{i-1}) - g(u_i, T(u_i, v)) - W(u_i) + 1$ units of time. In other words, u_i will wait $D(u_i)$ units of time before it transmits messages to v. It is like job scheduling problem. Thus, we conclude that the time at which vertex u_k passes messages to vertex v is the gathering time of tree T. □

Lemma 3.1 *For any vertex u, $t(u) = g(u, T(u, u')) + W(u)$.*

Proof: It is the same as Lemma 2.1 □

Lemma 3.2 *Assume that u and v are adjacent vertices in tree T and have vertex weights $W(u)$ and $W(v)$ respectively. Let $g(u, T(u, v)) + W(u) \leq g(v, (T(v, u)) + W(v)$.*

1. $g(u, T) = g(v, T(v, u)) + W(v) + 1$

2. $g(v, T(v, u)) \leq g(v, T) \leq max\{g(v, T(v, u)) + 1,\ W(u) + g(u, T(u, v)) + 1\}$

Proof:

1. From the assumption we have $g(u, T(u, v)) < g(v, (T(v, u)) + W(v)$. Thus by Theorem 3.1, we will gather messages from vertices in $T(u, v)$ to vertex u at first and the messages gathered from vertex v will not incur any delay. Thus, $g(u, T) = g(v, T(v, u)) + W(v) + 1$.

2. $T(v, u)$ is a subtree of tree T, and therefore $g(v, T(v, u)) \leq g(v, T)$. In our model, the vertex v has two ways to gather messages. One way is to gather messages from vertices in $T(v, u)$ first, and another is to gather messages from vertices from u in $T(u, v)$. Each costs $g(v, T(v, u))$ and $g(u, T(u, v)) + W(u) + 1$ respectively. So we select the maximum between $g(v, T(v, u))$ and $g(u, T(u, v)) + W(u) + 1$.

   Case 1: $g(v, T(v, u)) \geq W(u) + g(u, T(u, v)) + 1$**.** In this case the vertex u will transmit messages to vertex v when u is ready to do so, and during the transmission v gathers messages from its neighbors in $T(v, u)$. By Theorem 3.1 it is easy to see that vertex v may take additional one unit of time to gather messages from vertex u during the transmission when v gathers messages from its neighbors in $T(v, u)$. Thus, $g(v, T(v, u)) \leq g(v, T) \leq g(v, T(v, u)) + 1$.

   Case 2: $W(u) + g(u, T(u, v)) + 1 > g(v, T(v, u))$**.** The vertex v will get all messages from the vertices in tree $T(v, u)$ in $g(v, T(v, u))$ units of time. Then vertex v can get messages of all vertices in $T(u, v)$ from vertex u at time $W(u)+g(u,T(u,v))+1$. Therefore, $g(v,T) = W(u)+g(u,T(u,v))+1$.

By the description above, $g(u,T) \leq max\{g(v,T(v,u))+1, W(u)+g(u,T(u,v))+1\}$. □

Lemma 3.3 *If $w_1,\ w_2,\ \ldots,\ w_{n-1}$ is the sequence of vertices selected by **Algorithm Gathering**, and w_n is the one remaining vertex of algorithm. Then $t(w_1) \leq t(w_2) \leq \ldots \leq t(w_n)$.*

Proof: It is the same as Lemma 2.3. □

Lemma 3.4 *Let vertices v and u be adjacent in T', and u be selected by **Algorithm Gathering** in the while loop. Then $g(u,T) \geq g(v,T)$.*

Proof: First if $V(T') = \{u,v\}$, then the labels on u and v are $t(u) = g(u,T(u,v)) + W(u)$ and $t(v) = g(v,T(v,u)) + W(v)$. By Lemma 3.3 we know $t(u) \leq t(v)$. By Lemma 3.2, $g(u,T) \geq g(v,T)$.

Second, we consider that $|V(T')| \geq 3$. We suppose that there is a leaf $x \neq u$ and $x \neq v$ at the time when vertex u is selected. Let x' be the vertex in T' adjacent to x. By Lemma 3.3 $g(x,T(x,x')) + W(x) \geq g(u,T(u,v)) + W(u)$, and we know that $T(x,x')$ is a subtree of $T(v,u)$. It implies that $g(v,T(v,u)) \geq g(x,T(x,x')) + W(x)+1$, which is no less than $g(u,T(u,v))+W(u)+1$,and so $g(v,T(v,u))+W(v) \geq g(u,T(u,v)) + W(u)$.

By Lemma 3.2, we have $g(u,T) \geq g(v,T)$. □

Lemma 3.5 *Let u and v be adjacent vertices in tree T with weights $W(u)$ and $W(v)$ respectively. Without loss of generality we suppose that $g(u, T(u, v)) + W(u) \leq g(v, T(v, u)) + W(v)$. For each vertex s in $T(u, v) \setminus \{u\}$, $g(s, T) > g(v, T)$*

Proof: By Lemma 3.2 we have two cases: one is $g(u, T(u, v)) + W(u) + 1 \leq g(v, T(v, u))$, and the other is $g(v, T(v, u)) < g(u, T(u, v)) + W(u) + 1$. At first we know that $g(s, T) \geq 1 + W(u) + 1 + W(v) + g(v, T(v, u))$.

Case 1: $g(u, T(u, v)) + W(u) + 1 \leq g(v, T(v, u))$**.** Then by Lemma 3.2 we have $g(v, T) \leq g(v, T(v, u)) + 1$. Therefore, $g(s, T) > g(v, T)$.

Case 2: $g(u, T(u, v)) + W(u) + 1 > g(v, T(v, u))$**.** Thus, by Lemma 3.2 $g(v, T) = g(u, T(u, v)) + W(u) + 1$. Since $g(u, T(u, v)) + W(u) \leq g(v, (T(v, u)) + W(v)$, we have $g(s, T) \geq 1 + W(u) + 1 + W(v) + g(v, T(v, u)) > g(u, T(u, v)) + W(u) + 1 = g(v, T)$. □

Lemma 3.6 *Let v be the single remaining vertex in **Algorithm Gathering** after the while loop. If there exists some vertex u such that $g(u, T)$ is equal to $g(v, T)$, then u and v must be adjacent.*

Proof: Suppose that vertex s is not adjacent to v in tree T, and vertex u is adjacent to v in the path from s to v. We shall prove that if s and v are not adjacent, then $g(s, T) > g(v, T)$.

By Lemma 3.3 we know $g(u, T(u, v)) + W(u) \leq g(v, T(v, u)) + W(v)$.

This shows that the constraint of Lemma 3.2 holds and vertex s is in subtree $T(u, v) \setminus \{u\}$.

By Lemma 3.5 we have $g(s,T) > g(v,T)$. □

This lemma shows that if the Gathering Center of tree T is not unique, then it must be a star. Thus, we only need to check the vertices adjacent to the remaining vertex v to find out if any of them is also a Gathering Center. The following theorem shows that the method of finding the gathering center is correct.

Theorem 3.2 *Suppose v is the single remaining vertex in **Algorithm Gathering** after the while loop and let $u_1, u_2, \ldots, u_k$ be the neighbors of v. Without loss of generality we let $g(u_1, T(u_1, v)) + W(u_1) \leq g(u_2, T(u_2, v)) + W(u_2) \leq \ldots \leq g(u_k, T(u_k, v)) + W(u_k)$.*

1. *If $W(v) = 0$, every vertex in the last delay block is also a gathering center.*
2. *If $W(v) \geq 1$, and $g(u_k, T(u_k, v)) + W(u_k) + D(u_k) - g(u_{k-1}, T(u_{k-1}, v)) - W(u_{k-1}) - D(u_{k-1}) - 1 = W(v)$, u_k is also a gathering center.*

Proof: If vertex u_s is also the gathering center of T, it means that other vertices in $T(v, u_s)$ which transmit messages to vertex v need $g(v,T) - W(v) - 1$ units of time. Thus vertex v transmits messages to vertex u_k at time $g(v,T) - 1$.

Case 1: $W(v) = 0$. If vertex u_s is not in the *last delay block*, the vertices $u_1, \ldots, u_{s-1}, u_{s+1}, \ldots, u_k$ passing messages to vertex v also need $g(v,T)$ units of time. Thus, $g(u_s, T) = g(v,T) + W(v) + 1 > g(v,T)$. If u_s is in the *last delay block*, the vertices $u_{s+1}, \ldots, u_k$ passing messages to vertex v can save one unit of time. It means that $g(v, T(v, u_s))) = g(v,T) - 1$. Thus, $g(u_s, T) = g(v,T) - 1 + W(v) + 1 = g(v,T)$.

Case 2: $W(v) \geq 1$**.** At first we know $g(v, T) = g(u_k, T(u_k, v)) + W(u_k) + D(u_k) + 1$. We select vertex v_s, for $1 \leq s \leq k-1$. By Theorem 3.1, $g(v, T) - 1 \leq g(v, T(v, u_s)) \leq g(v, T)$. Thus $g(u_s, T) = g(v, T(v, u_s)) + W(v) + 1 > g(v, T)$. Therefore vertex v_s, for $1 \leq s \leq k-1$, are not gathering center of tree T. If $(g(u_k, T(u_k, v)) + W(u_k) + D(u_k)) - g(u_{k-1}, T(u_{k-1}, v)) - W(u_{k-1}) - D(u_{k-1}) - 1 = W(v)$ then $g(v, T(v, u_k)) = g(u_{k-1}, T(u_{k-1}, v)) + W(u_{k-1}) + D(u_{k-1}) + 1$. Therefore $g(u_k, T) = g(v, T(v, u_k)) + W(v) + 1 = g(u_k, T(u_k, v)) + W(u_k) + D(u_k) + 1 = g(v, T)$. □

Property 3.1 *Let x be a vertex in tree T and $x \notin GC(T)$. There exists a path with the smallest distance from x to a vertex $v \in GC(T)$ containing vertex u, where vertex u is adjacent v such that $g(x, T) = k + g(u, T)$, where k is the distance from x to u.*

Proof: It is similar as Property 2.1. □

Lemma 3.7 *The time complexity of* ***Algorithm Gathering*** *for solving the gathering problem in heterogeneous tree networks is $O(n \log n)$*

Proof: It is the same as Lemma 2.7. □

Finally to obtain an optimal sequence of calls, we use the labels $t(v)$ and $D(v)$ of each vertex v in tree T. By Theorem 3.1 we know the optimal strategy of gathering in trees. Let vertex $v \in$ tree T and $u_1,\ u_2,\ \ldots,\ u_k$ be children of v. Without loss of generality, we suppose that $t(u_1) \leq t(u_2) \leq \ldots \leq t(u_k)$. Every vertex u_i will transmit messages to v at time $t(u_i) + D(u_i)$. When u_k has finished transmitting, we can compute $t(v) = t(u_k) + D(u_k) + 1 + W(v)$. Thus we can get the optimal sequence of calls and know what actions to take at each unit of time.

4 Set to Set Broadcasting

In this chapter, we use algorithms obtained in Chapter 2 and Chapter 3 to solve the *Set to Set Broadcasting Problem*. Let A and B be two nonempty set of $V(T)$. We let $t(A, B, T)$ be the broadcasting time of set A passing messages to set B in tree T. Our method will find the broadcasting center and gathering center of tree T. Some properties of broadcasting center are useful to solve this problem.

Before starting to describe our algorithm, we define some notation. Let T_A (respectively T_B) be the minimum subtree of T containing A (respectively B). It implies that all leaves of T_A are in set A. Our algorithm is divided into two parts:

1. Find the gathering center $v_a \in GC(T_A)$ and the broadcasting center $v_b \in BC(T_B)$, such that v_a and v_b have the shortest distance.

2. If $v_a = v_b$ then A gathers messages at vertex v_b and then v_b broadcasts to B. If not, there is a path P between v_a and v_b, and we then find the point u_a and u_b on P adjacent respectively to v_a and v_b. Set A gathers messages at a vertex between u_a and u_b, then broadcasts to B.

Algorithm Set to Set Broadcasting;

begin {

Compute T_A and T_B;

Use **Algorithm Gathering** and **Algorithm Broadcasting** to find $GC(T_A)$ and

$BC(T_B)$;

if $|GC(T_A) \bigcap BC(T_B)| \geq 1$

then [$A \subseteq V(T_A)$ transmits messages to a vertex c which is in $GC(T_A) \bigcap BC(T_B)$

by an optimal sequence of calls based on *message gathering sequence protocol.*

c broadcasts to $B \subseteq V(T_B)$ by an optimal call sequence of calls based on

message broadcasting sequence protocol.

Thus, $t(A, B, T) \longleftarrow g(T_A) + b(T_B)$];

else

[Let path P be the shortest path from $v_c \in GC(T_A)$ to $v_b \in BC(T_B)$. Suppose

that there exists a vertex u_a(respectively u_b) which is adjacent to v_a

(respectively v_b) in path P;

$A \in V(T_A)$ gathers messages at vertex u_a;

u_a transmits messages to u_b;

u_b broadcasts messages to $B \in V(T_B)$;

$t(A, B, T) \longleftarrow g(u_a, T(u_a, v_a)) + \delta(u_a, u_b) + b(u_b, T(u_b, v_b))$, where $\delta(a, b)$ denotes

the path length between vertices a and b on T.];

} **end**

We need the following lemmas to prove the correctness of **Algorithm Set to Set Broadcasting**.

Lemma 4.1 *Given a tree T, let vertex v be the remaining vertex in* ***Algorithm Broadcasting*** *after the while loop, and $u_1,\ u_2,\ \ldots,\ u_k$ be the neighbors of v in tree T. We denote $\hat{T}_i = T(v, u_i) \cup \{u_i\}$. Then*

$$b(u_i, \hat{T}_i) = b(u_i, T) = W(u_i) + 1 + b(v, T(v, u_i)),\ for\ 1 \leq i \leq k$$

Proof: Vertex u_i is a leaf in tree $\hat{T}_i$. Thus, $b(u_i, \hat{T}_i) = W(u_i) + 1 + b(v, T(v, u_i))$.

By Lemma 2.3 we have $b(v, T(v, u_i)) \geq b(u_i, T(u_i, v))$.

By Lemma 2.4 we know $W(u_i) + b(v, T(v, u_i)) \geq W(v) + b(u, T(u_i, v))$.

Finally by Lemma 2.2 we have $b(u_i) = W(u_i) + 1 + b(v, T(v, u_i))$.

Thus we have $b(u_i, \hat{T}_i) = b(u_i, T) = W(u_i) + 1 + b(v, T(v, u_i))$. □

This lemma shows an obvious property. Let u_i be a vertex adjacent to the broadcasting center v. The broadcasting time in subtree $T(u_i, v)$ satisfies the inequality $b(u_i, T(u_i, v)) \leq W(u_i) + 1 + b(v, T(v, u_i))$.

Lemma 4.2 *Give a tree T, suppose that vertex v is the remaining vertex in **Algorithm Gathering** after the while loop, and $u_1, u_2, \ldots, u_k$ are neighbors of v in tree T. Let $\hat{T}_i = T(v, u_i) \cup \{u_i\}$. Then $g(u_i, \hat{T}_i) = g(u_i, T) = W(v) + g(v, T(v, u_i)) + 1$ for $1 \leq i \leq k$.*

Proof: It is the same as Lemma 4.1. □

Lemma 4.3 *If T_A and T_B are two subtrees of T such that $V(T_A) \cap V(T_B) = \{x\}$, then $t(A, B, T) = t(A, \{x\}, T) + t(\{x\}, B, T)$.*

Proof: If Q_1 and Q_2 are a sequence of calls by which A gathers the messages at x and x broadcasts the messages to B respectively, then the concatenation $Q_1||Q_2$ of Q_1 and Q_2 is a feasible solution of set to set broadcasting from A to B. So $t(A, B, T) \leq t(A, \{x\}, T) + t(\{x\}, B, T)$.

Suppose that sequence Q^1 is an optimal sequence of calls to solve the set to set

[1] A sequence of calls can be represented by a sequence of edges $e(u, v)$ plus time instant at which the call is made between sender and receiver, i.e., vertices u and v.

broadcasting problem from A to B. We permute the calls in Q. Suppose we ignore the calling times of the calling sequence, and consider Q simply as a set of edges. Let $Q_3 = Q \bigcap E(T_A)$ and $Q_4 = Q \bigcap E(T_B)$, where $E(T_A)$ ($E(T_B)$) denotes the set of edges of tree T_A (T_B respectively). Consider the new sequence $Q_3||Q_4$. Note that Q_3 consists of the calls from vertices in set A to gather the messages at vertex x, and Q_4 consists of the calls from vertex x to broadcast the messages to vertices in set B. Thus, $t(A, B, T) = |Q_3| + |Q_4| \geq t(A, \{x\}, T) + t(\{x\}, B, T)$, where $|Q|$ denotes the time taken by the calling sequence Q. □

Theorem 4.1 ***Algorithm Set to Set Broadcasting*** *gives an optimal sequence of calls to broadcast from vertices in set A to vertices in set B in tree T. If $|BC(T_A) \cap BC(T_B)| \geq 1$, then $t(A, B, T) = g(T_A) + b(T_B)$. Otherwise $t(A, B, T) = g(T_A) + k + b(T_B)$, where k is the shortest distance from $BC(T_A)$ to $BC(T_B)$.*

Proof: We distinguish two cases

Case 1: $|GC(T_A) \cap BC(T_B)| \geq 1$**.** In this case we can choose a vertex $x \in GC(T_A) \cap BC(T_B)$. We gather messages at x from A and then broadcast from x to B. It costs $g(T_A) + b(T_B)$. The optimality is shown below.

If we choose another vertex y such that $y \notin GC(T_A) \bigcap BC(T_B)$, it has three situations.

- $y \notin GC(T_A) \bigcup BC(T_B)$. The gathering time $g(y, T_A) > g(T_A)$ and $b(y, T_B) > b(T_B)$. Thus, $g(y, T_A) + b(y, T_B) > g(T_A) + b(T_B)$.
- $y \in GC(T_A)$ and $y \notin BC(T_B)$. The gathering time $g(y, T_A) = g(T_A)$ and the broadcasting time $b(y, T_B) > b(T_B)$. Thus, $g(y, T_A) + b(y, T_B) >$

$g(T_A) + b(T_B)$.

- $y \in BC(T_B)$ and $y \notin GC(T_A)$. The gathering time $g(y, T_A) > g(T_A)$ and broadcasting time $b(y, T_B) = b(T_B)$. Thus, $g(y, T_A) + b(y, T_B) > g(T_A) + b(T_B)$.

From the above discussion we know that $x \in GC(T_A) \bigcap BC(T_B)$ is the best choice.

Case 2: $|GC(T_A) \cap BC(T_B)| = 0$**.** There exists a shortest path P' from $v_a \in GC(T_A)$ to $v_b \in BC(T_B)$. Let u_a be the vertex on P' adjacent to v_a and u_b be the vertex on P' adjacent to v_b. The shortest path between u_a and u_b is denoted as P and suppose that for u_a to pass messages to u_b it needs k units of time. So $t(A, B, T) \leq g(u_a, T_A) + k + b(u_b, T_B)$.

$t(A, B, T) = t(V(T_A), V(T_B), T) \geq t(V(\hat{T}_{u_a}), V(\hat{T}_{u_b}), T)$. This follows from the following. If $u_a \in V(T_A), V(\hat{T}_{u_a}) \subseteq V(T_A)$. If $u_a \notin V(T_A)$, u_a does not affect the gathering of T_A and u_a is in the unique path from T_A to T_B. Thus, $t(V(T_A), V(T_B), T) = t(V(\hat{T}_{u_a}), V(T_B), T)$. The same holds for u_b.

Let $T_1 = \hat{T}_{u_a} \cup \{P\}$, $T_2 = \hat{T}_{u_b}$. So $\hat{T}_1 \cap \hat{T}_2 =$ vertex $\{u_b\}$. Vertex u_a to vertex u_b has the unique path P.

By Lemma 4.3 $t(A, B, T) \geq t(V(\hat{T}_{u_a}), V(\hat{T}_{u_b}), T) = t(V(T_1), V(T_2), T) = t(V(\hat{T}_{u_a}), u_b, T) + t(u_b, V(\hat{T}_{u_b}), T) = g(u_a, V(\hat{T}_{u_a})) + k + b(u_b, V(\hat{T}_{u_b}))$.

By Lemma 4.1 and Lemma 4.2 we know $g(u_a, \hat{T}_{u_a}) = g(u_a, T_A)$ and $b(u_b, \hat{T}_{u_b}) = b(u_b, T_B)$.

Thus, $t(A, B, T) \geq t(V(\hat{T}_{u_a}), \{u_b\}, T) + t(\{u_b\}, V(\hat{T}_{u_b}), T) = g(u_a, V(\hat{T}_{u_a})) + k + b(u_b, V(\hat{T}_{u_b})) = g(u_a, T_A) + k + b(u_b, T_B))$. □

Lemma 4.4 *The time complexity of **Algorithm Set to Set Broadcasting** for solving the Set to Set Broadcasting Problem in heterogeneous tree networks is $O(n \log n)$, where n is the total number of nodes in the heterogeneous tree networks.*

Proof: This algorithm is divided into three parts. The first one is to compute the gathering time and $GC(T_A)$ of T_A and broadcasting time and $BC(T_B)$ of T_B. From the algorithms given in Chapter 2 and Chapter 3, we know that the time complexity of this part is $O(n \log n)$. The second one is to compute the shortest path from v_a to v_b such that $v_a \in GC(T_A)$ and $v_b \in BC(T_B)$. It takes $O(n)$ time. The last part is to compute an optimal sequence of calls. We can easily get the optimal sequence of calls of gathering and broadcasting by concatenating the optimal sequences of calls obtained from the algorithms given in Chapter 2 and Chapter 3. The size of the optimal sequence of calls is at most n. Therefore, the time complexity of the algorithm is $O(n \log n)$. □

5 Broadcasting Center for a Tree with Vertex and Edge Weights

In this chapter, we will provide an idea to calculate the minimum broadcasting time and find the broadcasting center of a heterogeneous tree network with both vertex weight and edge weight under the telephone model. We will get an optimal sequence of telephone calls in $O(n \log n)$ time, where n is the number of nodes in the network.

We recall our communication model in the following. Given a connected graph $G = (V, E)$ with each vertex $v_i \in V$ having weight $W(v_i)$, where the weight is a non-negative integer. Each edge $e = (u, v) \in E$ has weight $W(u, v)$, where the weight is also a non-negative integer. This is the same as vertex weight model discussed earlier if the edge weight of each edge in E is 0. Transmitting messages in G has the following constraints:

1. A vertex can only send messages to its adjacent vertex, and it takes $W(u, v)$ units of time.

2. A vertex can only communicate (send or receive) with one other vertex at a time.

3. After receiving a message, a vertex v needs $W(v)$ units of time to process that message. Only after the processing can it send the message to other vertices.

4. When finishing the processing of the received messages or a transmission of messages to an adjacent vertex, a vertex can send messages to another vertex in next unit of time. We call it a *switching delay*.

Although this problem is like Chapter 2, the method in Chapter 2 does not work for this model. Suppose a vertex v has neighbors $u_1,\ u_2,\ \ldots,\ u_k$ in T. Without loss of generality we suppose that $b(u_1, T(u_1, v)) + W(u_1, v) \geq b(u_2, T(u_2, v)) + W(u_2, v) \geq \ldots \geq b(u_k, T(u_k, v)) + W(u_k, v)$. By the Theorem in Chapter 2 when v needs to broadcast messages, we will pass messages to u_i from 1 to k in that order. So v will transmit messages to u_1 at time $W(v) + 1$, but transmitting messages across to edge (u_1, v) needs $W(u_1, v)$ units of time. Thus vertex v is occupied by u_1 when messages are being transmitted to u_1. v will be ready to transmit messages

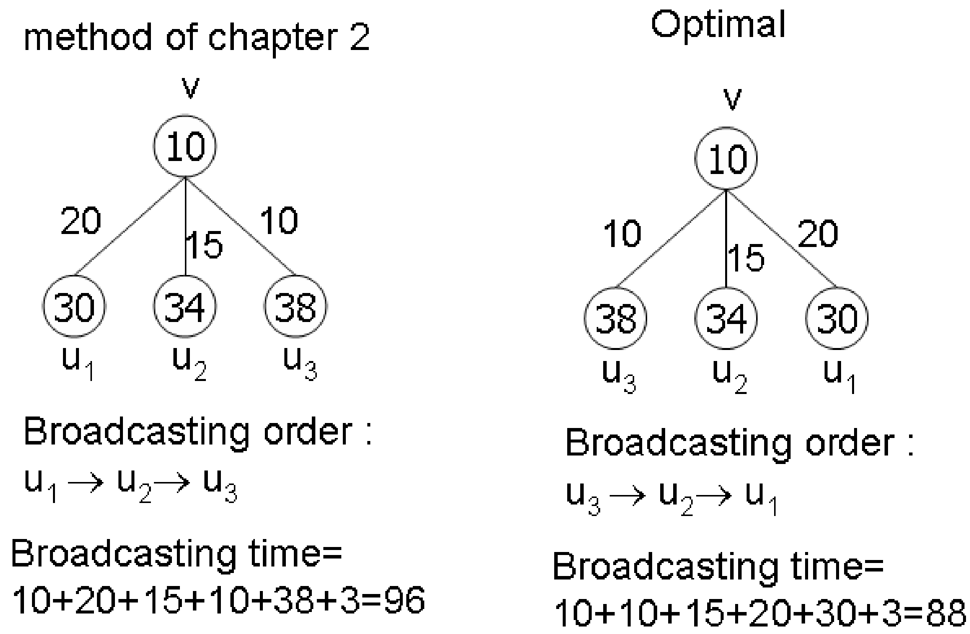

Figure 4: The mistake of Theorem 2.1 of Broadcasting with Edge Weight

to u_2 when u_1 has received the messages. Thus, v will transmit messages to u_2 at time $W(v) + W(v, u_1) + 2$. In other words v transmits messages to u_i at time $W(v) + W(u_1, v) + \ldots + W(u_{i-1}, v) + i$. In Figure 4 we can easily see that the transmission order given in of Chapter 2 is not the best strategy.

Suppose we assume another communication model, e.g., multiplexing communication model, to simulate the broadcasting problem of a heterogeneous tree network with both vertex and edge weights under the telephone model. We add a relay vertex r_i on each edge $e(v, u_i)$ between v and u_i with edge weights $W(v, r_i) = 0$, $W(r_i, u_i)$ equal to $W(v, u_i)$ and vertex weight $W(r_i) = 0$. Thus, vertex v can transmit messages to r_i in one unit time in the order specified by the identity permutation of {1, 2, . . . , k } according to the *message broadcasting sequence protocol* given in Chapter 2. When v finishing a transmission to a relay node, v can send messages to another relay node in next unit of time instant due to the switching delay. Thus, r_i will transmit messages to u_i in a way similar to the telephone model. It means that vertex v transmits messages to $r_1,\ r_2,\ \ldots,\ r_k$ at time $W(v)+1, W(v)+2, \ldots, W(v)+k$ and then r_i can send messages to u_i immediately. Thus, the algorithm given in Chapter 2 will be applicable here and we can use the algorithm similar to Chapter 2 to solve the Broadcasting problem with both vertex and edge weights.

6 Conclusion

In this thesis we have presented several algorithms whose time complexities are all $O(n \log n)$. We can determine an optimal sequence of calls to solve the Broadcasting Problem for heterogeneous tree networks. We also solve the Gathering Problem in $O(n \log n)$ time using a similar idea as in Broadcasting Problem. Finally we use the solutions to the Broadcasting and Gathering Problems to get an optimal solution to Set to Set Broadcast Problem for heterogeneous tree networks in $O(n \log n)$ time.

How to solve the Broadcasting Problem with both vertex and edge weights for heterogeneous tree networks under the ordinary telephone model, without resorting to the modification of communication model, e.g., multiplexing communication model, remains to be an open problem.

References

[1] A. Bar-Noy and S. Kipnis, Designing broadcasting algorithms in the Postal model for message-passing systems, Math. Systems Theory, 27 (1994), pp. 431V452.

[2] A. Bar-Noy, S. Guha, J. Naor and B. Schieber. Multicasting in Heterogeneous Networks. Proceedings of the 13th Annual ACM Symposium on Theory of Com-

puting, pp. 448V453, 1998.

[3] G. Fox, M. Johnson, G. Lyzenga, S. Otto, J. Salmon and D. Walker, Solving Problems on Concurrent Processors, Volume I : General Techniques and Regular Problems, Prentice- Hall, Englewood Cliffs, NJ, 1988.

[4] M. R. Garey and D. S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, W. H. Freeman, San Francisco, CA, 1979.

[5] R. Karp, A. Sahay, E. Santos and K. E. Schauser, Optimal broadcast and summation in the LogP model, in Proceedings of the Fifth Symposium on Parallel Algorithms and Architectures (SPAA), Association for Computing Machinery, New York, 1993.

[6] Hsun-Ming Lee and Gerard J. Chang, Set to set broadcasting in communication networks, Discrete Applied Mathematics, v.40 n.3, p.411-421, Dec. 14, 1992 [doi¿10.1016/0166-218X(92)90010-8].

[7] D. Richards and A.L. Liestman, Generaliazation of broadcasting and gossiping, networks 18(1988) 125-138.

[8] P.J. Slater, E. Cockayne and S.T. Hedetnjemi. Information disseminationin trees. SIAM J. Comput. 10 (1981) 692-701

[9] M. Middendorf, Minimum broadcast time is NP-complete for 3-regular planar graphs and deadline 2, Inform. Process. Lett., 46 (1993), pp. 281V287.

[10] G. Kortsarz and D. Peleg, Approximation algorithm for minimum time broadcast, SIAM J. Discrete Math., 8 (1995), pp. 401V427.

[11] R. Ravi, Rapid rumor ramification: Approximating the minimum broadcasting time, in Proceedings of the 35th Symposium on Foundations of Computer Science (FOCS), Santa Fe, NM, IEEE Press, Piscataway, NJ, 1994, pp. 202V213.