On Multicast Algorithms for Heterogeneous Networks of Workstations*

R. Libeskind-Hadas J. Hartline P. Boothe G. Rae J. Swisher

Department of Computer Science
Harvey Mudd College
Claremont, California 91711

{hadas, jhartlin, pboothe, grae, jswisher}@cs.hmc.edu

*This work was supported by the National Science Foundation under grant CCR-9900491.

Running Head: Multicast in Heterogeneous Networks of Workstations

Communicating Author:

Ran Libeskind-Hadas
Department of Computer Science
Harvey Mudd College
301 E. 12th Street
Claremont, CA 91711

Phone: (909)-621-8976
e-mail: hadas@cs.hmc.edu

Abstract

Networks of workstations (NOWs) provide an economical platform for high performance parallel computing. Such networks may comprise a variety of different types of workstations and network devices. This paper addresses the problem of efficient multicast in a heterogeneous communication model. Although the problem of finding optimal multicast schedules is known to be NP-complete in this model [7], a greedy algorithm [1, 7] has been shown experimentally to find good solutions in practice [1]. In this paper we show that the greedy algorithm finds provably near-optimal schedules in polynomial time and that optimal schedules can be found in polynomial time when the number of distinct types of workstations is bounded by a constant.

Specifically, this paper presents three results. First, when there are n workstations of some constant k distinct types, the greedy algorithm is shown to find schedules that complete at most a constant additive term later than optimal. Second, an algorithm is given that finds optimal schedules in time $O(n^{2k})$. Finally, it is shown that for the general problem, the greedy algorithm finds solutions that complete the multicast in at most twice the optimal time.

Index terms: Networks of workstations, heterogeneous networks, multicast communication, approximation algorithms, dynamic programming.

1 Introduction

A number of communication models have been proposed to characterize the latencies incurred in message passing systems in general and networks of workstations (NOWs) in particular. For example, in the *one-port model* [9], a node x takes one unit of time to send a message to any node y. After one unit of time has elapsed, node x and node y both have copies of the message. Therefore, during the second unit of time, x and y can concurrently send the message to other destination nodes. In this model, all communication takes place at unit time steps. More sophisticated communication models have recently been proposed to capture other parameters that contribute to the communication latency in message passing systems. Among these are the *postal model* [2], the *LogP model* [5], and extensions of these models [12].

Given a communication model and a *multicast set* S containing a source node and destination nodes, the objective of the *multicast scheduling problem* is to find a communication schedule that minimizes the time until all destination nodes have received the message. Such a schedule is known as an *optimal multicast schedule*. For example, in the one-port model a simple *recursive doubling algorithm* [4, 6, 9, 11] can be used to find an optimal multicast schedule. Optimal multicast algorithms are also known for several other *homogeneous* communication models in which all nodes are assumed to have identical latencies [2, 10, 12]. In the case of NOWs, the constituent workstations, networking devices, and communication protocols are frequently heterogeneous, resulting in varying computation and communication speeds. Thus, new communication models and multicast algorithms are required.

Banikazemi et al. [1] and Hall et al. [7] have independently proposed a *heterogeneous node model* which associates a single latency parameter, called the *message initiation cost*, with each node in the network. This cost accounts for the overhead involved in preparing the message for transmission. In this model a node x incurs its message initiation cost $c(x)$ to send the message to any destination node, y. At time $c(x)$, node y receives the message and may begin sending the message to another node, incurring its message initiation cost $c(y)$. Concurrently, node x may send the message to another node, again incurring its message initiation cost $c(x)$. Because wormhole or virtual cut-through routing are typically used in such networks, the network latency is largely independent of the location of the destination node in the network. Thus, the network latency can be incorporated into the message initiation cost. Although Hall et al. have shown that the problem of finding optimal multicasts is NP-complete in this model, Banikazemi et al. have shown experimentally that a greedy algorithm often finds near-optimal schedules.

In related work, Bhat et al. have proposed an alternative model that accounts for heterogeneity

in both the nodes and the network [3]. This model is particularly well-suited for wide-area networks where network latencies over "long haul" links may be very different from those within a local area network. Itkis et al. have studied multicasting in a model in which all nodes are identical but a node may select one of several different "services" each time it sends a message, where each service has an associated latency and price [8].

In this paper we show that provably near-optimal multicast schedules can be found in polynomial time. In particular, we begin by considering the case that the number of distinct types of workstations is some constant k. In this case, we show that the greedy algorithm [1, 7] finds multicasts schedules in time $O(n \log n)$ that are at most a constant additive term larger than optimal. We also show that an optimal multicast schedule can be found in time $O(n^{2k})$. We then use these results to show that for the general problem in which there are an arbitrary number of types of workstations, the $O(n \log n)$ greedy algorithm finds solutions that are no worse than twice optimal. Extensions to other possible heterogeneous communication models are discussed in the last section.

2 Preliminaries

A *multicast set* is a set S comprising a source node and one or more destination nodes. For each node $v \in S$, $c(v)$ denotes the message initiation cost of node v. A *multicast schedule* for S is a directed tree T with one vertex for each node in S. The root of T corresponds to the source node and all remaining vertices correspond to destination nodes. Henceforth, we use "node" and "vertex" interchangeably. Each non-root vertex v has exactly one incoming edge representing transmission of the message to v. Each non-leaf vertex v has one or more outgoing edges corresponding to transmissions of the message from v to other destination nodes. These edges are ordered from left to right to indicate the order, from first to last, in which v transmits the message to its children. Alternatively, we say that a list $(w_1, \ldots, w_\ell)$ is the *arrival ordered* list of children of v if v sends the message to node w_i before sending to node w_{i+1}, $1 \leq i < \ell$.

The *arrival time* for node v, denoted $t(v)$, is the time at which the message arrives at node v. The arrival time of the message at the root is, by definition, 0. For each non-leaf node v with the arrival ordered list of children $(w_1, \ldots, w_\ell)$, $t(w_i) \geq t(v) + i \times c(v)$, $1 \leq i < \ell$. The *initiation time* for the pair v, w_i is defined to be $t(w_i) - c(v)$; the time at which node v began incurring the message initiation cost for the message to be sent to node w_i. Node v is said to *initiate* the message to node w_i at time $t(w_i) - c(v)$. The *completion time* of a schedule T is $\max_{v \in T} t(v)$, the earliest time at which all nodes have received the message. In some cases it will be convenient to label schedules T^* where $*$ is some string of symbols. In such cases, $t^*(x)$ denotes the arrival time of node x in

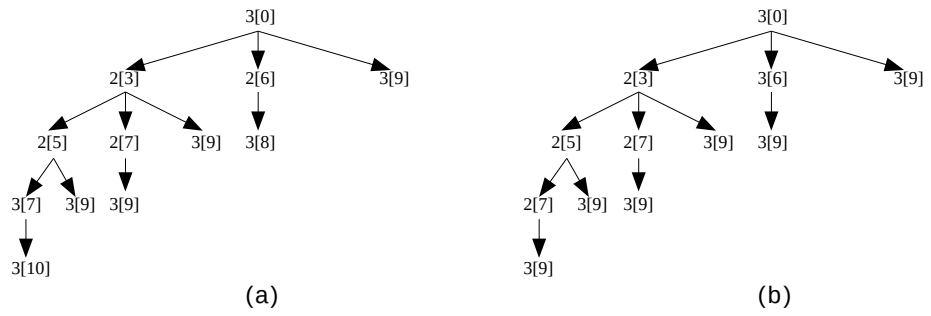

Figure 1: Two schedules for a source with initiation cost 3, four destinations with initiation cost 2, and seven destinations with initiation cost 3. Each node v is labelled with the pair $c(v)[t(v)]$ indicating the message initiation cost and arrival time, respectively. (a) Schedule produced by the greedy algorithm with completion time 10. (b) A schedule with completion time 9.

this schedule.

The greedy algorithm proposed independently by Banikazemi et al. [1] and by Hall et al. [7] is performed by the source node as follows: Given a multicast set S, let multicast schedule T initially contain only the source node $s \in S$ as the root. Let $S' = S - \{s\}$. Identify a node $v \in T$ that can complete transmission of the message as early as possible, breaking ties arbitrarily. Among all nodes in S' identify one with minimum message initiation cost, breaking ties arbitrarily. Let w denote such a node. Node w is now removed from S' and included in T as the next child of v. The process is repeated until S' is empty.

Lemma 1 *The running time of the greedy algorithm for a multicast set of n nodes is $O(n \log n)$.*

Proof: The algorithm requires that the $n-1$ destination nodes first be sorted in non-decreasing order of message initiation cost. This can be done in $O(n \log n)$ time. The nodes in schedule T can be maintained in a priority queue in which the key associated with each element in the priority queue is the earliest time at which the node can next complete transmission of the message. Initially, the source node s is inserted into an empty priority queue with the key equal to $c(s)$ since this is the earliest time that the source node can complete transmission of the message. At each iteration of the algorithm, the node v with the smallest key is removed from the priority queue. Let t denote the value of the key for node v. The next node $w \in S'$ is now inserted into the priority queue with key equal to $t + c(w)$. Next, node v is reinserted into the priority queue with key equal to $t + c(v)$. The process is repeated $n-1$ times. By using a heap to implement the priority queue, each deletion and pair of insertions performed per iteration can be accomplished in $O(\log n)$ time. Thus the total running time is $O(n \log n)$. □

Although the greedy algorithm is intuitively appealing, it is known to produce non-optimal multicast schedules [1, 3]. Figure 1 depicts two multicast schedules for the case of a source with

initiation cost of 3 and eleven destination nodes of which four destinations have message initiation costs of 2 and the remaining seven destinations have message initiation costs of 3. Each node v is labelled with the pair $c(v)[t(v)]$ indicating the message initiation cost and arrival time, respectively. Figure 1(a) shows the schedule found by the greedy algorithm, completing at time 10. Figure 1(b) shows an alternate schedule that completes at time 9.

A node v with arrival ordered list of children $(w_1, \ldots, w_\ell)$ is said to *idle* if $t(w_i) > t(v) + i \times c(v)$ for some i, $1 \leq i \leq \ell$. A multicast schedule is said to be *non-idling* if it contains no idle nodes. Clearly, for any schedule T with one or more idle nodes, the schedule T' constructed by removing the idle times from T has completion time no larger than that of T. Unless explicitly stated otherwise, all schedules are henceforth assumed to be non-idling.

A multicast schedule T is said be *monotonic* if there does not exist a slow non-root node in T that sends the message to a faster node. That is, a schedule T is monotonic if for every $u, v \in T$ such that u is not the root of the tree, if (u, v) is a directed edge in T then $c(u) \leq c(v)$. Note that by definition, every schedule produced by the greedy algorithm is monotonic. We now show that for every schedule T, there exists a corresponding monotonic schedule T' such that the completion time of T' is no larger than that of T.

Lemma 2 *For every schedule T for multicast set S there exists a monotonic schedule T' for S such that the completion time of T' is no larger than the completion time of T.*

Proof: Let (u, v) be a directed edge in T such that u is not the root of the tree and $c(u) > c(v)$. Let $\alpha = (\alpha_1, \ldots, \alpha_a)$ denote the arrival ordered list of children of u that receive the message before v, let $\beta = (\beta_1, \ldots, \beta_b)$ denote the arrival ordered list of children of u that receive the message after v, and let $\gamma = (\gamma_1, \ldots, \gamma_c)$ denote the arrival ordered list of children of v. Consider the transformation shown in Figure 2. The thin edges represent a transmission to a single node while the thick edges indicate zero or more transmissions to an arrival ordered list of nodes. In this transformation, the subtree of T rooted at u is modified so that the parent of u now sends to v. Node v then sends the message to each of the elements of α, followed by u, followed by the elements of γ. Transmissions from nodes in α, β, and γ are unaltered as are all other transmissions in the schedule.

Let T' denote the schedule constructed by performing this transformation on schedule T. We now show that the completion time of schedule T' is less than or equal to that of T. Note that $t'(v) = t(u)$. Next, each node $\alpha_i \in \alpha$ receives the message earlier in T' than in T since

$$t'(\alpha_i) = t'(v) + i \times c(v) < t(u) + i \times c(u) = t(\alpha_i).$$

Similarly, for each $\beta_i \in \beta$,

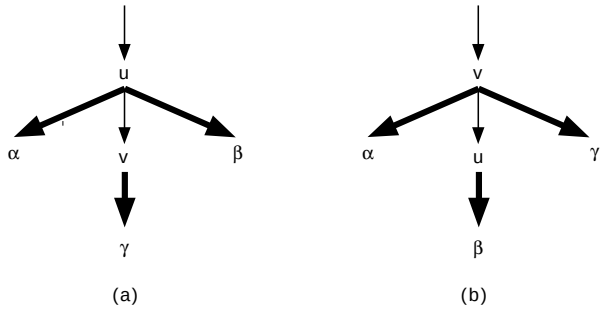

Figure 2: Transformation in Lemma 2 swapping the order of u and v in the schedule. Thick edges indicate zero or more transmissions to an arrival ordered list of nodes. (a) The subtree rooted at u before the transformation. (b) The subtree after the transformation.

$$t'(\beta_i) = t'(v) + (|\alpha| + 1) \times c(v) + i \times c(u) < t(u) + (|\alpha| + 1) \times c(u) + i \times c(u) = t(\beta_i).$$

And, for each $\gamma_i \in \gamma$,

$$t'(\gamma_i) = t'(v) + (|\alpha| + 1) \times c(v) + i \times c(v) < t(u) + (|\alpha| + 1) \times c(u) + i \times c(v) = t(\gamma_i).$$

Although $t'(u) > t(u)$, we observe that

$$t'(u) = t'(v) + (|\alpha| + 1) \times c(v) < t(u) + (|\alpha| + 1) \times c(u) = t(v).$$

Thus, $t'(u)$ is less than the completion time of schedule T. Since all other nodes receive the message in T' at least as early as in T, the completion time of T' is no larger than that of T.

Each application of this transformation strictly reduces the number of pairs of nodes $x, y \in T$ such that x is a non-root ancestor of y but $c(x) > c(y)$. Thus, a a finite number of applications of the transformation yields a schedule in which no non-root node sends the message to a faster node, resulting in a monotonic schedule whose completion time is no larger than that of the original schedule. □

Without loss of generality, we henceforth restrict our attention to monotonic schedules. We now examine another property of multicast schedules. A multicast schedule T is said be *layered* if there does not exist a slow non-root node in T that receives the message earlier than a faster non-root node. That is, a schedule T is layered if for every pair of non-root nodes $u, v \in T$ if $c(u) < c(v)$ then $t(u) \leq t(v)$. Note that by definition, every schedule produced by the greedy algorithm is layered. Unlike the monotonic property, it is not the case that every schedule has a corresponding layered schedule with the same or smaller completion time. However, we now give a lemma and a corollary which show that for any multicast set, the greedy algorithm produces schedules with the smallest completion time among all layered schedules.

Lemma 3 *Let $S = \{s_0, \ldots, s_{n-1}\}$ and $S' = \{s'_0, \ldots, s'_{n-1}\}$ be two multicast sets with sources s_0 and s'_0, respectively. Let $c(s_i) \leq c(s_{i+1})$ and $c(s'_i) \leq c(s'_{i+1})$, $1 \leq i \leq n-2$, and $c(s_i) \leq c(s'_i)$, $0 \leq i \leq n-1$. Let T denote a schedule for S found by the greedy algorithm and let T' denote any layered schedule for S'. Then the completion time of T is no larger than the completion time of T'.*

Proof: Let $t(s_i)$ and $t'(s'_i)$ denote the arrival times of the message at nodes s_i and s'_i, $0 \leq i \leq n-1$, in schedules T and T', respectively. Since nodes with the same message initiation costs can be interchanged without affecting arrival times in the schedule, without loss of generality if $c(s_i) = c(s_j)$ and $i < j$, then $t(s_i) \leq t(s_j)$. Similarly, if $c(s'_i) = c(s'_j)$ and $i < j$, then $t'(s'_i) \leq t'(s'_j)$. Then, since T and T' are layered, $t(s_i) \leq t(s_{i+1})$ and $t'(s'_i) \leq t'(s'_{i+1})$, $1 \leq i \leq n-2$.

By way of contradiction, assume that the completion time of schedule T' is less than that of T. Let $0 \leq j \leq n-1$ be the smallest index such that $t'(s'_j) < t(s_j)$. Note that $j \geq 1$ since $t(s_0) = t'(s_0) = 0$ and that $t(s_i) \leq t'(s'_i)$, $0 \leq i \leq j-1$. In T, nodes $s_0, \ldots, s_{j-1}$ are the only nodes that can possibly receive the message strictly before time $t(s_j)$ and thus by time $t'(s'_j)$ in particular. Therefore, in T, nodes $s_0, \ldots, s_{j-1}$ collectively complete at most $j-1$ message transmissions by time $t'(s'_j)$. Let k denote the number of message transmissions in T' collectively completed by nodes $s'_0, \ldots, s'_{j-1}$ by time $t'(s'_j)$. Since node s'_j receives the message at time $t'(s'_j)$ from some s'_i, $1 \leq i \leq j-1$, nodes $s'_0, \ldots, s'_{j-1}$ collectively complete at least j message transmissions by time $t'(s'_j)$ and thus $k \geq j$. However, since $t(s_i) \leq t'(s'_i)$ and $c(s_i) \leq c(s'_i)$, $1 \leq i \leq j-1$, nodes $s_0, \ldots, s_{j-1}$ in T collectively have at least k points at which message transmissions can be completed before time $t'(s'_j)$. Since the greedy algorithm performs message transmissions as early as possible, $s_0, \ldots, s_{j-1}$ collectively complete at least k message transmissions by time $t'(s'_j)$. Since $k > j-1$, this is a contradiction. □

Corollary 1 *For any multicast set S, the schedule produced by the greedy algorithm has the minimum completion time among all layered schedules for S.*

Proof: Follows immediately from Lemma 3 by letting $S = S'$. □

3 Multicast for Limited Heterogeneity

Hall et al. [7] have shown that the optimal multicast problem is NP-complete for the heterogeneous node model. In practice, the number of distinct types of workstations in a heterogeneous NOW may be small although the number of workstations may be large. In this section we investigate networks in which there are an arbitrary number of nodes, n, but a limited number, k, of different types of workstations. We show that in this case the greedy algorithm constructs provably near-optimal

schedules. Moreover, optimal schedules can be found in polynomial time where the polynomial depends on k.

Let $C(i)$ denote the message initiation cost for a node of type i, $1 \leq i \leq k$, such that $C(i) < C(j)$ for $1 \leq i < j \leq k$. We show that the greedy algorithm constructs schedules with completion times that are at most $\sum_{i=1}^{k-1} C(i)$ larger than optimal. In addition, we give an algorithm that finds optimal solutions in time $O(n^{2k})$ where n is the number of nodes in the network.

We begin by considering the case that $k = 1$. In this case, the heterogeneous node model reduces to the standard one-port model [9] in which all nodes send messages in synchronized steps. The aforementioned recursive doubling algorithm is known to be optimal for this model [4]. We observe that the greedy algorithm is equivalent to the recursive doubling algorithm in this case since it ensures that at each time step all nodes with the message send to nodes which have not yet received the message. Thus, the greedy algorithm is optimal for $k = 1$.

For $k = 2$ the greedy algorithm is no longer optimal as demonstrated in the example in Figure 1. We next show that for any multicast set, the greedy algorithm constructs a schedule whose message completion time is at most $\sum_{i=1}^{k-1} C(i)$ larger than the completion time of the optimal schedule. For example, in the case that $k = 2$, this implies that the greedy algorithm constructs a schedule whose completion time is at most $C(1)$ larger than that of an optimal schedule, where $C(1)$ is the message initiation cost of the fastest type of node in the network. The following lemma will be used to prove this result.

Lemma 4 *Let S be a multicast set, let T be a monotonic schedule for this set, and let v be a node in this schedule. Let u be a node in T with smallest value $t(u)$ such that $t(u) < t(v)$ and $c(u) > c(v)$. If there exists a positive integer ℓ such that $\ell \times c(v) \leq t(v) - t(u) \leq \ell \times c(u)$ then there exists a schedule T' for S satisfying the following properties:*

1. *$t'(u) > t'(v)$.*
2. *$t(w) = t'(w)$ for all $w \in S$ such that w is not a descendant of u or v in T.*
3. *The completion time of T' is no larger than that of T.*
4. *T' is monotonic.*

Proof: Let $\alpha = (\alpha_1, \ldots, \alpha_a)$ and $\beta = (\beta_1, \ldots, \beta_b)$ denote the arrival ordered list of children of nodes u and v, respectively, in schedule T. Let $t(v) - t(u) = q \times c(v) + r$ where q is a positive integer and $0 \leq r < c(v)$. Schedule T' is constructed from T by exchanging nodes u and v. In addition, in T' the arrival ordered list of children of u becomes $(\alpha_{q+1}, \ldots, \alpha_a)$ while the arrival ordered list of children of v becomes $(\alpha_1, \ldots, \alpha_q) \circ \beta$ where $\circ$ is the list concatenation operator. If $a \leq q$ then u has no children and v sends the message to the arrival ordered list $\alpha \circ \beta$. This

transformation is illustrated in Figure 3. The first two properties from the lemma follow by the definition of T'.

Next, we show that the completion time of T' is no larger than that of T. By definition, $\ell \leq q$ and thus $q \geq 1$. If $r > \ell \times (c(u) - c(v))$ then $t(v) - t(u) = q \times c(v) + r > \ell \times c(u) + (q - \ell) \times c(v) \geq \ell \times c(u)$ and thus $t(v) - t(u) > \ell \times c(u)$, contradicting the assumption of the lemma. Thus $r \leq \ell \times (c(u) - c(v)) \leq q \times (c(u) - c(v))$. We now examine each node whose arrival time may have changed as a result of applying the transformation. First, for $1 \leq i \leq q$, $t'(\alpha_i) = t'(v) + i \times c(v) = t(u) + i \times c(v) < t(u) + i \times c(u) = t(\alpha_i)$. For $i > q$, $t'(\alpha_i) = t'(u) + (i - q) \times c(u) = t(v) + (i - q) \times c(u)$ whereas $t(\alpha_i) = t(u) + i \times c(u)$. Thus, $t'(\alpha_i) - t(\alpha_i) = t(v) - t(u) - q \times c(u) = q \times c(v) + r - q \times c(u) = r - q \times (c(u) - c(v))$. Since $r \leq q \times (c(u) - c(v))$, this quantity is at most zero. Finally, for each $\beta_i \in \beta$, $t(\beta_i) = t(v) + i \times c(v)$ whereas $t'(\beta_i) \leq t'(v) + (q + i) \times c(v) = t(u) + (q + i) \times c(v)$, with inequality due to the possibility that $a \leq q$. Thus, $t'(\beta_i) - t(\beta_i) \leq t(u) - t(v) + q \times c(v) = -r \leq 0$. Since the nodes in α and β receive at least as early in T' as in T, the descendants of these nodes also receive at least as early in T' as in T. Nodes u and v exchange arrival times and no other nodes in T are affected by this transformation. Consequently, the completion time of T' is no larger than that of T.

Finally, we show that since T is monotonic, T' is monotonic as well. All children of v in T are also children of v in T'. Nodes $\alpha_1, \ldots, \alpha_q$ are children of u in T and children of v in T'. Since $c(v) < c(u)$, by the monotonicity of T it follows that $c(v) < c(\alpha_i)$, $1 \leq i \leq q$. The children of u in T' are a subset of the children of u in T and thus $c(u) < c(\alpha_i)$, $q + 1 \leq i \leq a$. The only remaining nodes that must be considered are p_u and p_v, the parents of nodes u and v in T, respectively. Node p_v sends to node u in T'. Since $c(p_v) \leq c(v)$ by the monotonicity of T and $c(v) < c(u)$, it follows that $c(p_v) < c(u)$. Finally, p_u, sends to v in T'. Since $t(p_u) < t(u) < t(v)$, if $c(p_u) > c(v)$ then $t(p_u) < t(v)$ and $c(p_u) > c(v)$ but $t(p_u) < t(u)$, contradicting the assumption of the lemma that u is the node with the smallest value of $t(u)$ such that $t(u) < t(v)$ and $c(u) > c(v)$. Thus, $c(p_u) \leq c(v)$ and therefore T' is monotonic. □

Theorem 1 *Let S be a multicast set with n nodes of k distinct types. Let $C(i)$ denote the message initiation cost for a node of type i with $C(i) < C(j)$ for $1 \leq i < j \leq k$. The greedy algorithm constructs a schedule whose completion time is no more than $\sum_{i=1}^{k-1} C(i)$ larger than the completion time of an optimal schedule.*

Proof: A schedule T is said to satisfy property P_i if:

1. T is monotonic and
2. For each node $u \in S$ with $c(u) < C(i)$, $t(u) \leq t(v)$ for all non-root nodes v such that $c(u) < c(v)$.

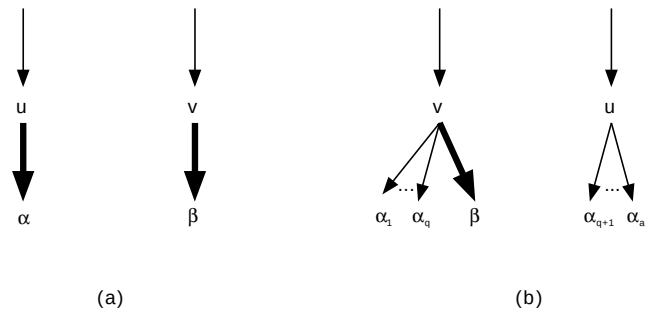

Figure 3: The transformation in Lemma 4 Thick edges indicate zero or more transmissions to an arrival ordered list of nodes. (a) Nodes u, v and their children in schedule T. (b) Nodes u, v and their children in schedule T'.

Let T^1 be an optimal monotonic schedule for S. The proof proceeds by applying a sequence of $k-1$ transformations from T^1 into schedules $T^2, \ldots, T^k$. Each schedule T^i is shown to satisfy property P_i. The completion time of T^{i+1} is also shown to be no more than $C(i)$ larger than that of T^i. Consequently, after completing the $k-1$ transformations, the final schedule T^k has completion time no more than $\Sigma_{i=1}^{k-1} C(i)$ larger than that of the optimal schedule T^1. Moreover, property P_k implies that T^k is a layered schedule Corollary 1 implies that the schedule found by the greedy algorithm completes at least as early as this layered schedule

Schedule T^1 trivially satisfies property P_1. The transformation from T^i to T^{i+1} is performed as a sequence of steps as follows: Consider the forest induced by removing from T^i the root node and all nodes with message initiation costs less than $C(i)$. Introduce temporarily the smallest possible delay in the arrival time at the root of each tree in this forest so that the arrival time becomes an integer multiple of $C(i)$. Note that each such delay is strictly smaller than $C(i)$. Denote this schedule by T. Schedule T may contain idle nodes due to the introduced delays In T, each node with message initiation cost of $C(i)$ receives the message at a time equal to an integer multiple of $C(i)$ since by the monotonicity of T^i each such node is either a root of a tree in the forest or is on a path of nodes with message initiation costs $C(i)$ rooted at a node with message initiation cost $C(i)$. Let v be a node of type i and let u be a node with smallest value $t(u)$ such that $t(u) < t(v)$ and $c(u) > c(v) = C(i)$, if such a pair of nodes exists Then $t(u)$ is also an integer multiple of $C(i)$ since by the monotonicity of T^i such a node is either the root of a tree in the forest or is at the end of a path of nodes all with message initiation costs of $C(i)$. Thus, $t(v) - t(u) = \ell \times c(v)$ for some positive integer ℓ. Since $c(u) > c(v)$, $t(v) - t(u) < \ell \times c(u)$. By Lemma 4, schedule T can be transformed into a monotonic schedule in which v receives the message earlier than u and the completion time of this new schedule is no larger than that of T Moreover, since the transformation affects only the subtrees rooted at the two exchanged nodes, only a finite number of applications of

this transformation are required until all nodes of type i receive the message at least as early as all nodes with message initiation costs larger than $C(i)$. Denote this schedule by T_1^{i+1}. By Lemma 4, T_1^{i+1} is monotonic. Moreover, the arrival times at nodes with message initiation costs less than $C(i)$ are unchanged by this transformation and nodes with message initiation costs equal to $C(i)$ now have arrival times that are less than or equal to the arrival times of nodes with larger message initiation costs. Thus, schedule T_1^{i+1} satisfies property P_{i+1}.

In T_1^{i+1} there may exist a node v of type i that receives the message at the same time as a node u with $c(u) > c(v) = C(i)$. If u and v are in different trees in the forest then the subtrees rooted at u and v are exchanged if the delay introduced at the root of the tree containing u was larger than the delay introduced at the root of the tree containing v. These exchanges preserve monotonicity: The parent of v, p_v, sends to node u. Since $c(p_v) \leq c(v)$ and $c(v) < c(u)$, $c(p_v) < c(u)$. The parent of u, p_u, sends to node v. The arrival time at p_u is strictly less than the arrival time at v, $c(p_u) \leq c(v)$ because T_1^{i+1} satisfies property P_{i+1}. Let T_2^{i+1} denote the schedule that results after performing all such exchanges.

Next, the delays are removed from T_2^{i+1} and the resulting schedule is denoted by T^{i+1}. The removal of the delays cannot affect monotonicity. Nodes with message initiation costs less than $C(i)$ are not affected by the transformation from T^i to T^{i+1}. Next consider a node v with message initiation cost equal to $C(i)$. In T_2^{i+1} such a node received at least as early as any node u with larger message initiation cost. If u received later than v then the difference in arrival times was at least $C(i)$. Since the introduced delays were strictly smaller than $C(i)$, node u still receives after node v after the delays are removed. If nodes u and v have the same arrival times in T_2^{i+1} then by the construction of T_2^{i+1} node v receives the message at least as early as node u when the delays are removed. Thus, T^{i+1} satisfies property P_{i+1}.

The transformation from T^i to T^{i+1} introduces a delay of less than $C(i)$ in the multicast completion time. Thus, the completion time of T^k is at most $\Sigma_{i=1}^{k-1} C(i)$ larger than the completion time of optimal schedule T^1. Schedule T^k is layered and therefore by Corollary 1 the greedy algorithm produces a schedule with completion time no larger than that of T^k. □

In some cases it may be desirable to find optimal multicast schedules. In particular, for small values of k it may be practical and desirable to precompute the table of all optimal schedules. We now show that for any fixed constant k, the optimal multicast problem for n nodes of k distinct types can be solved in time $O(n^{2k})$.

Let $\tau(s, i_1, \ldots, i_k)$ represent the minimum time required to perform a multicast from a source of type s, $1 \leq s \leq k$ to i_j nodes of type j, $1 \leq j \leq k$. Recall that $C(i)$ denotes the message initiation cost of a node of type i. Our algorithm is based on the following lemma.

Lemma 5 *For every* $1 \leq s \leq k$, $i_j \geq 0$, $1 \leq j \leq k$,

$$\tau(s, 0, 0, \ldots, 0) = 0$$

$$\tau(s, i_1, \ldots, i_k) = \min_{1 \leq \ell \leq k, 0 \leq y_1 \leq i_1, \ldots, 0 \leq y_\ell \leq i_\ell - 1, \ldots, 0 \leq y_k \leq i_k} \max\{\tau(\ell, y_1, \ldots, y_\ell, \ldots, y_k) + C(s), \tau(s, i_1 - y_1, \ldots, i_\ell - 1 - y_\ell, \ldots, i_k - y_k) + C(s)\}$$

Proof: The first equation is by definition. For the second equation, in any schedule in which a source node of type s sends to i_j nodes of type j, $1 \leq j \leq k$, the source node's first transmission is sent to some node of type ℓ, $1 \leq \ell \leq k$. This node of type ℓ is the root of a subtree containing $0 \leq y_j \leq i_j$ nodes of type j for each $1 \leq j \leq k$ with the exception of type ℓ, for which $y_\ell \leq i_\ell - 1$ since the selected node of type ℓ is one of the i_ℓ destination nodes for the original multicast. The source node incurs a message initiation cost of $C(s)$ and the optimal solution for this subproblem completes after an additional $\tau(\ell, y_1, \ldots, y_\ell, \ldots, y_k)$ units of time. Once the source node has delivered the message to its first destination, the source node can continue transmitting the message to the remaining destinations. The source node begins performing these remaining transmissions at time $C(s)$ and, by definition, the optimal solution for these remaining destinations takes time $\tau(s, i_1 - y_1, \ldots, i_\ell - 1 - y_\ell, \ldots, i_k - y_k)$. Therefore, the maximum of $\tau(\ell, y_1, \ldots, y_\ell, \ldots, y_k) + C(s)$ and $\tau(s, i_1 - y_1, \ldots, i_\ell - 1 - y_\ell, \ldots, i_k - y_k) + C(s)$ is the actual completion time of the multicast. Finally, by computing the minimum over all possible values of $\ell, y_1, \ldots, y_k$, the completion time of an optimal schedule is found. □

Theorem 2 *For any constant k, given n nodes of k types an optimal multicast schedule can be found in time $O(n^{2k})$.*

Proof: The algorithm applies dynamic programming to the relation in Lemma 5. Specifically, let $n_1, n_2, \ldots, n_k$ represent the number of nodes of each of the k types. The algorithm now performs as follows:

for $s = 1$ **to** k set $\tau(s, 0, 0, \ldots, 0) = 0$
  for $s = 1$ **to** k
    for $i_1 = 0$ **to** n_1
      for $i_2 = 0$ **to** n_2
        ...
          for $i_k = 0$ **to** n_k
            compute $\tau(s, i_1, \ldots, i_k)$ using the relation in Lemma 5.

This dynamic program computes $O(k \times n_1 \times \ldots \times n_k)$ values of the form $\tau(s, i_1, \ldots, i_k)$. The computation of each such value requires $O(k \times i_1 \times \ldots \times i_k)$ steps. Since $i_j \leq n_j \leq n$ for each $1 \leq j \leq k$, the total running time is $O(k^2 n^{2k}) = O(n^{2k})$ for any constant k. □

Note that the dynamic programming table in Theorem 2 contains the values of $\tau(s, i_1, \ldots, i_k)$ for all $1 \leq s \leq k$, $1 \leq i_j \leq n_j$, $1 \leq j \leq k$. Thus, for a network with small k it may be desirable to precompute the dynamic programming table and annotate each entry in the table with the optimal schedule. In this way, an optimal schedule can subsequently be found in constant time for any multicast in this network.

4 An Approximation Algorithm for the General Problem

In this section we show that the greedy algorithm is an approximation algorithm for the multicast problem with a ratio bound of two. In other words, for any multicast set the completion time of the schedule produced by the greedy algorithm is no more than twice the completion time of the optimal schedule. This result is based on the following lemma.

Lemma 6 *Let S be a multicast set and let T be a non-idling schedule for S. Let u, v be two non-root nodes in T such that $t(u) < t(v)$ and $c(u) = \ell \times c(v)$ for some positive integer ℓ. Then there exists a schedule T' satisfying the following properties:*

1. *$t'(u) > t'(v)$.*
2. *$t(w) = t'(w)$ for all $w \in S$ such that w is not a descendant of u or v in T.*
3. *T' is non-idling.*
4. *The completion time of T' is no larger than that of T.*

Proof: Let $\alpha = \{\alpha_1, \ldots, \alpha_a\}$ and $\beta = \{\beta_1, \ldots, \beta_b\}$ denote the arrival ordered list of children of nodes u and v, respectively, in schedule T. Let $t(v) = t(u) + \delta$. Since T is non-idling, $t(\alpha_i) = t(u) + i \times c(u) = t(u) + i \times \ell \times c(v)$, $1 \leq i \leq a$, and $t(\beta_i) = t(u) + \delta + i \times c(v)$, $1 \leq i \leq b$. Partition β into $a + 1$ lists β^i, $1 \leq i \leq a + 1$ as follows:

1. β^1 is the arrival ordered list of all $x \in \beta$ such that $t(x) < t(\alpha_1) + \delta$.
2. For $2 \leq i \leq a$, β^i is the arrival ordered list of all $x \in \beta$, such that $t(\alpha_{i-1}) + \delta \leq t(x) < t(\alpha_i) + \delta$.
3. β^{a+1} is the arrival ordered list of all $x \in \beta$ such that $t(\alpha_a) + \delta \leq t(x)$.

Let β^i_j denote the j^{th} element in list β^i, let β^i_{last} denote the last element in list β^i, and let $\beta^i_{\overline{j}}$ denote the list of nodes formed by removing node β^i_j from list β^i. Note that a list β^i may be empty, in which case β^j is also empty for $i \leq j \leq a + 1$ by the assumption that T is non-idling.

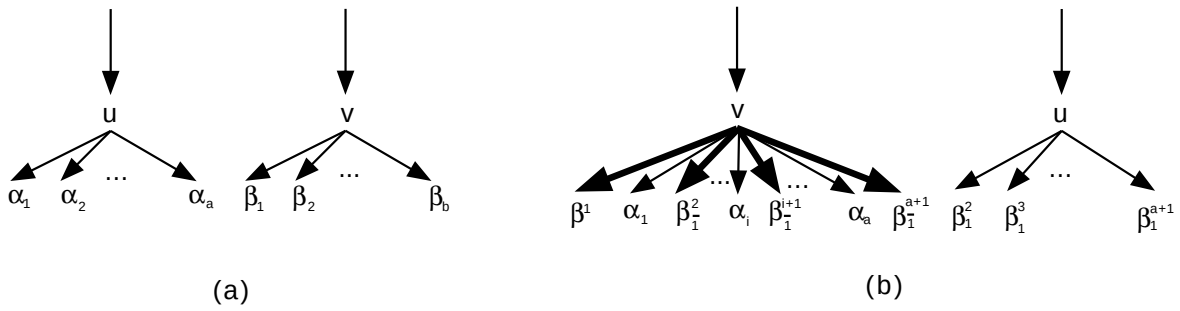

Figure 4: The transformation in Lemma 6. Thick edges indicate zero or more transmissions to an arrival ordered list of nodes. (a) Nodes u, v and their children in schedule T. (b) Nodes u, v and their children in schedule T'.

Schedule T' is constructed as follows: First, nodes u and v exchange positions. The children of v in T' are given by the arrival ordered list $\beta \circ (\alpha_1) \circ \beta_1^2 \circ \ldots \circ (\alpha_i) \circ \beta_1^{i+1} \circ \ldots \circ (\alpha_a) \circ \beta_1^{a+1}$ where $\circ$ is the list concatenation operator. Node u sends the message to the list $(\beta_1^2) \circ \ldots \circ (\beta_1^{a+1})$. The transformation is illustrated in Figure 4.

The first three properties of the lemma follow by construction of T'. We now show that the completion time of T' is no larger than that of T. Without loss of generality, we assume that β^1 and β_1^i, $2 \leq i \leq a+1$, are non-empty. Note that by construction of T', if one or more such lists is empty, some nodes in T' may have even earlier arrival times. By definition of β^i and the assumption that T is non-idling $t(\beta_1^i) = t(u) + \delta + (i-1) \times \ell \times c(v)$, $2 \leq i \leq a+1$. Similarly, $t(\beta_{\text{last}}^i) = t(\beta_1^{i+1}) - c(v) = t(u) + \delta + i \times \ell \times c(v) - c(v)$, $1 \leq i \leq a$.

Consider first the nodes β_1^i, $2 \leq i \leq a+1$. By construction, $t'(\beta_1^i) = t'(u) + (i-1) \times c(u) = t(u) + \delta + (i-1) \times \ell \times c(v) = t(\beta_1^i)$. All other nodes in β are children of v in T' and, by construction of T', if $x \in \beta$ is the i^{th} child of v in T' then it is also the i^{th} child of v in T. Since $t'(v) = t(v) - \delta$, all such nodes have earlier arrival times in T' than in T. Finally, consider the nodes α_i, $1 \leq i \leq a$. In T', $t'(\alpha_i) = t'(\beta_{\text{last}}^i) + c(v)$. However, $t'(\beta_{\text{last}}^i) = t(\beta_{\text{last}}^i) - \delta = t(u) + \delta + i \times \ell \times c(v) - c(v) - \delta$. Thus, $t'(\alpha_i) = t(u) + i \times \ell \times c(v) = t(\alpha_i)$. Therefore, the completion time of T' is no larger than that of T. □

Theorem 3 *For any multicast set S, the greedy algorithm produces a schedule with completion time no larger than twice that of an optimal solution.*

Proof: For a given multicast set S, let OPT denote the completion time of an optimal schedule and let GREEDY denote the completion time of the schedule produced by the greedy algorithm. Let f denote the minimum message initiation cost over all nodes in S. Let S' be the multicast set constructed as follows: For each $u \in S$, introduce a corresponding u' in set S' such that $c(u') = 2^k \times f$ for the smallest integer value k such that $2^k \times f \geq c(u)$. Let OPT$'$ and GREEDY$'$

denote the completion times of an optimal schedule and a greedy schedule, respectively, for set S'. Since $c(u') \leq 2 \times c(u)$ for each node $u \in S$, $\text{OPT}' \leq 2 \times \text{OPT}$. Let $T_{\text{OPT}'}$ be an optimal schedule for S'. Since the message initiation cost of each node in S' is a power of two, Lemma 6 can be applied to any pair of nodes $u', v' \in S'$ such that u' receives the message before v' in $T_{\text{OPT}'}$ but $c(u') > c(v')$. Since the transformation of Lemma 6 affects only the subtrees rooted at the exchanged nodes, a finite number of applications of the transformation can be applied to transform $T_{\text{OPT}'}$ into a layered schedule for S' with completion time OPT'. By Corollary 1, the greedy algorithm finds a schedule for S' with completion time OPT'. By Lemma 3, $\text{GREEDY} \leq \text{GREEDY}'$. Thus, $\text{OPT} \leq \text{GREEDY} \leq \text{GREEDY}' = \text{OPT}' \leq 2 \times \text{OPT}$. □

5 Conclusions and Future Work

In this paper we have considered the problem of multicasting in the heterogeneous node model. We have shown that for a fixed number of workstation types, each with a fixed message initiation cost, a greedy algorithm can be used to find solutions that are within a constant additive term of optimal. We have also shown that optimal solutions can be found in polynomial time where the polynomial depends on the number of workstation types. Finally, we have shown that for the general problem, a greedy algorithm finds solutions that are within a factor of two of optimal.

It is unknown whether the bounds on the approximation algorithms presented here are tight and this is a topic for future research. Another natural question is whether similar results can be obtained for other heterogeneous communication models. For example, one possible extension of the LogP model [5] to heterogeneous networks associates with each node v a *sending overhead* $o_s(v)$, *receiving overhead* $o_r(v)$, and a *gap* $g(v)$, specifying the amount of time that node v is busy sending a message, receiving a message, and the gap between consecutive sends or receives at that node. The $O(n^{2k})$ dynamic programming algorithm described in Section 3 can be adapted for this communication model. On the other hand, the approximation algorithms described here do not generalize to this model. The search for multicast algorithms in such models is an important field for further study. In addition, similar optimization problems arise in other collective communication operations and these problems also present a wide and interesting area for future research.

Acknowledgements

The authors wish to thank the three anonymous reviewers for many valuable comments and suggestions which improved this paper.

References

[1] M. Banikazemi, V. Moorthy, and D. Panda. Efficient collective communication on heterogeneous networks of workstations. In *Proc. of the 1998 Int. Conf. on Parallel Processing*, August 1998.

[2] A. Bar-Noy and S. Kipnis. Designing broadcast algorithms in the postal model for message-passing systems. *Mathematical Systems Theory*, 27:431–452, 1994.

[3] P. Bhat, C.S. Raghavendra, and V. Prasanna. Efficient collective communication in distributed heterogeneous systems. In *Proc. of the 19th IEEE Intl. Conf. on Distributed Computing and Systems*, 1999.

[4] J. Bruck, L. de Coster, N. DeWulf, C.-T. Ho, and R. Lauwereins. On the design and implementation of broadcast and global combine operations using the postal model. *IEEE Transactions on Parallel and Distributed Systems*, 7(3):256–265, March 1996.

[5] D. Culler et al. LogP: Towards a realistic model of parallel computation. In *Proc. of the Fourth ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming*, San Diego, CA, May 1993.

[6] A. M. Farley. Broadcast time in communication networks. *SIAM Journal on Applied Mathematics*, 39(2):385–390, October 1980.

[7] N. Hall, W.-P. Liu, and J. Sidney. Scheduling in broadcast networks. *Networks*, 32:233–253, 1998.

[8] G. Itkis, I. Newman, and A. Schuster. Broadcasting on a budget in the multi-service communication model. In *Proceedings of the Fifth International Conference on High Performance Computing*, December 1998.

[9] S. Johnsson and C.-T. Ho. Broadcasting and personalized communication in hypercubes. *IEEE Transactions on Computers*, 38(9):1249–1268, September 1989.

[10] R. Karp, A. Sahay, and E. Santos. Optimal broadcast and summation in the LogP model. Technical Report CSD-92-721, Department of Computer Science, University of California, Berkeley, 1992.

[11] P. McKinley, Y. Tsai, and D. Robinson. Collective communication in wormhole-routed massively parallel computers. *IEEE Computer*, pages 39–50, Dec. 1995.

[12] J. Park, H. Choi, N. Nupairoj, and L. Ni. Construction of optimal multicast trees based on the parameterized communication model. In *Proc. of the Int. Conf. on Parallel Processing*, pages 180–187, Aug. 1996.