SIAM J. Comput. © 2000 Society for Industrial and Applied Mathematics
Vol. 30, No. 2, pp. 347–358

Downloaded 01/21/20 to 129.177.122.70. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

MESSAGE MULTICASTING IN HETEROGENEOUS NETWORKS*

AMOTZ BAR-NOY[†], SUDIPTO GUHA[‡], JOSEPH (SEFFI) NAOR[§], AND BARUCH SCHIEBER[¶]

Abstract. In heterogeneous networks, sending messages may incur different delays on different links, and each node may have a different switching time between messages. The well-studied telephone model is obtained when all link delays and switching times are equal to one unit. We investigate the problem of finding the minimum time required to multicast a message from one source to a subset of the nodes of size k. The problem is NP-hard even in the basic telephone model. We present a polynomial-time algorithm that approximates the minimum multicast time within a factor of $O(\log k)$. Our algorithm improves on the best known approximation factor for the telephone model by a factor of $O(\frac{\log n}{\log\log k})$. No approximation algorithms were known for the general model considered in this paper.

Key words. approximation algorithms, multicast, postal model, LogP model, heterogeneous networks, combinatorial optimization

AMS subject classifications. 68Q25, 68R10, 05C85, 68W25, 90B18, 68M10

PII. S0097539798347906

1. Introduction.

The task of disseminating a message from a source node to the rest of the nodes in a communication network is called *broadcasting*. The goal is to complete the task as fast as possible assuming all nodes in the network participate in the effort. When the message needs to be disseminated only to a subset of the nodes, this task is referred to as *multicasting*. Broadcasting and multicasting are important and basic communication primitives in many multiprocessor systems. Current networks usually provide point-to-point communication only between some of the pairs of the nodes in the network. Yet, in many applications, a node in the network may wish to send a message to a subset of the nodes, where some of them are not connected to the sender directly. Due to the significance of this operation, it is important to design efficient algorithms for it.

Broadcast and multicast operations are frequently used in many applications for message-passing systems (see [10]). It is also provided as a communication primitive by several collective communication libraries, such as Express by Parasoft [7] and the message-passing library (MPL) [1, 2] of the IBM SP2 parallel systems. This operation is also included as part of the collective communication routines in the

*Received by the editors November 23, 1998; accepted for publication (in revised form) November 18, 1999; published electronically June 3, 2000. Part of this research was done while the first three authors visited IBM T.J. Watson Research Center. A preliminary version of this paper appeared in Proceedings of the 30th ACM Symposium on Theory of Computing, Dallas, TX, 1998, pp. 448–453.

http://www.siam.org/journals/sicomp/30-2/34790.html

[†] AT&T Research Labs, 180 Park Ave., P.O. Box 971, Florham Park, NJ 07932 (amotz@research.att.com). On leave from the Department of Electrical Engineering–Systems, Tel Aviv University, Tel Aviv 69978, Israel.

[‡] Computer Science Department, Stanford University, Stanford, CA 94305 (sudipto@cs.stanford.edu). This author was supported by an IBM fellowship, ARO MURI grant DAAH04-96-1-0007, and NSF award CCR-9357849, with matching funds from IBM, Schlumberger Foundation, Shell Foundation, and Xerox Corporation.

[§] Bell Laboratories, Lucent Technologies, 600 Mountain Ave., Murray Hill, NJ 07974 (naor@research.bell-labs.com). On leave from the Computer Science Department, Technion, Haifa 32000, Israel.

[¶] IBM T.J. Watson Research Center, P.O. Box 218, Yorktown Heights, NY 10598 (sbar@watson.ibm.com).

Downloaded 01/21/20 to 129.177.122.70. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

message-passing interface (MPI) standard proposal [22]. Application domains that use broadcast and multicast operations extensively include scientific computations, network management protocols, database transactions, and multimedia applications. In most of these applications the efficiency depends on the time it takes to complete the broadcast or multicast operations.

There are two basic models in which trivial optimal solutions exist. In the first model, all nodes are assumed to be connected, a node may send a message to at most one other node in each round, and it takes one unit of time (round) for a message to cross a link. Therefore, in each round the number of nodes receiving the message can be doubled. If the target set of nodes is of size k, then this process terminates in $\lceil \log k \rceil$ rounds. In the second model the communication network is represented by an arbitrary graph, where each node is capable of sending a message to all of its neighbors in one unit of time. Here, the number of rounds required to deliver a message to a subset of the nodes is the maximum distance from the source node to any of the nodes in the subset.

The model in which a node may send a message to at most one other node in each round is known as the *telephone* model. It is known that for arbitrary communication graphs, the problem of finding an optimal broadcast in the telephone model is NP-hard [11], even for 3-regular planar graphs [19]. Following the two easy cases given above, it is not hard to verify that in the telephone model two trivial lower bounds hold for the minimum broadcast time. The first one is $\lceil \log n \rceil$, where n denotes the number of nodes in the graph, and the second one is the maximum distance from the source node to any of the other nodes. Research in the past three decades has focused on finding optimal broadcast algorithms for various classes of graphs such as trees, grids, and hypercubes. Also, researchers have looked for graphs with minimum number of links for which a broadcast time of $\lceil \log n \rceil$ can be achieved from any source node. Problems related to broadcast which were extensively investigated are the problems of broadcast multiple messages, gossiping, and computing certain functions on all n inputs in a network. See, e.g., [4, 5, 6, 8, 12, 13, 14, 18, 21, 24, 25].

An assumption central to the telephone model is that both sender and receiver are busy during the whole sending process. That is, only after the receiver received the message, both ends may send the message to other nodes. More realistic models in this context are the postal model [3] and the LogP model [17]. The idea there is that the sender may send another message before the current message is completely received by the receiver, and the receiver is free during the early stages of the sending process. We note that in both the postal model and the LogP model it is assumed that the delay of a message between any pair of nodes is the same.

Optimal solutions for broadcast in the postal model are known for the case of a complete graph and for some other classes of graphs. However, not much is known for arbitrary graphs. In the postal model, researchers have also concentrated on other dissemination primitives and almost always assumed that the communication graph is complete.

1.1. Our results.

In this paper we define a more general model based on the postal model and call it the *heterogeneous postal* model. Assume node u sends a message to node v at time 0 and the message arrives at v at time λ_{uv}. The assumption is that u is free to send a new message at time s_u, and v is free from time 0 to time $\lambda_{uv} - r_v$. We call λ_{uv} the *delay* of the link (u, v), s_u the *sending* (or *switching*) time of u, and r_v the *receiving* time of v. By definition, both s_u and r_v are smaller than λ_{uv}. In the single-message multicast problem each node receives no more than a single

Downloaded 01/21/20 to 129.177.122.70. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

message. Thus, for this problem the receiving time has almost no relevance. Because of this, and to keep the presentation clearer, we assume for the rest of the paper that $r_u = s_u$ for all nodes u. Observe that when the delay, sending time, and receiving time are all equal to 1, we obtain the telephone model.

We believe that our framework may be useful to model modern communication networks, where the major components—the processors and the communication links—are not homogeneous. Some processors are faster than others, and some links have more bandwidth than others. These disparities are captured by the different values of the delay and the switching time.

Since finding the minimum multicast time is NP-hard even in the telephone model, we turn our focus to approximation algorithms. The main result we present is an approximation algorithm for computing a multicast scheme in the heterogeneous postal model. The approximation factor is $O(\log k)$, where k denotes the number of processors in the target set. Previous approximation algorithms for multicasting were known only in the telephone model. Kortsarz and Peleg [16] gave an approximation algorithm that produces a solution whose value is bounded away from the optimal solution by an $O(\sqrt{n})$ additive term. This term is quite large, especially for graphs in which the broadcast (multicast) time is polylogarithmic. Later, Ravi [20] gave an algorithm that achieves a multiplicative approximation factor of $O(\frac{\log n \log k}{\log \log k})$.

We also show that it is NP-hard to approximate the minimum broadcast time within a factor of three in a model which is only slightly more complicated than the telephone model.

The rest of the paper is organized as follows. In section 2 we define our model. In section 3 we describe our solution. Finally, in section 4 we show that this problem is hard to approximate by a small constant factor.

2. The model and the problem. We define our model as follows. Let $G = (V, E)$ be an undirected graph representing a communication network, where V is a set of n nodes and E is the set of point-to-point communication links. Let $U \subseteq V$ denote a special set of *terminals*, and let r be a special node termed the *root*. Let the cardinality of the set U be k. To simplify notation assume that $r \in U$.

We associate with each node $v \in V$ a parameter s_v that denotes the sending time. We sometimes refer to s_v as the switching time of v to indicate that this is the time it takes node v to send a new message. In other words, $1/s_v$ is the number of messages node v can send in one round (unit of time). We associate with each node $v \in V$ a parameter r_v that denotes the receiving time. We assume that $r_v = s_v$, for each node v. We associate with each link $(u, v) \in E$ a length λ_{uv} that denotes the communication delay between nodes u and v. By definition, λ_{uv} is greater than both s_u and r_v $(= s_v)$. We can think of the delay λ_{uv} as taking into account the sending time at u and the receiving time at v.

Let the *generalized degree* of node $v \in V$ be the actual degree of v in the graph G multiplied by the switching time s_v. Observe that the generalized degree measures the time it would have taken the node v to send a message to all of its neighbors.

Our goal is to find a minimum-time multicast scheme, that is, a scheme in which the time it takes for all nodes in the set U to receive the message from the root r is minimized. Without loss of generality, we may consider only multicast schemes that are "not lazy," i.e., schemes in which a node that has not finished sending the message to its neighbors (but has already started) is not idle. Such multicast schemes can be represented by an outward directed tree T that is rooted at r and spans all the nodes in U, together with orderings on the edges outgoing from each node in the tree. The

Downloaded 01/21/20 to 129.177.122.70. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

multicast scheme corresponding to such a tree and orderings is a multicast in which each node in the tree upon receiving the message (through its single incoming edge) sends the message along each of its outgoing edges in the specified order. From now on, we refer to the tree in the representation of a multicast scheme as the tree "used" by the scheme.

For a rooted tree T, denote by Δ_T its maximum generalized degree, and by L_T the maximum distance from r to any of the nodes in U (with respect to the lengths λ_{xy} associated with each link (x, y)). By definition, the multicast time of tree T is greater than Δ_T and greater than L_T. Hence, we obtain the following lemma.

LEMMA 1. *Let OPT denote the multicast time of an optimal solution using tree T^*; then $OPT \geq \frac{1}{2}(\Delta_{T^*} + L_{T^*})$*

3. The approximation algorithm.

In this section we describe the approximation algorithm for multicasting a message to a set of terminals U from a root node $r \in U$.

The main tool used by our algorithm is a procedure $ComputeCore(U')$ that computes for a given set of terminals U', where $r \in U'$,

(1) a subset $W \subset U'$ which we call the *core* of U', of size at most $\frac{3}{4}|U'|$, where $r \in W$;

(2) a scheme to disseminate a message known to all the nodes in W to the rest of the nodes in U' in time proportional to the minimum multicast time from r to U'.

The algorithm that computes the multicast scheme proceeds in ℓ phases. Let $U_0 = U$. Upon termination, $U_\ell = \{r\}$. In the ith phase, $i = 1, \ldots, \ell$, procedure *Compute-Core*(U_{i-1}) is invoked to compute

(1) the core of U_{i-1}, denoted by U_i;

(2) a scheme to disseminate the message from U_i to the set U_{i-1} in time proportional to the minimum multicast time from r to U_{i-1}.

Since $|U_i| \leq \frac{3}{4} \cdot |U_{i-1}|$ and $|U_0| = k$, we have that $\ell = O(\log k)$. The resulting multicast scheme is given by looking at the rounds of the algorithm in backward order. Namely, starting at $i = \ell$ downwards, in each round of the multicast scheme the message is disseminated from U_i to U_{i-1}. Since $U_\ell \subset U_{\ell-1} \subset \cdots \subset U_0 = U$, each dissemination phase takes time proportional to the minimum multicast time from r to U. It follows that the multicast time is up to $O(\log k)$ times the optimal multicast time.

In the rest of the section we describe the procedure $ComputeCore(U')$. Let OPT be the minimum multicast time from r to U'. Lemma 1 implies that there exists a tree T^* spanning the set U' such that $\Delta_{T^*} + L_{T^*} \leq 2 \cdot OPT$. The procedure $ComputeCore(U')$ has two main parts. In the first part, we find a set of $|U'|$ paths, one for each terminal, where the ith path connects the terminal u_i to another terminal called $Mate(u_i)$. The paths have the following *path properties*:

Length property: The length of each path is at most $4 \cdot (\Delta_{T^*} + L_{T^*})$.

Congestion property: The generalized degree of the nodes in the graph induced by the paths is at most $6 \cdot (\Delta_{T^*} + L_{T^*})$.

In the second part we design a dissemination scheme using the above paths. We do it by transforming the paths into a set of disjoint *spider* graphs—graphs in which at most one node has degree more than two. These spider graphs have the following *spider properties*:

- Each spider contains at least two terminals from U'.
- The set of spiders spans at least half the nodes in U'.
- The diameter of each spider is at most $4 \cdot (\Delta_{T^*} + L_{T^*})$.

Downloaded 01/21/20 to 129.177.122.70. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

- The generalized degree of the *center* of a spider is at most $6 \cdot (\Delta_{T^*} + L_{T^*})$, where the center of a spider is the unique node with degree larger than two, if such exists, or one of the endpoints of the spider otherwise.

Now, for each spider, we arbitrarily select one of the nodes from U' to the core of U'. Note that each such node can multicast the message to the rest of the terminals in its spider in $O(\Delta_{T^*} + L_{T^*})$ time (linear in OPT). We add all the terminals not contained in any of the spiders to the core of U'. We claim that the size of the core is at most $\frac{3}{4}|U'|$. To see this, let x denote the number of spiders and let y be the number of the terminals in all the spiders. It follows that the size of the core is $|U'| - y + x$. By the first spider property we have that $x \le y/2$ and by the second spider property we get that $y \ge |U'|/2$. Thus,

$$|core(U')| = |U'| - y + x \le |U'| - \frac{y}{2} \le |U'| - \frac{1}{4}|U'| = \frac{3}{4}|U'| \ .$$

We now turn to describe each of the two parts of the procedure $ComputeCore(U')$.

3.1. Finding a set of paths.

We first claim the following lemma, which is analogous to the "tree-pairing" lemma of Ravi [20].

LEMMA 2. *Let T be a tree that spans a set $U' \subseteq V$, and suppose that $|U'|$ is even. There exists a way to pair the nodes of U' and to find paths (in the tree T) connecting each pair such that*

(1) *the paths are edge disjoint;*
(2) *the length of each path is bounded by $2L_T$;*
(3) *the generalized degree of each node in the graph induced by the paths is at most Δ_T.*

Proof. The tree-pairing lemma [20] states that there exists a pairing such that the paths in T connecting each of the pairs are edge disjoint. Consider these paths. Clearly the length of each of these paths is bounded by $2L_T$. The degree, and hence the generalized degree, of every node in the graph induced by the paths is no more than the (generalized) degree in T since we use only the edges of the tree T. Hence, it is bounded by Δ_T. □

The following corollary handles the odd case as well.

COROLLARY 3. *Let T be a tree that spans a set $U' \subseteq V$. There exists a way to pair the nodes of U' and to find paths (in the tree T) connecting each pair such that*

(1) *the length of each path is bounded by $2L_T$;*
(2) *the generalized degree of each node in the graph induced by the paths is at most $2\Delta_T$.*

Proof. The corollary clearly holds if $|U'|$ is even. If $|U'|$ is odd, we pair $|U'| - 1$ of the nodes as in Lemma 2 and pair the last node with any other node. The length of the path connecting the last pair is still bounded by $2L_T$. However, the degree of the subgraph may double up to $2\Delta_T$. □

Recall that the tree T^* spans the nodes of U' and $(\Delta_{T*} + L_{T*}) \le 2 \cdot OPT$. Our objective is to find the set of paths as guaranteed by Corollary 3 with respect to T^*. However, we do not know T^*. Thus, instead, we find a set of *fractional* paths satisfying similar properties. To this end, we write a linear program for finding a set of (fractional) paths that minimizes the sum of two quantities: (1) the maximum over all pairs of the average length of the paths connecting a pair, and (2) the maximum generalized degree of the subgraph induced by the paths connecting the pairs.

The linear program is a variant of multicommodity flow. For each edge (u, v), we define the directed edges (u, v) and (v, u) both of length λ_{uv}. Let $U' = \{v_1, \ldots, v_h\}$.

Downloaded 01/21/20 to 129.177.122.70. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

With each node $v_j \in U'$ we associate commodity j. Node v_j is the source of commodity j and we create an artificial sink t_j with $r_{t_j} = s_{t_j} = 0$. We connect each of the nodes $v' \in U'$, where $v_j \neq v'$, to t_j by a directed edge (v', t_j) of length 0. The objective is to minimize $(L + \Delta)$, where exactly one unit of flow has to be shipped from each v_j to t_j, such that the average length of the flow paths from v_j to t_j is at most $2L$, and the maximum weighted congestion (generalized degree) of the induced subgraph is at most 3Δ.

More formally, let A denote the set of directed edges, and let $f^i(u,v)$ denote the flow of commodity i on directed edge (u,v). The linear programming formulation is as follows.

$$
\begin{array}{llcl}
\multicolumn{4}{c}{\text{Minimize} \quad \Delta + L} \\
\textit{subject to:} & & & \\
\text{For all } 1 \le i \le h & & & \\
\text{and } v \in V - \{v_i, t_i\}\text{:} & \displaystyle\sum_{(u,v)\in A} f^i(u,v) - \sum_{(v,w)\in A} f^i(v,w) & = & 0. \\
\text{For all } 1 \le i \le h\text{:} & \displaystyle\sum_{(u,t_i)\in A} f^i(u,t_i) & = & 1. \\
\text{For all } 1 \le i \le h\text{:} & \displaystyle\sum_{(v_i,u)\in A} f^i(v_i,u) & = & 1. \\
\text{For all } 1 \le i \le h\text{:} & \displaystyle\sum_{(u,v)\in A} f^i(u,v)\lambda_{uv} & \le & 2L. \\
\text{For all } v \in V - \{t_1, \ldots, t_h\}\text{:} & \displaystyle s_v \cdot \sum_i \sum_{u\in V} \left(f^i(u,v) + f^i(v,u)\right) & \le & 3\Delta. \\
\text{For all } 1 \le i \le h\text{:} & f^i(u,v) & \ge & 0.
\end{array}
$$

We now show that the set of paths guaranteed by Corollary 3 with respect to T^* can be modified so as to obtain an integral solution for the linear program as follows. If $|U'|$ is even, the solution is obtained by using each path connecting a pair (u_i, u_j) to ship one unit of flow from u_j through u_i to t_j, and another unit of flow from u_i through u_j to t_i. The length of each path is bounded by $2L_{T^*}$, and since we use each path twice, the generalized degree is bounded by $2\Delta_{T^*}$. If $|U'|$ is odd, the solution is obtained by using each of the $\frac{1}{2}(|U'|-1)$ paths connecting the first $|U'|-1$ nodes of U' twice (once in each direction), and using the path connecting the last node in U' to its mate to ship flow out of this node. The length of each path is still bounded by $2L_{T^*}$. However, because of the additional path, the degree is only bounded by $3\Delta_{T^*}$.

It follows that the value of the objective function for this solution is $\Delta_{T^*} + L_{T^*}$, and thus the linear program is guaranteed to find a solution whose value is upper bounded by this value. Let L_T and Δ_T denote the values of the length and congestion in the optimal solution of the above linear program.

The optimal solution is a "fractional" solution in the sense that the (unit) flow of each commodity is split between several flow paths. We round the fractional solution into an integral solution using an algorithm proposed by Srinivasan and Teo [23]. This algorithm builds on a theorem proved by Karp et al. [15]. For completeness, and

Downloaded 01/21/20 to 129.177.122.70. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

since the details are slightly different, we now describe the rounding of the fractional solution.

THEOREM 4 (see [15]). *Let A be a real valued $r \times s$ matrix, and y be a real valued s-vector. Let b be a real valued vector such that $Ay = b$. Let t be a positive real number such that in every column of A,*

(1) *the sum of all positive entries $\leq t$; and*

(2) *the sum of all negative entries $\geq -t$.*

Then, we can compute (in polynomial time) an integral vector $\bar{y}$ such that for every i, either $\bar{y}_i = \lfloor y \rfloor$ or $\bar{y}_i = \lceil y \rceil$ and $A\bar{y} = \bar{b}$, where $\bar{b}_i - b_i < t$.

We now show how to find an integral flow of congestion at most $6\Delta_T + 4L_T$, where each flow path (of each commodity) has length at most $4L_T$. We first decompose the flow into (polynomially many) flow paths. If any path in this decomposition is longer than $4L_T$, we discard it. We observe that since the average length is less than $2L_T$, discarding these long paths leaves at least half of a unit of flow between each pair (v_i, t_i). We scale the flows appropriately such that the total flow to each t_i is exactly 1. This can at most double the flow on an edge, and the total congestion is now at most $6\Delta_T$.

Let $P_1, P_2, \ldots$ denote the length-bounded flow paths. Denote the set of nodes in a path P_i by $V(P_i)$ and the set of edges by $E(P_i)$. Let $f(P_i)$ denote the amount of flow pushed on path P_i. Define the set $\mathcal{P}^j$ as the set of all paths that carry flow of the jth commodity. Observe that each path belongs to exactly one $\mathcal{P}^j$. The linear system $Ay = b$ needed for Theorem 4 is defined by the following linear equations, where the ith equation corresponds to the ith row of A and the ith element of b.

$$\begin{array}{llcl} \text{For each } v & s_v \cdot \sum\limits_{i:\, v \in V(P_i)} f(P_i) & \leq & 6\Delta_T; \\ \text{for all } j & -4L_T \cdot \sum\limits_{i:\, P_i \in \mathcal{P}^j} f(P_i) & = & -4L_T. \end{array}$$

The second set of inequalities captures the fact that the flow on all the paths corresponding to commodity j is exactly 1. Now the sum of the positive entries in a column is

$$\sum_{v \in V(P_i)} s_v \leq \sum_{(v,w) \in E(p_i)} \lambda_{vw} + s_{t_j} = (\text{length of path } P_j) \leq 4L_T \ .$$

The second part of the inequality follows since $s_v \leq \lambda_{vw}$ for all v, w and $s_{t_j} = 0$. The sum of the negative entries in a column is at most $4L_T$; this follows due to the fact that each P_i belongs to exactly one $\mathcal{P}^j$. Invoking Theorem 4 gives us a set of paths such that

$$\begin{array}{llcl} \text{for each } v & s_v \cdot \sum\limits_{i:\, v \in V(P_i)} \bar{f}(P_i) & < & 6\Delta_T + 4L_T; \\ \text{for all } j & -4L_T \cdot \sum\limits_{i:\, P_i \in \mathcal{P}^j} \bar{f}(P_i) & < & 0. \end{array}$$

The second set of inequalities implies that each commodity has at least one flow path. So we have a set of flow paths such that the congestion is at most $6\Delta_T + 4L_T$ and their length is at most $4L_T$. Since $\Delta_T + L_T \leq \Delta_{T^*} + L_{T^*}$ these paths satisfy the length and congestion properties as desired.

Downloaded 01/21/20 to 129.177.122.70. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

3.2. Finding a spider decomposition.

We now show how to obtain a spider decomposition satisfying the spider properties previously defined. Recall that we are now given a set of paths connecting each terminal u_j with another terminal $Mate(u_j)$, and that this set of paths satisfies the length and congestion properties.

We find a set of at least $|U'|/2$ *trees* that satisfy the following properties which are similar to the spider properties.

- Each tree spans at least two terminals from U'.
- The diameter of each tree is at most $4L_T \leq 4 \cdot (\Delta_{T^*} + L_{T^*})$.
- The generalized degree of each node in each of the trees is at most $6\Delta_T + 4L_T \leq 6(\Delta_{T^*} + L_{T^*})$.

Before showing how to find these trees, we show how to transform them into the required spiders. Repeatedly, consider the tree edges, and remove a tree edge if it separates the tree into two subtrees such that either both subtrees contain at least two terminals or one of them contains no terminals. (In this case this subtree is removed as well.) Repeat this process until no more edges can be removed. The process terminates since the number of edges is finite. Observe that upon termination if a connected component is not a spider, then another edge could be deleted. Thus, we get the following claim.

CLAIM 5. *When the process terminates, each connected component is a spider.*

Clearly, all the terminals spanned by the trees are also spanned by the spiders. The diameter of each of these spiders is at most $4L_T$, since the distance between every pair of nodes in U' spanned by a tree is at most $4L_T$ to begin with. Also, the generalized degree of the "center" of the spider is at most the generalized degree of its originating tree since we have not introduced any new edges in the process. We conclude that the spiders satisfy the desired spider properties.

Now, we show how to find the required trees. Define G_p to be the undirected graph induced by the paths from each terminal to its mate. Observe that a spanning forest of this graph may not satisfy the required diameter property above, and hence some extra refinement is necessary.

For each node u in G_p, find a unique terminal in U' that is closest to u (with respect to the lengths λ_{xy} associated with each link (x, y)). Ties are broken arbitrarily.

We modify the paths starting at each terminal as follows. From each terminal u begin tracing the path connecting u to $Mate(u)$. At some node v along this path, the closest terminal to v will not be u. We are guaranteed to encounter such a node because the closest node to $Mate(u)$ is $Mate(u)$ itself. From this node v trace the path to its closest terminal. This creates a path from u to another terminal denoted $NewMate(u)$. Note that $NewMate(u)$ may be different from $Mate(u)$. However, we are guaranteed that the path from u to $NewMate(u)$ is not longer than the path from u to $Mate(u)$ and thus bounded by $4L_T$.

Define an auxiliary directed graph H on the set of terminals U' with the set of edges $(u \rightarrow NewMate(u))$ for $u \in U'$. By definition each node in H has outdegree one. Thus, each connected component of (the undirected skeleton of) H contains exactly one directed cycle. Discard one edge from each such connected component to make it a rooted tree in which all edges are oriented towards the root. (The root is unique since the outdegree of each node is one.) Note that every nontrivial strongly connected component of H is a cycle. Thus, this can be done just by discarding an arbitrary edge from each strongly connected component of H. Let H' be the resulting forest.

Define the level of a node in H' to be the number of edges in the path to it from the root of its component. (We flip the direction of the edges in H' for the purpose

Downloaded 01/21/20 to 129.177.122.70. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

of measuring distances.) Distinguish between nodes of even level and nodes of odd level. Each edge of H' goes either from an odd-level node to an even-level node or vice-versa.

Consider two collections of stars in H'. One collection consisting of edges from odd-level nodes to even-level nodes, and the other consisting of edges from even-level nodes to odd-level nodes. Every terminal with positive indegree and outdegree (in H') is spanned by a star in each one of the two collections. Every terminal with either indegree or outdegree zero (in H') is spanned by a star in only one of the two collections. However, by a simple pigeonhole argument, at least one of the collections spans at least half of the terminals.

Consider such a collection. First, note that each star in this collection induces an undirected tree in the original graph when replacing each star edge by its originating path. We now claim the following.

LEMMA 6. *The induced trees of any two stars belonging to the same collection are node disjoint.*

Proof. To obtain a contradiction assume they are not disjoint. Then there exist two distinct terminals with the same even or odd parity, say, u and v, such that $NewMate(u) \neq NewMate(v)$, but the paths traced from u to $NewMate(u)$ and from v to $NewMate(v)$ have a common node x. Consider the terminal chosen by x as its closest terminal. We distinguish between two cases.

Case 1. The terminal chosen by x is u. Then u must be $NewMate(v)$, contradicting the fact that u and v are of the same parity. The case where v is the chosen terminal of x is symmetric.

Case 2. The terminal chosen by x is $NewMate(u)$. Then $NewMate(v)$ must be the same as $NewMate(u)$, a contradiction. The case where $NewMate(v)$ is the chosen terminal of x is symmetric. □

It is easy to see that the trees induced by the stars in the collection satisfy the required properties. This concludes the construction.

4. Hardness of approximations.

In this section we show that the best possible approximation factor of the minimum broadcast time in the heterogeneous postal model is $3 - \epsilon$. We show this hardness result even for a restricted model in which $s_i \in \{0, 1\}$ and $\lambda_{uv} \in \{1, d\}$ for some constant d. Note that when $s_i = 0$, node u_i can broadcast the message concurrently to all of its neighbors. The proof is by a reduction to the set cover problem. In the unweighted version of the set cover problem we are given a set U of elements and a collection S of subsets of U. The goal is to find the smallest number of subsets from S whose union is the set U. Feige [9] proved the following hardness result.

LEMMA 7. *Unless $NP \subseteq DTIME(n^{\log\log n})$, the set cover problem cannot be approximated by a factor which is better than $\ln n$, and hence it cannot be approximated within any constant factor.*

In our proof, we will only use the fact that it is NP-hard to approximate the optimal set cover within any constant factor.

THEOREM 8. *It is NP-hard to approximate the minimum broadcast time of any graph within a factor of $3 - \epsilon$.*

Proof. Assume to the contrary that there exists an algorithm that violates the claim of the theorem for some ϵ. We show how to approximate the set cover problem within a constant factor using this algorithm.

To approximate the set cover problem we "guess" the size of the optimal set cover and use our approximate minimum broadcast time algorithm to check our guess. Since

Downloaded 01/21/20 to 129.177.122.70. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

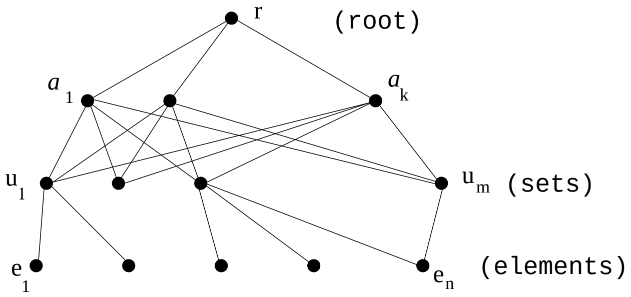

FIG. 1. *The graph G.*

the size of the optimal set cover is polynomial, we need to check only a polynomial number of guesses.

Consider an instance of set cover $I = (U, S)$, where U is the set of elements, and S a collection of subsets of U. Let $|U| = n$ and $|S| = m$. Let the guess on the size of the optimal set cover be k. We construct the following graph G (see Figure 1). The graph G consists of $1 + n + m + k$ vertices: a distinguished root vertex r, vertices $e_1, \ldots, e_n$ corresponding to the elements of U, vertices $u_1, \ldots, u_m$ corresponding to the subsets, and k additional vertices $a_1, \ldots, a_k$.

The root r has switching time $s(r) = 0$ and is connected to $a_1, \ldots, a_k$ by edges with delay $\lambda_{ra_\ell} = 1$. Each vertex a_ℓ has switching time $s(a_\ell) = 1$ and is connected to all u_j with delay $\lambda_{a_\ell u_j} = 1$. Each vertex u_j has switching time $s(u_j) = 0$ and is connected to a vertex e_i iff the jth set contains the ith element. The delay of such an edge is $\lambda_{u_j e_i} = d$, where $d > \frac{4-2\epsilon}{\epsilon}$ is a constant. Each vertex e_i has switching time $s(e_i) = 1$. Finally, to complete the instance of the multicasting problem, the target multicast set consists of all vertices e_i.

We first show that if there is a set cover of size k, then there is a multicast scheme of length $d + 2$. After time 1, all the vertices a_ℓ receive the message. After time 2, all the vertices u_j corresponding to sets which are in this cover receive the message. This is possible since all a_ℓ are connected to all u_j. Finally, these vertices send the message to all the elements that they cover. Since $s(u_j) = 0$ and $\lambda_{u_j e_i} = d$, it follows that the multicast time is $d + 2$.

Suppose that the algorithm for the multicasting problem completes the multicasting at time t. By the contradiction assumption, its approximation factor is $3 - \epsilon$. Since by our guess on the size of the set cover the optimal multicast time no more than $d+2$, we have $t \leq (3-\epsilon)(d+2) = 3d+6-2\epsilon-\epsilon d < 3d+6-2\epsilon-4+2\epsilon = 3d+2$. The strict inequality follows from the choice of d.

We first claim that all the vertices u_j that participate in the multicast receive the message from some a_ℓ. Otherwise there exists a vertex $e_{i'}$ that received the message via a path of a type $(r, a_\ell, u_j, e_i, u_{j'}, e_{i'})$. This means that $e_{i'}$ received the message at or after time $3d + 2 > t$. Our second claim is that each vertex a_ℓ sends the message to at most $2d$ vertices u_j. This is because the $(2d + 1)$st vertex would receive the message at time $2d + 2$ and would not be able to help in the multicast effort that is completed before time $3d + 2$.

Combining our two claims we get that the multicasting was completed with the help of $2dk$ vertices u_j. The corresponding $2dk$ sets cover all the elements e_i. This violates Lemma 7, which states that the set cover problem cannot be approximated within any constant factor. □

Downloaded 01/21/20 to 129.177.122.70. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Remark. In our proof we considered a restricted model in which the switching time may get only two possible values and the delay may get only three possible values (assuming that when an edge does not exist, then the delay is infinity). Observe that this hardness result does not apply to the telephone model, in which the switching time is always 1 and the delay is either 1 or infinity. We have similar hardness results for other special cases. However, none of them is better than 3 and all use similar techniques.

Acknowledgment. We thank David Williamson for his helpful comments.

REFERENCES

[1] V. Bala, J. Bruck, R. Bryant, R. Cypher, P. deJong, P. Elustondo, D. Frye, A. Ho, C. T. Ho, G. Irwin, S. Kipnis, R. Lawrence, and M. Snir, *The IBM external user interface for scalable parallel systems*, Parallel Comput., 20 (1994), pp. 445–462.

[2] V. Bala, J. Bruck, R. Cypher, P. Elustondo, A. Ho, C. T. Ho, S. Kipnis, and M. Snir, *CCL: A portable and tunable collective communication library for scalable parallel computers*, in Proceedings of the Eighth International Parallel Processing Symposium, Cancun, Mexico, IEEE Press, Piscataway, NJ, 1994, pp. 835–844.

[3] A. Bar-Noy and S. Kipnis, *Designing broadcasting algorithms in the Postal model for message-passing systems*, Math. Systems Theory, 27 (1994), pp. 431–452.

[4] A. Bar-Noy, S. Kipnis, and B. Schieber, *Optimal multiple message broadcasting in telephone-like communication systems*, in Proceedings of the Sixth Symposium on Parallel and Distributed Processing, Dallas, TX, IEEE Press, Piscataway, NJ, 1994, pp. 216–223.

[5] E. Cockayne and A. Thomason, *Optimal multi-message broadcasting in complete graphs*, in Proceedings of the Eleventh SE Conference on Combinatorics, Graph Theory, and Computing, 1980, pp. 181–199.

[6] M. J. Dinneen, M. R. Fellows, and V. Faber, *Algebraic construction of efficient networks*, Applied Algebra, Algebraic Algorithms, and Error Correcting Codes (AAECC), Lecture Notes in Comput. Sci. 539, Springer, Berlin, 1991, pp. 152–158.

[7] *Express User's Guide, Version* 3.0, Parasoft Corporation, Monrovia, CA, 1995.

[8] A. M. Farley, *Broadcast time in communication networks*, SIAM J. Appl. Math., 39 (1980), pp. 385–390.

[9] U. Feige, *A threshold of* $\ln n$ *for approximating set cover*, in Proceedings of the 28th Symposium on Theory of Computing (STOC), Association for Computing Machinery, New York, 1996, pp. 314–318.

[10] 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.

[11] 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.

[12] L. Gargano and U. Vaccaro, *On the construction of minimal broadcast networks*, Networks, 19 (1989), pp. 373–389.

[13] M. Grigni and D. Peleg, *Tight bounds on minimum broadcast networks*, SIAM J. Discrete Math., 4 (1991), pp. 207–222.

[14] S. M. Hedetniemi, S. T. Hedetniemi, and A. L. Liestman, *A survey of gossiping and broadcasting in communication networks*, Networks, 18 (1988), pp. 319–349.

[15] R. M. Karp, F. T. Leighton, R. L. Rivest, C. D. Thompson, U. V. Vazirani, and V. V. Vazirani, *Global wire routing in two-dimensional arrays*, Algorithmica, 2 (1987), pp. 113–129.

[16] G. Kortsarz and D. Peleg, *Approximation algorithm for minimum time broadcast*, SIAM J. Discrete Math., 8 (1995), pp. 401–427.

[17] 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.

[18] A. L. Leistman and J. G. Peters, *Broadcast networks of bounded degree*, SIAM J. Discrete Math., 1 (1988), pp. 531–540.

[19] M. Middendorf, *Minimum broadcast time is NP-complete for* 3*-regular planar graphs and deadline* 2, Inform. Process. Lett., 46 (1993), pp. 281–287.

[20] 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. 202–213.

[21] D. Richards and A. L. Leisterman, *Generalizations of broadcasting and gossiping*, Networks, 18 (1988), pp. 125–138.

[22] M. Snir, S. Otto, S. Huss-Lederman, D. Walker, and J. Dongarra, *MPI—The Complete Reference, Vol.* 1*, The MPI*-1 *Core*, 2nd ed., MIT Press, Cambridge, MA, 1998.

[23] A. Srinivasan and C.-P. Teo, *A constant-factor approximation algorithm for packet routing, and balancing local vs. global criteria*, in Proceedings of the 29th Symposium on Theory of Computing (STOC), Association for Computing Machinery, New York, 1997, pp. 636–643.

[24] J. A. Ventura and X. Weng, *A new method for constructing minimal broadcast networks*, Networks, 23 (1993), pp. 481–497.

[25] D. B. West, *A class of solutions to the gossip problem, Part* I, Discrete Math., 39 (1992), pp. 307–326.

Downloaded 01/21/20 to 129.177.122.70. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php