Algorithmica (2010) 57: 848–868
DOI 10.1007/s00453-008-9249-0

Broadcasting on Networks of Workstations

Samir Khuller · Yoo-Ah Kim · Yung-Chun Justin Wan

Received: 21 February 2006 / Accepted: 23 October 2008 / Published online: 11 November 2008

Abstract Broadcasting and multicasting are fundamental operations. In this work we develop algorithms for performing broadcast and multicast in clusters of workstations. In this model, sending a message to a machine in the same cluster takes 1 time unit, and sending a message to a machine in a different cluster takes $C(\geq 1)$ time units. The clusters may have arbitrary sizes. Lowekamp and Beguelin proposed heuristics for this model, but their algorithms may produce broadcast times that are arbitrarily worse than optimal. We develop the first constant factor approximation algorithms for this model. Algorithm LCF (Largest Cluster First) for the basic model is simple, efficient and has a worst case approximation guarantee of 2. We then extend these models to more complex models where we remove the assumption that an unbounded amount of communication may happen using the global network. The algorithms for these models build on the LCF method developed for the basic problem. Finally, we develop broadcasting algorithms for the postal model where the sending node does not block for C time units when the message is in transit.

Research supported by NSF Award CCR-0113192.

S. Khuller · Y.-C.J. Wan
Department of Computer Science, University of Maryland, College Park, MD 20742, USA

S. Khuller
e-mail: samir@cs.umd.edu

Y.-C.J. Wan
e-mail: ycwan@cs.umd.edu

Y.-A. Kim (✉)
Department of Computer Science and Engineering, University of Connecticut, Storrs, CT 06269, USA
e-mail: ykim@engr.uconn.edu

1 Introduction

Networks of Workstations (NOWs) are a popular alternative to massively parallel machines and are widely used (for example the Condor project at Wisconsin [18] and the Berkeley NOW project [17]). By simply using off-the-shelf PC's, a very powerful workstation cluster can be created, and this can provide a high amount of parallelism at relatively low cost. With the recent interest in grid computing [8] there is an increased interest to harness the computing power of these clusters to have them work together to solve large applications that involve intensive computation. Several projects such as Magpie [13, 14] are developing platforms to allow applications to run smoothly by providing primitives for performing basic operations such as broadcast, multicast, scatter, reduce, etc. Many of these primitives are implemented using simple heuristics. Our goal is to develop models, and an understanding of the difficult issues and challenges in implementing broadcast and multicast on such platforms. Several approximation algorithms have been developed in the theory literature, but they are for different models (typically an underlying communication graph exists that forbids any communication between non-adjacent nodes).

One fundamental operation that is used in such clusters, is that of *broadcast* (this is a primitive in many message passing systems such as MPI [4, 9, 11, 16]). Some of this framework has been extended to clustered wide area systems (see [13, 14]) of the type we are addressing. In addition it is used as a primitive in many parallel algorithms. The main objective of a broadcast operation is to quickly distribute data to the entire network for processing. Another situation is when the system is performing a parallel search, then the successful node needs to inform all other nodes that the search has concluded successfully. Various models for heterogeneous environments have been proposed in the literature. One general model is the one proposed by Bar-Noy et al. [2] where the communication costs between links are not uniform. In addition, the sender may engage in another communication before the current one is complete. An approximation factor with a guarantee of $O(\log k)$ is given for the operation of performing a multicast. Other popular models in the theory literature generally assume an underlying communication graph, with the property that only nodes adjacent in this graph may communicate. See [6, 7] for recent approximation algorithms on this model. However, this model is too restrictive and allows direct communication only between nodes adjacent in a certain communication graph. Broadcasting efficiently is an essential operation and many works are devoted to this (see [1, 3, 10, 12, 19] and references therein). For example, the LogP model [5] suggests an alternative framework when dealing with machines in a single cluster. Broadcasting algorithms [12] for the LogP model have been developed and shown to be optimal.

Consider several clusters of workstations. Each local cluster (sometimes this is also called a subnet) is connected on a fast local area network, and inter-cluster communication is via a wide area network. In such situations, the time taken for a pair of machines in the same cluster to communicate, can be significantly smaller than the communication time of a pair of machines in different clusters. In fact, in the work by Lowekamp and Beguelin [15] they also suggest methods for obtaining the subnets/clusters based on communication delays between pairs of nodes. Several other papers consider similar hierarchical models [13, 14, 20].

Motivated by this, the *communication model* we consider is the following. There are k clusters of machines. Cluster i has size n_i. We will assume that in one time unit, a machine can send a message to any machine in its own cluster. However, sending a message from one machine to another machine in a different cluster takes $C(\geq 1)$ time units. Even if the machines in a cluster are heterogeneous, their transmission times are usually much less than the communication time across clusters. We also assume that a machine can be sending or receiving a message from only one machine at any point of time. In addition, assume that each cluster advertises a single address to which messages are sent. Each cluster thus receives a message only once at this machine and then the message is propagated to different machines in the cluster. Thus new machines may be added or dropped without having to inform other clusters of the exact set of new addresses (we only need to keep track of the sizes of the clusters). In this model Lowekamp and Beguelin [15] propose some simple heuristics for performing broadcast and multicast. However, these heuristics may produce solutions arbitrarily far from optimal. Other heuristic algorithms for collective communications have been proposed in multi-tier networks [13, 14, 20] but there has been little work done on theoretical analysis of the algorithms.

One potential concern with the above model is that it allows an arbitrary number of machines to communicate in every time step. This is of concern if the global network connecting the different clusters does not have enough capacity to permit such arbitrary communication patterns. There are several ways in which we can restrict the model. One model that we propose is the *bounded degree model* where each cluster i is associated with a parameter d_i that restricts the *number* of machines from this cluster that can communicate with machines outside this cluster in each time step. Another possible manner in which we may restrict global communication in each time step is to restrict the *total* number of simultaneous transfers that may be going on in each time step without restricting the number of transfers into/out of a single cluster. We call this model the *bounded size matching model*.

In addition, we consider a *postal model* [1] where each message simply has a latency of C time units when the message is sent from one machine to another machine belonging to a different cluster. The sender is busy for only one time unit while the message is being injected into the network. The message takes C units of transit time and the receiver is busy for one unit of time when the message arrives. This model essentially captures the communication pattern as discussed in several papers that deal with implementations of systems to support such primitives (see [12–14]). This model is motivated by the interest in Grid computing [8] and computing on clusters of machines. In fact, the work on the Magpie project [13, 14] specifically supports this communication model.

We develop constant factor approximation algorithms for broadcasting and multicasting for those models. In many of these cases the algorithm we develop, called Algorithm *Largest Cluster First* (*LCF*) plays a central role. We first show that there is a simple analysis that shows that the worst case broadcast time can be bounded by a factor of 3 for this algorithm in the basic model. We then improve the *lower bound* on *OPT* by introducing the concept of "experiencing" inter-cluster transfers to improve the approximation factor to 2. This lower bound in a sense combines the difficulty of propagating messages to a large number of clusters with the fact that some of the clusters may be very large.

1.1 Formal Description of Problem and Summary of Results

Assume that there are k clusters of nodes (machines). Cluster K_i has size $n_i, i = 0 \ldots (k-1)$, the number of nodes in the i-th cluster. The total number of nodes, denoted by N, is $\sum_{i=0}^{k-1} n_i$. In the broadcasting problem, we want to have all nodes in the network receive a message with the minimum amount of time. In the multicast problem, only a subset of nodes in each cluster may want to receive the message but other nodes can participate in message transfers to reduce the multicast time. We assume that the broadcast/multicast originates at a node in K_0. We order the *remaining* clusters in non-increasing size order. Hence $n_1 \geq n_2 \geq \cdots \geq n_{k-1}$. Note that n_0 could be smaller or larger than n_1; since it is the cluster that originates the broadcast/multicast.

A node may forward a message, once it has received the message. All nodes are available to receive a message from the beginning of the schedule. If the message is sent to a node in its cluster, the message arrives one time unit later. If the message is sent to a node in a different cluster then the message arrives $C \geq 1$ time units later (we call this $1/C$ model). Both the sending and receiving nodes are busy during those C time units. Note that C need not be an integer. In addition, we assume that each cluster receives a message only once from other clusters at the machine whose address is advertised. Note that in some cases, the broadcast time can be reduced by having many messages arrive at the same cluster. *However, when we compare to the optimal solution we do not make any assumptions about the communication structure of the optimal solution.*

We consider two models to restrict the network capacity. The *bounded degree model* restricts the number of inter-cluster transfers associated with a particular cluster, while the *bounded size matching model* restricts the total number of inter-cluster transfers at any given time.

We also consider a slightly different postal model (Sect. 5) where a node is busy for only one time unit when it sends a message. The time a message arrives at a receiver depends on whether the sender and receiver are in the same cluster or not—it takes one time unit if it is a local transfer, and C time units otherwise.

The results for these models can be described as follows.

1. We develop algorithms for broadcasting and multicasting in the basic $1/C$ model, and show that these algorithms produce solutions where the time to perform the broadcast/multicast is not more than optimal by a factor of 2. Moreover, this bound is tight for both algorithms.
2. For the *bounded degree model* we show how to reduce the problem to an instance of the basic model to develop a 3 approximation algorithm for both broadcasting and multicasting.
3. For the *bounded size matching* model we develop an algorithm for which we prove a factor 2 approximation. The corresponding approximation for multicasting is also 2.
4. For the *postal model* our algorithm has an approximation factor of 3 for both broadcasting and multicasting. In addition, we present another algorithm, called *Interleaved LCF* and show that the broadcast time is at most 2 times OPT' where OPT' is the minimum broadcast time among schedules that minimize the total number of global transfers.

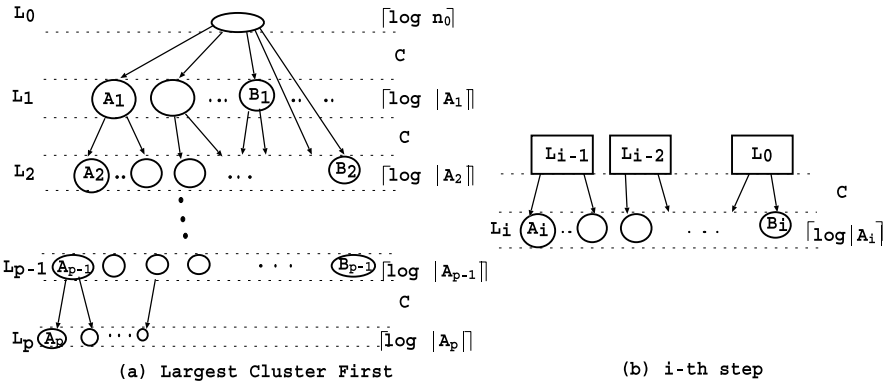


Fig. 1 (**a**) Illustration of *LCF*. (**b**) At i-th step, all nodes in clusters $K \in L_j$ $(j = 0, \ldots, i-1)$ send messages to L_i and then clusters in L_i perform local broadcasting. Therefore, i-th step takes $C + \lceil \log A_i \rceil$

In some applications, the cluster sizes may not be known accurately in advance. We study the effect of having inaccurate information regarding the sizes of the clusters (Appendix B) and demonstrate, even if there is a multiplicative factor of 2 inaccuracy in the sizes of the clusters, there is hardly any change in the performance of the broadcast algorithms.

2 Broadcasting

The high level description of the algorithm is as follows: The source node first performs a local broadcast within its cluster. This takes $\lceil \log n_0 \rceil$ time units. After all the nodes of K_0 have the message, they send the message to the first n_0 clusters. That is, each node in K_0 sends a message to a distinct cluster. This takes exactly C time units. Each cluster that receives a message, performs a local broadcast within its cluster. All nodes that have received a message then send the message to distinct clusters. Again this takes C more time units. Repeat this until all the nodes receive the message. We call this algorithm *Largest Cluster First (LCF)* as we always choose the largest cluster as a receiver among clusters that have not received the message (see Fig. 1 for example).

Let L_i be the set of clusters that receive the message at i-th global step. For example, L_0 includes K_0 and L_1 includes all clusters that receive the message from K_0 at the end of the first global phase. Let N_i be the total number of nodes in clusters belonging to L_i. That is, $N_i = \sum_{K \in L_i} |K|$. The following is the formal description of the algorithm.

Algorithm *LCF*

1. Broadcast locally in K_0. Then we have that $L_0 = K_0$ and $N_0 = n_0$.
2. At i-th step (repeat until all clusters have at least one informed node)
   (a) Global phase: Pick $\sum_{j=0 \ldots i-1} N_j$ largest clusters that are not informed as yet. Each node in $\bigcup_{j=0 \ldots i-1} L_j$ sends one message to each of those clusters.

(b) Local phase: Clusters in L_i do local broadcasting. (For further optimization, we can interrupt all local broadcasting if the number of nodes having the message is at least as big as the number of clusters that have not received a message and perform the remaining local broadcasting later.)

In our algorithm each cluster receives only one message from other clusters. That is, the total number of global transfers is minimized (we need $k-1$ global transfers). This property is important since we may want to avoid wasting wide area bandwidth which is typically more expensive. Note that we can further reduce the broadcast time by allowing nodes to start global transfers without idle time once all nodes in the same cluster get informed. That is, a node can repeatedly perform a global transfer every C time units after all nodes in its cluster receive the message.

In the following subsection, we prove that *LCF* gives a 2-approximation for broadcasting (even without the optimization mentioned above). We do not make any assumptions about the communication structure of the optimal solution when we compare to the optimal solution. Moreover, we show that our analysis is tight in Sect. 2.2.

2.1 Analysis

Let p be the number of global transfer steps that *LCF* uses. Then we have the following theorem.

Theorem 2.1 *The broadcast time of Algorithm* LCF *is at most* $2\log N + pC + 3$.

Our analysis frequently uses the following simple fact.

Fact 2.1 *It takes at least* $\log n$ *time units to broadcast a message to n different nodes.*

Let us define A_i (resp. B_i) to be the biggest (resp. smallest) cluster in L_i. We need the following two lemmas to prove this theorem.

Lemma 2.1 *For* $i = 0 \ldots p-1$, $n_0 \cdot |B_1| \cdots |B_i| \leq N_i$.

Proof We prove this by induction. For $i = 0$, it is true since $N_0 = n_0$. Suppose that at the i-th step ($i < p-1$), $n_0 \cdot |B_1| \cdots |B_i| \leq N_i$. Since at $(i+1)$-th global transfer step, every node in N_i (as well as nodes in N_j, $j < i$) will send the message to a cluster in L_{i+1}, $|L_{i+1}| \geq N_i$ (for $i < p-1$). Furthermore, the size of clusters in L_{i+1} is at least $|B_{i+1}|$ by definition. Therefore,

$$\begin{aligned} N_{i+1} &= \sum_{K \in L_{i+1}} |K| \geq \sum_{K \in L_{i+1}} |B_{i+1}| \\ &= |L_{i+1}| \cdot |B_{i+1}| \geq N_i \cdot |B_{i+1}| \geq n_0 \cdot |B_1| \cdots |B_i| \cdot |B_{i+1}|. \end{aligned}$$

Thus the lemma follows. □

Lemma 2.2 $\log |A_1| < \log N - (p-2)$.

Proof After all nodes in A_1 receive the message, we need $p-1$ more global transfer steps. With $|A_1|$ copies, we can make $|A_1| \cdot 2^i$ nodes receive the message after i global transfer steps by doubling the number of copies in each global step. Therefore $|A_1| \cdot 2^{p-2} < N$ (otherwise, we do not need p-th global broadcasting step), which gives the lemma. □

Proof of Theorem 2.1 The upper bound on the total broadcast time for local transfer steps, is $\lceil \log n_0 \rceil + \lceil \log |A_1| \rceil + \cdots + \lceil \log |A_p| \rceil$. Since we have $|A_i| \leq |B_{i-1}|$ (for $2 \leq i \leq p$) in *LCF*, it is upper bounded by $\lceil \log n_0 \rceil + \lceil \log |A_1| \rceil + \lceil \log |B_1| \rceil + \cdots + \lceil \log |B_{p-1}| \rceil$. Then the total broadcast time only for local transfer steps is at most

$$\begin{aligned}
&\lceil \log n_0 \rceil + \lceil \log |A_1| \rceil + \lceil \log |B_1| \rceil + \cdots + \lceil \log |B_{p-1}| \rceil \\
&\quad \leq \log n_0 + \log |B_1| + \cdots + \log |B_{p-1}| + \log |A_1| + p + 1 \\
&\quad < \log(n_0 \cdot |B_1| \cdots |B_{p-1}|) + \log N - (p-2) + p + 1 \\
&\quad \leq \log N_{p-1} + \log N + 3 \leq 2 \log N + 3.
\end{aligned}$$

The second line can be obtained by rearranging the terms and removing ceilings, and the third and fourth lines follow by Lemma 2.2 and Lemma 2.1, respectively. Our schedule uses p global transfer steps, taking a total of pC additional time units. Thus the theorem follows. □

In the next lemma, we prove that pC is a lower bound on the optimal broadcast time. To prove this we count the number of inter-cluster transfers a node *experiences* as follows. Given an optimal broadcast schedule, let the path from the source to the node p_i, be $a_0, a_1, \ldots, a_l = p_i$. That is, suppose that in the optimal solution the source a_0 sends the message to a_1 and a_1 sends to a_2 and so on. Finally a_{l-1} sends the message to node p_i $(= a_l)$. Let e_j $(j = 0 \ldots l-1)$ represent the number of nodes that receive the message from a_j via inter-cluster transfers until a_{j+1} receives the message (including the transfer to a_{j+1} if they are in different clusters). In addition, let e_l be the number of nodes that receive the message from node p_i via inter-cluster transfers. That is, p_i sends the message to e_l nodes in other clusters. Then we say node p_i experiences e inter-cluster transfers where $e = \sum_{j=0}^{l} e_j$. Figure 2 shows an example of how to count the number of inter-cluster transfers that a node experiences. In the example, node p_i experiences 11 inter-cluster transfers. If there is any node that experiences p inter-cluster transfers in an optimal solution, then pC is a lower bound on the optimal.

Lemma 2.3 *At least one node in the optimal solution experiences p inter-cluster transfers.*

Proof Imagine a (more powerful) model in which once a node in a cluster receives the message, all nodes in the cluster receive the message instantly (that is, local transfers take zero unit of time). In this model the broadcast time is given by the maximum number of inter-cluster transfers that any node experiences. We will prove that *LCF* gives an optimal solution for this model. Since *LCF* uses p global transfer steps, the

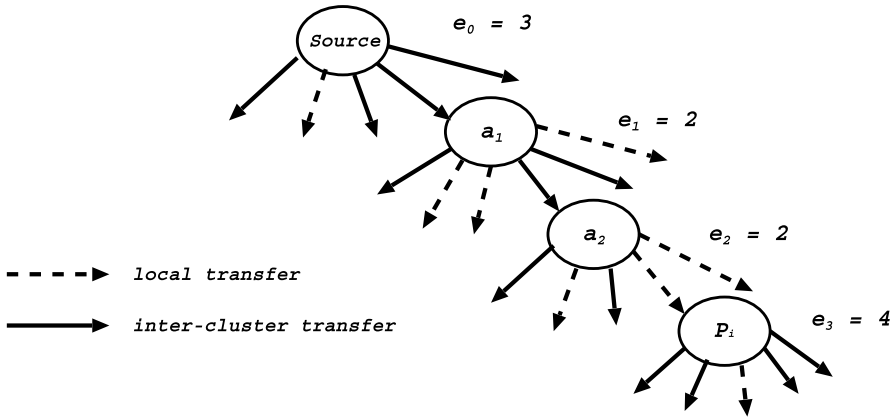


Fig. 2 An example to show the inter-cluster transfers node p_i experiences

optimal broadcast time is pC in this model. Since this lower bound is for a stronger model, it also works in our model.

Suppose that there is a pair of clusters K_i and K_j $(0 < i < j \leq k)$ such that K_j receives the message earlier than K_i in the optimal solution for this model. Let us assume that K_i receives the message at time t_i and K_j receive it at time t_j $(t_i > t_j)$ in the solution. We modify the schedule and show that the modification does not increase the broadcast time. At time t_j in the modified schedule, K_i (instead of K_j) receive the message performs global transfers in the same way as K_j does until t_i. (This can be done since the size of K_i is at least as big as K_j and local broadcasting take zero unit of time). At time t_i, K_j receives the message and after that the original schedule can be performed. This exchange does not increase the broadcast time and therefore, *LCF* gives an optimal solution for the model with zero local transfer costs.

We now prove the lemma by contradiction. Suppose that there is an optimal solution (for the original model) in which all nodes experience at most $p-1$ inter-cluster transfers. Then in the model with zero local transfer costs we should be able to find a solution with broadcasting time $(p-1)C$ by ignoring local transfers, which is a contradiction. □

As there is at least one node experiencing p global transfers, pC is a lower bound on the broadcast time. Combined with Fact 2.1 and Theorem 2.1, it directly gives us a 3-approximation algorithm for broadcasting (with an additive term of 3). In the following, we present a tighter analysis, which shows that *LCF* gives a 2-approximation. Note that the lower bound pC considers only the global communications and the lower bound $\log N$ counts the local transfers. To get an approximation factor of 2, we prove the following theorem that combines the two lower bounds.

Theorem 2.2 *The optimal solution has to take at least* $(p-1)(C-1) + \lceil \log N/2 \rceil$ *time units.*

Proof Consider an optimal schedule. We partition all nodes into two sets, S_l and S_s, where S_l contains all nodes which experienced at least $p-1$ inter-cluster transfers, and S_s contains all nodes which experienced at most $p-2$ inter-cluster transfers. We now show that $|S_s| < N/2$. Suppose this is not the case, it means that the optimal solution can inform at least $N/2$ nodes using at most $p-2$ inter-cluster transfers. Using one more step of transfers, we can double the number of nodes having the message and inform all N nodes. This is a contradiction, since we use less than p inter-cluster transfers (see Lemma 2.3). Therefore we have $|S_l| \geq N/2$. Since originally we have one copy of the message, informing nodes in S_l takes at least $\lceil \log N/2 \rceil$ transfers. So at least one node (say, p_e) in S_l experienced $\lceil \log N/2 \rceil$ transfers (either inter-cluster or local transfers). We know that all nodes in S_l experienced at least $p-1$ inter-cluster transfers. Therefore, node p_e needs at least $(p-1)C + (\lceil \log N/2 \rceil - (p-1))$ time units to finish. □

We now prove a central result about Algorithm *LCF*, which will be used later.

Lemma 2.4 *Our algorithm takes at most* $2OPT + 7$ *time units where OPT is the minimum broadcast time. Moreover, it takes at most* $2OPT$ *time units when both* p *and* C *are not very small* (*i.e., when* $(p-2)(C-2) \geq 7$).

Proof If C is less than 2, we can treat the nodes as one large cluster and do broadcasting. This takes at most $C\lceil \log N \rceil$ and is a 2-approximation algorithm. The problem is also trivial if p is 1, because in this case $n_0 \geq k$, the total number of clusters. Therefore we consider the case where both values are at least 2. Here we make use of Theorems 2.1 and 2.2.

$$\begin{aligned}
&(2OPT + 7) - (2\log N + pC + 3) \\
&\quad \geq (2((p-1)(C-1) + \lceil \log N/2 \rceil) + 7) - (2\log N + pC + 3) \\
&\quad \geq (p-2)(C-2) \geq 0 \quad (\text{when } p \geq 2 \text{ and } C \geq 2).
\end{aligned}$$ □

Theorem 2.3 *There is a polynomial-time* 2*-approximation algorithm for the broadcasting problem.*

2.2 Bad Example

There are instances for which the broadcast time of *LCF* is almost 2 times the optimal (see Fig. 3). Suppose that we have 2 clusters, K_0 and K_1, each of size n_0 and n_1 ($n_0 \leq n_1$), respectively. In addition, there are $n_0 - 1$ more clusters, each of size 1. A node in K_0 has a message to broadcast. It is easy to see that the broadcast time of Algorithm *LCF* is $\lceil \log n_0 \rceil + C + \lceil \log n_1 \rceil$. However, the broadcasting can be made faster by sending a message to K_1 before local broadcast in K_0 is finished. A possible schedule is (i) make one local copy in K_0 (ii) one node in K_0 send a message to K_1 and another node does local broadcast in K_0 (iii) after finishing local broadcast, nodes in K_0 send messages to the remaining $n_0 - 1$ clusters, and (iv) clusters other than K_0

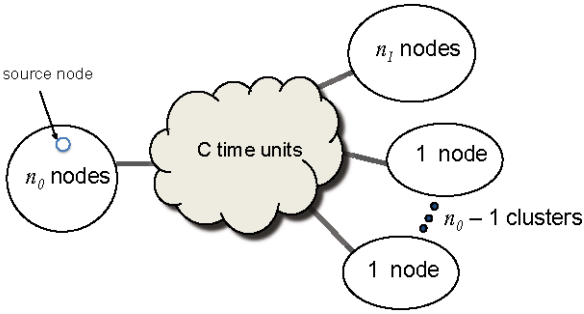


Fig. 3 The broadcast time of Algorithm *LCF* is $\lceil \log n_0 \rceil + C + \lceil \log n_1 \rceil$ whereas the optimal broadcast time is $1 + \max\{C + \lceil \log n_1 \rceil, \lceil \log(n_0 - 1) \rceil + C\}$

do local broadcasting as soon as they receive a message. The broadcast time of this solution is $1 + \max\{C + \lceil \log n_1 \rceil, \lceil \log(n_0 - 1) \rceil + C\}$. In the case where $n_0 \approx n_1$, the approximation factor is given as $(C + 2 \log n_0)/(1 + C + \log n_0)$. When $\log n_0 \gg C$, the broadcast time of *LCF* can be almost 2 times the optimal.

3 Multicasting

For multicasting, we need to have only a subset of nodes receive the message. We may reduce the multicast time significantly by making use of large clusters that may not belong to the multicast group. Let n'_i denote the number of nodes in K_i that belong to the multicast group. Let M denote the set of clusters (except K_0) in which some nodes want to receive the message and k' denote the size of set M. Formally, $M = \{K_i | n'_i > 0 \text{ and } i > 0\}$ and $k' = |M|$.

Let $LCF(m)$ be algorithm *LCF* to make m copies. That is, $LCF(m)$ runs in the same way as *LCF* but stops as soon as the total number of nodes that received the message is at least m (we may generate up to $2(m-1)$ copies). For example, *LCF* for broadcasting is $LCF(N)$.

Algorithm *LCF Multicast*

1. Run $LCF(k')$ by using any node whether it belongs to the multicast group or not.
2. Send one copy to each cluster in M if it has not received any message as yet.
3. Perform local broadcasting in clusters of M only for multicast nodes.

3.1 Analysis

Let p' be the number of global transfer steps $LCF(k')$ uses. Suppose D be the time units taken in the last local broadcast step in $LCF(k')$ (after the p'-th global transfer steps). Note that some nodes in clusters performing the last local broadcast may not receive the message, since we stop as soon as the total number of nodes having the message is at least k', and hence $D \leq \lceil \log |A_{p'}| \rceil$. Note that D may be greater than $\lceil \log |B_{p'}| \rceil$ and thus some clusters may stop local broadcast before the D-th round.

Let $A_{p'+1}$ be the biggest cluster among clusters which have not received a copy in $LCF(k')$.

To prove that the algorithm is a 2-approximation, we need the following proposition.

Proposition 3.1 $(\log n_0 \cdot |B_1| \cdots |B_{p'-2}| + D) + \max(\lceil \log |A_{p'}| \rceil - D, \lceil \log |A_{p'+1}| \rceil) \leq \log k' + 2$

Proof We prove the inequality by considering the following three cases.

Case I ($\lceil \log |A_{p'}| \rceil - D > \lceil \log |A_{p'+1}| \rceil$) It is easy to see that $(\log n_0 \cdot |B_1| \cdots |B_{p'-2}| + D) + (\lceil \log |A_{p'}| \rceil - D) \leq \log n_0 \cdot |B_1| \cdots |B_{p'-1}| + 1 \leq \log k' + 1$, and the lemma follows.

Case IIa ($\lceil \log |A_{p'}| \rceil - D \leq \lceil \log |A_{p'+1}| \rceil$ and $D > \lceil \log |B_{p'}| \rceil$) Note that $2^D \leq 2|A_{p'}| \leq 2|B_{p'-1}|$ and $|A_{p'+1}| \leq |B_{p'}|$; we have $(\log n_0 \cdot |B_1| \cdots |B_{p'-2}| + D) + (\lceil \log |A_{p'+1}| \rceil) < \log n_0 \cdot |B_1| \cdots |B_{p'}| + 2 \leq \log N_{p'-1} \cdot |B_{p'}| + 2$. After the p'-th global transfer step, one node in each of $N_{p'-1}$ clusters has just received the message. Each of these clusters will generate at least $|B_{p'}|$ copies (since $D > \lceil \log |B_{p'}| \rceil$), so $N_{p'-1} \cdot |B_{p'}| < k'$, and the lemma follows.

Case IIb ($\lceil \log |A_{p'}| \rceil - D \leq \lceil \log |A_{p'+1}| \rceil$ and $D \leq \lceil \log |B_{p'}| \rceil$) Note that $|A_{p'+1}| \leq |B_{p'-1}|$; we have $(\log n_0 \cdot |B_1| \cdots |B_{p'-2}| + D) + (\lceil \log |A_{p'+1}| \rceil) \leq \log n_0 \cdot |B_1| \cdots |B_{p'-1}| \cdot 2^D + 1 \leq \log N_{p'-1} \cdot 2^D + 1$. After the p'-th global transfer step, each cluster which has just received the message will generate at least 2^D copies, so $\log N_{p'-1} \cdot 2^{D-1} < k'$, and the lemma follows. □

Theorem 3.1 *Our multicast algorithm takes at most* $2 \log k' + p'C + C + 4$ *time units.*

Proof In a manner similar to the proof of Theorem 2.1, the broadcast time spent only in local transfer steps in $LCF(k')$ is at most

$$\begin{aligned} &\lceil \log n_0 \rceil + \lceil \log |A_1| \rceil + \cdots + \lceil \log |A_{p'-1}| \rceil + D \\ &\leq \log n_0 + \log |B_1| + \cdots + \log |B_{p'-2}| + D + \log |A_1| + p' \\ &< \log n_0 \cdot |B_1| \cdots |B_{p'-2}| + D + \log k' + 2. \end{aligned}$$

The second inequality holds because $\log |A_1| < \log k' - (p' - 2)$ by Lemma 2.2. Moreover, the global transfer steps in $LCF(k')$ and the second phase take $p'C$ and C time units, respectively. Note that $n'_i \leq n_i$. In the third phase, all clusters which receive a message during the first phase need at most $\lceil \log |A_{p'}| \rceil - D$ time units to do local broadcast. For the remaining clusters which receive a message during the second phase, local broadcasting takes at most $\lceil \log |A_{p'+1}| \rceil$ time units as they are of size at most $|A_{p'+1}|$. Therefore, Phase 3 requires at most

$\max(\lceil \log |A_{p'}| \rceil - D, \lceil \log |A_{p'+1}| \rceil)$ time units. Thus the total time units required for multicasting is given as

$$\log n_0 \cdot |B_1| \cdots |B_{p'-2}| + D + \log k' + 2 + p'C + C + \max(\lceil \log |A_{p'}| \rceil - D, \lceil \log |A_{p'+1}| \rceil).$$

Using Proposition 3.1, the theorem follows. □

Lemma 3.1 *At least one node in the optimal solution experiences p' inter-cluster transfers.*

Proof The basic argument is the same as the one in Lemma 2.3. Note that $LCF(k')$ uses any node whether it belongs to the multicast group or not. If the optimal solution does not use any node that $LCF(k')$ uses, it cannot create new copies of the message faster than $LCF(k')$. □

Theorem 3.2 *The optimal solution takes at least $(p'-1)(C-1) + \lceil \log \frac{k'}{2} \rceil$ time units.*

Proof The proof is similar to the proof of Theorem 2.2. We partition all nodes into two sets, S_l and S_s. We now show that there are less than $\frac{k'}{2}$ distinct multicast clusters in S_s. Suppose this is not the case, it means that OPT can inform at least $\frac{k'}{2}$ distinct multicast clusters using at most $p'-2$ inter-cluster transfers. Using one more round of transfers, all k' multicast clusters can receive the message, which is a contradiction. Therefore we have at least $\frac{k'}{2}$ distinct multicast clusters in S_l and $|S_l| \geq \frac{k'}{2}$. □

Lemma 3.2 *Our algorithm takes at most $2OPT + 10$ time units. Moreover, it takes at most $2OPT$ time units when both p' and C are not very small (i.e., when $(p'-3)(C-2) \geq 10$).*

Proof The problem is trivial when C is less than 2 or $p' = 1$. When $p' = 2$, we can do an exhaustive search on the number of clusters in M which receives the message in the first global transfer step in $LCF(k')$. By making use of Theorems 3.1 and 3.2, and an analysis similar to that in Lemma 2.4, we can show that $(2OPT + 10) - (2\log k' + p'C + C + 4) \geq (p'-3)(C-2) \geq 0$ when $p' \geq 3$ and $C \geq 2$. □

Theorem 3.3 *There is a polynomial-time 2-approximation algorithm for the multicast problem.*

4 Bounding Global Transfers

In the previous sections, we assumed that any node may communicate with any other node in other clusters, and the underlying network connecting clusters has unlimited capacity. A more practical model is to restrict the number of pairs of inter-cluster transfers that can happen simultaneously. In this section we present two models to

restrict the network capacity. The *bounded degree model* restricts the number of inter-cluster transfers associated with a particular cluster, while the *bounded size matching model* restricts the total number of inter-cluster transfers at any given time.

4.1 Bounded Degree Model: Broadcasting

We associate an additional parameter d_i with each cluster K_i, which limits the number of inter-cluster transfers from or to nodes in the cluster in a time unit. We call this limitation a degree constraint. We denote an instance of this model to be $I(n_i, d_i)$, meaning that there are n_i nodes in cluster K_i, and at most d_i of those may participate in inter-cluster transfers at a time.

Algorithm *Bounded Degree Broadcast*
Given Instance $I(n_i, d_i)$, arbitrarily select a subset K_i' of d_i nodes in each cluster, and consider only the K_i' subsets as clusters. We then have a new instance $I(d_i, d_i)$. Note that $I(d_i, d_i)$ can be viewed as an instance of the general broadcast problem on the unrestricted model.

1. Run Algorithm *LCF* in Sect. 2 on $I(d_i, d_i)$.
2. Since there are d_i informed nodes in each cluster, do local broadcasting to send the message to the remaining $n_i - d_i$ nodes.

An important observation is that since there is only one message to be broadcast, it does not matter which subset of nodes in a cluster perform inter-cluster transfers. What matters is the number of informed nodes in the clusters at any given time. The following lemma compares the optimal number of time units taken by instances using the two different models.

Lemma 4.1 *The optimal schedule of Instance $I(d_i, d_i)$ takes no more than the optimal schedule of the corresponding instance $I(n_i, d_i)$.*

Proof It is easy to see that given an optimal schedule, which completes in OPT time units, of Instance $I(n_i, d_i)$, there is a schedule, which completes in at most OPT round, of the corresponding Instance $I(d_i, d_i)$. That is, let S_i be a set of the first d_i nodes in K_i that receive the message in an optimal schedule of Instance $I(n_i, d_i)$. We can then simply throw out all transfers (both inter-cluster and local transfers) for the node in $K_i \setminus S_i$ because we only need d_i nodes in $I(d_i, d_i)$. □

Lemma 4.2 *Our algorithm takes at most $3OPT + 7$ time units.*

Proof Using Lemma 2.4 and Lemma 4.1, the first phase takes at most $2OPT + 7$ time units. In Phase 2, local broadcasting takes at most $\max_i \lceil \log \frac{n_i}{d_i} \rceil$ time units, which is at most OPT. □

Theorem 4.1 *There is a 3-approximation algorithm for broadcasting in the bounded degree model.*

4.2 Bounded Degree Model: Multicasting

In multicasting, only a subset M_i (possibly empty) of nodes in cluster K_i needs the message. Nodes in $K_i \setminus M_i$ may help passing the message around. Let n'_i be $|M_i|$. Note that $n_i \geq d_i$ and $n_i \geq n'_i$. Observe that although we may make use of nodes in $K_i \setminus M_i$, we never need more than d_i nodes in each cluster, because of the degree constraint in the number of inter-cluster transfers. Therefore, if $d_i \leq n'_i$, nodes in $K_i \setminus M_i$ are never needed.

Algorithm *Bounded Degree Multicast*
Set $n_i = \max(n'_i, d_i)$ and then arbitrarily select d_i nodes for each cluster, with priority given to nodes in M_i. Only the selected d_i nodes can send or receive the message in the first phase. Note that some of selected nodes are not in the multicast group.

1. Run the *LCF Multicast* algorithm on the selected nodes. Now there are $\min(d_i, n'_i)$ nodes having the message in each cluster that belong to the multicast group.
2. Perform local broadcasting for uninformed nodes in the multicast group.

After adjusting n_i, we have either $n'_i < n_i = d_i$ or $d_i \leq n'_i = n_i$. After selecting d_i nodes for each cluster, we create a valid multicast instance as described in Sect. 3 (without the degree constraint). Thus the algorithm works correctly.

Theorem 4.2 *There is a* 3 *approximation algorithm for multicasting in the bounded degree model.*

Proof By Lemma 3.2, the multicast steps takes $2OPT + 10$ time units. Moreover, the local broadcasting phase only needs to satisfy at most $n'_i - d_i$ nodes, which takes at most OPT time units. Thus the algorithm takes at most $3OPT + 10$ time units. □

4.3 Bounded Size Matching: Broadcasting

In this model, we bound the number of inter-cluster transfers that can be performed simultaneously. Let us assume that only B inter-cluster transfers are allowed at a time. Note that we can assume $B \leq \lfloor N/2 \rfloor$ since this is the maximum number of simultaneous transfers allowed by our matching-based communication model.

Algorithm *Bounded Size Broadcast*

1. Run $LCF(B)$ to make B copies of the message.
2. Every C time units we make B more copies by inter-cluster transfers until all clusters have at least one copy of the message.
3. Perform local broadcast to inform all the nodes in each cluster.

We show that the algorithm gives a 2-approximation for the bounded matching model. Let p_B be the number of global transfer steps $LCF(B)$ uses, and p_L be the number of global transfer steps in the second phase of the algorithm.

Theorem 4.3 *There is a* 2*-approximation algorithm for broadcasting in the bounded matching model.*

Proof It takes $2\log B + p_B C + p_L C + 4$ time units for broadcasting when only B inter-cluster transfers is allowed at a time whereas the optimal solution takes at least $(p_B + p_L - 1)(C - 1) + \lceil \log B \rceil$ time units. Using the proof technique in Lemma 2.4, we can prove that the algorithm takes at most $2OPT + 6$ time units, which gives the theorem. □

Remark Note that by setting $B = \lfloor N/2 \rfloor$, we can improve the broadcast time for the basic model (without any bound on the global transfers) by one. This is because in this algorithm we stop performing local transfers when the number of copies is $\lfloor N/2 \rfloor$ (as more copies cannot contribute to global transfers) and start global transfers.

4.4 Bounded Size Matching: Multicasting

Suppose that only a subset M_i of nodes in cluster K_i needs the message. Define $M = \{K_i | n'_i > 0 \text{ and } i > 0\}$ and $k' = |M|$. We assume $k' > B$ as otherwise we can use the *LCF Multicast* algorithm. We run $LCF(B)$ by using any node available. Then every C time units we make B more copies by inter-cluster transfers until all clusters in M have at least one copy of the message. Lastly do local broadcast to inform all the nodes in each cluster in M.

Theorem 4.4 *There is a* 2*-approximation algorithm for multicasting in the bounded size matching model.*

Proof It is easy to show that the algorithm described above takes at most $2OPT + 10$ time units, which gives the theorem. □

5 Postal Model

In the previous sections, we assumed that when node p_i sends a message to node p_j in another cluster, p_i is busy until p_j finishes receiving the message (this takes C time units). However, in some situations, it may not be realistic since p_i may become free after sending the message and does not have to wait until p_j receives the message.

In this section, we assume that a node is busy for only one time unit when it sends a message. The time a message arrives at the receiver depends on whether the sender and receiver are in the same cluster or not—it takes one time unit if it is a local transfer (within the cluster), and C time units if it is an inter-cluster transfer. We show that *LCF* gives a 3-approximation in this model.

While it is still an open question whether our analysis of 3-approximation is tight or not in the postal model, we present another broadcasting algorithm called *Interleaved LCF* in Sect. 5.2, which gives a 2-approximation in a certain condition. Recall that it is desirable to minimize the total number of global transfers as well as minimizing the broadcast time. Let OPT' denote the minimum broadcast time among all schedules that minimize the total number of global transfers. We show that the broadcast time of the schedule generated by *Interleaved LCF* is at most 2 times OPT'.

5.1 LCF in the Postal Model

LCF in this model works similarly as in the basic model but a node can initiate more than one global transfer in a global phase as senders are busy for only one time unit per global transfer. We define A_i, B_i, and L_i in a similar way as in the basic model, and then perform local and global transfers in turn. That is, we perform local transfers for $\log\lceil|A_0|\rceil$ time units, global transfers for C time units, local transfers for $\log\lceil|A_1|\rceil$ time units, and so on.

A more detailed description of the algorithm is as follows: Recall that $A_0 = K_0$. After finishing local transfers in K_0, all nodes in K_0 start global transfers. They can initiate a global transfer at every time unit. After C time units, n_0 clusters receive a message (denoted as L_1). Let A_1 (resp. B_1) be the biggest(resp. smallest) cluster among them. Now K_0 stops global transfers for $\lceil\log|A_1|\rceil$ time units. Global transfers already initiated continue to be done, and if a cluster receives t time unit later than A_1 then it may start its local transfers up to t time units later than A_1. For those $\lceil\log|A_1|\rceil$ time units of local transfers (or in general, $\lceil\log|A_i|\rceil$ time units in i-th local transfers), every cluster that received the message so far performs (only) local broadcasting and it will be idle even if it finishes local broadcasting earlier (in case that the size of the cluster is smaller than A_i). Note that A_i is the biggest cluster in L_i and therefore, $\lceil\log|A_i|\rceil$ time units will be spent for i-th local broadcasting. After $\lceil\log|A_i|\rceil$ time units, we have all clusters that have finished local broadcasting phase perform global transfers every time unit *for C time units*. Clusters that have not finished local transfers keep performing local broadcasting and then start global transfers. Repeat this until all nodes get informed. In general, we define L_i as clusters that receive messages in the first global transfers of i-th step (so they can participate in global phase of $(i+1)$-th step from the beginning). A_i (B_i) is the biggest (smallest) cluster in L_i. See Fig. 4(a) for example.

Suppose that the schedule requires p global transfer steps. Then it is easy to see that Lemmas 2.1 and 2.2 hold. There is one subtle case where there are some clusters that receive the message later than A_p by the transfers initiated in p-th global phases (see Fig. 4(b)).

Lemma 5.1 *The broadcast time of* LCF *is at most* $2\log N + pC + 3$ *when* A_p *is one of the last clusters that receive the message, and* $2\log N + pC + c' + 4$ *when there are some clusters that receive the message* c' *time units later than* A_p.

Proof The total time units taken for local transfers when A_p is one of the last clusters that receive the message is

$$\begin{aligned}
&\lceil\log n_0\rceil + \lceil\log|A_1|\rceil + \cdots + \log\lceil|A_p|\rceil \\
&\quad\le \log n_0 + \log|A_1| + \cdots + \log|A_p| + (p+1) \\
&\quad\le \log n_0 + \log|A_1| + \log|B_1| \cdots + \log|B_{p-1}| + (p+1) \\
&\quad= \log n_0 \cdot |A_1| \cdot |B_1| \cdots |B_{p-1}| + (p+1) \\
&\quad\le \log|A_1| + \log N_{p-1} + (p+1) \quad \text{(by Lemma 2.1)} \\
&\quad\le 2\log N + 3 \quad \text{(by Lemma 2.2).}
\end{aligned}$$

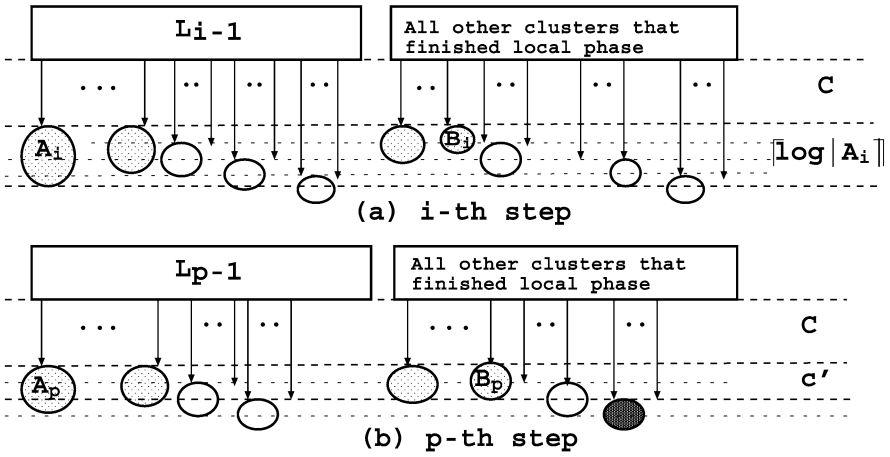


Fig. 4 (**a**) The figure shows i-th step of LCF in the postal model. In this example $C = 4$ so a node can initiate (at most) 4 global transfers in a row. *Dotted clusters* belong to L_i. (**b**) In p-th step, there can be some nodes that receive messages later than A_p. The *dark circle* is the last cluster that receives the message

Therefore, the broadcast time of the schedule is at most $2\log N + pC + 3$.

We now deal with the case when there are other clusters that receive the message later than A_p. Let A_{p+1} denote the biggest cluster that receives the message last, and it receives the message c' time units later than A_p ($c' < C$ since otherwise we would have another global phase). We need additional $c' + \lceil \log |A_{p+1}| \rceil (\leq c' + \lceil \log |B_p| \rceil)$ time units to finish. Since $n_0 \cdot |B_1| \cdots |B_p| \leq N_p$ (by Lemma 2.1) the broadcast time of *LCF* is at most $2\log N + pC + c' + 4$. □

Lemma 5.2 *If we assume that local transfers take zero unit of time, the broadcast time is at least pC when A_p is one of the last clusters that receive the message. If there are some clusters that receive the message c' time units later than A_p then the broadcast time is at least $pC + c'$.*

Proof Consider the schedule given by *LCF* when local transfers take zero units of time. It is easy to see that *LCF* gives an optimal solution in the case when A_p is one of the last clusters that receive the message. Moreover it is the same as the schedule by the *LCF* if we ignore local transfer steps. Since *LCF* needs p global steps, pC is a lower bound on the optimal solution.

Suppose now that there are some clusters that receive the message c' time units later than A_p in *LCF*. Then A_{p+1} receives the message c' time units later than A_p in the model with zero local transfer cost (since the schedule can be obtained by ignoring local steps). Therefore $pC + c'$ is a lower bound on the optimal solution. □

By Lemmas 5.1 and 5.2, we have the following theorem.

Theorem 5.1 *Algorithm LCF gives a* 3*-approximation for broadcasting in the postal model* (*with additive term of* 4).

It is easy to see the following theorem for multicasting.

Theorem 5.2 *There is a* 3*-approximation for multicasting in the postal model.*

5.2 Interleaved LCF

We present another algorithm called *Interleaved LCF*, which gives a 2-approximation *among all the schedules that use the minimum number of global transfers*.

Algorithm *Interleaved LCF*
At every two time units, a node that has the message alternately performs the following two steps.

1. Local transfer: if there is any node in the same cluster that has not received the message, then send the message to it.
2. Global transfer: if there is any cluster that has not received the message, choose the biggest cluster among them and send the message to a node in the cluster.

Let us only consider a set of schedules (denoted as S) that minimize the total number of global transfers. Note that schedules in S have the property that each cluster receives only one message from outside ($k-1$ in total). Let OPT_S be the minimum broadcast time among all schedules in S. Then we have the following theorem (see Appendix A for the proof).

Theorem 5.3 *The broadcast time of Interleaved LCF is at most* 2 *times* OPT_S.

6 Conclusion and Open Problems

The main objective of this work is to establish worst case approximation guarantees for a simple broadcasting heuristic. Broadcasting in graphs that are cliques, is usually straightforward. Graphs that are completely arbitrary, make the problem not only computationally intractable, but hard to approximate well in a worst case sense. We propose studying this problem in an intermediate model, where nodes can communicate quickly with a subset of closeby nodes, and the cost to communicate with everyone else is an order of magnitude higher (and assumed to be uniform). This suggests that having a more reasonable communication model is the way to approach the problem.

We note that it is not known if the problem of minimizing the broadcast time (in any of these models) is *NP*-hard or not. In addition, we are currently examining generalizations of this model when the communication time in different clusters may be different due to different speed networks and different speed processors. Another interesting generalized model would be to have a "two level LogP" model with different parameters for the local networks (intra-cluster) and global networks (inter-cluster).

Appendix A: Analysis of *Interleaved LCF*

Lemma A.1 *There is a schedule in S with broadcast time OPT_S in which for any pair of clusters K_i, K_j $(n_i > n_j)$, K_i receives a message no later than K_j.*

Proof Given a schedule in S with broadcast time OPT_S, if there is a pair of clusters K_i, K_j $(n_i > n_j)$ and K_i receives a message at time t_i and K_j receives a message at time t_j $(t_i > t_j)$, then we can modify the schedule so that K_i receives the message no later than K_j without increasing the broadcast time.

At t_j K_i (instead of K_j) receives the message. K_i can do all transfers that K_j does till time t_i. At time t_i, K_j receives a message. Let x_t nodes in K_i received the message just after time t in the *original* schedule. Similarly, let y_t nodes in K_j received the message just after time t. Then $x_{t_i} = 1$ and $y_{t_i} \leq n_j$. Note that we cannot swap the roles of two clusters just after t_i since K_i has y_{t_i} messages and K_j has only one message. Therefore, K_i should keep performing transfers as if it is K_j for some time. Let t' be the last time when $x_{t'} \leq y_{t'}$. That is, $x_{t'+1} > y_{t'+1}$.

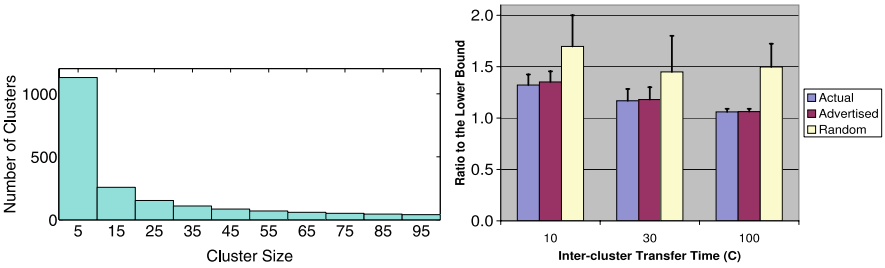
At time $t' + 1$ we need to carefully choose which transfers we should do. Note that just before time $t' + 1$, K_i has $y_{t'}$ messages and K_j has $x_{t'}$ messages. In K_i we choose $x_{t'+1} - y_{t'}$ nodes to make local transfers so that after $t' + 1$, K_i has $x_{t'+1}$ copies of message. Since $x_{t'+1} \leq 2x_{t'} \leq 2y_{t'}$, $x_{t'+1} - y_{t'} \leq y_{t'}$ and therefore, we have enough nodes to choose. Similarly, in K_j $y_{t'+1} - x_{t'}$ nodes do local transfers so that K_j has $y_{t'+1}$ after $t' + 1$. The total number of global transfers coming from K_i and K_j in the original schedule is at most $x_{t'} + y_{t'} - (x_{t'+1} - x_{t'}) - (y_{t'+1} - y_{t'}) = x_{t'} + y_{t'} - (y_{t'+1} - x_{t'}) - (x_{t'+1} - y_{t'})$ and this is exactly the number of remaining nodes. Therefore, we have the remaining nodes enough to make global transfers. After $t' + 1$, we can do transfers as in the original schedule. □

We can now consider schedules with the property in Lemma A.1 only. Due to the property, a node knows the receiver to send a message when it performs a global transfer—the largest cluster that has not received any message. The only thing a node needs to decide at each time is whether it will make a local transfer or global transfer. By performing local and global transfer alternatively, we can bound the broadcast time by a factor of two.

Proof of Theorem 5.3 Given an optimal schedule with the property in Lemma A.1, modify the schedule so that each operation takes 2 units of time. That is, if a node performs a local transfer then it is idle in the next time slot and if a node performs a global transfer, it is idle in the previous time slot. The broadcast time of the modified schedule is at most 2 times the optimal. It is easy to see that the schedule by *Interleaved LCF* should not be worse than the modified schedule since in *Interleaved LCF*, the nodes performs local and global transfers alternatively with no idle time. □

Appendix B: Experimental Results

One issue with our broadcasting protocol is that it assumes knowledge of the sizes of the clusters. In some applications, the cluster sizes may not be known accurately in advance. What effect can this have on the broadcasting algorithm Largest Cluster



(a) A histogram of the cluster sizes (b) Results

Fig. 5 (**a**) Distribution of cluster sizes. (**b**) The ratio of the number of time units taken by the algorithms to the lower bound, averaged over 5 inputs, with different values of C (10, 30, 100, and 1000). The maximum ratio appears on the top of each bar

First for example? In the simplest model, we study the effect of having inaccurate information regarding the sizes of the clusters.

We run the *LCF* algorithm using the correct cluster sizes; in addition, we run the *LCF* algorithm by basing the order on advertised sizes that may be off by a factor of 2. For example, for each cluster we let the advertised size be fixed, but change the actual size randomly by either doubling it, or halving it. We now run the protocol where the cluster ordering is made by using the advertised size. We found that there is hardly any change in the performance of the broadcasting algorithm.

In the following experiment, we have roughly 2000 clusters and each cluster has advertised size between 1 and 100 nodes. We choose the actual cluster size using a Zipf distribution, i.e., Prob(a cluster having size i) $= \frac{c}{i^{1-\theta}}$, $\forall i = 1, \ldots, 100$ and $0 \leq \theta \leq 1$, where $c = \frac{1}{H_M^{1-\theta}}$, $H_M^{1-\theta} = \sum_{j=1}^{100} \frac{1}{j^{1-\theta}}$, and θ determines the degree of skewness. We assign θ to be 0. Figure 5(a) shows a histogram of the cluster sizes. We also let C be a parameter and vary this from 10 to 1000. This is a reasonable range as we expect the time to send a message across clusters to be within this range, assuming that the time to send a message within a cluster takes unit time. (If it takes a few milliseconds to send a message from one machine to another locally, it may take upto a few hundred, or thousand milliseconds to send a message to a machine belonging to another cluster.)

(*Actual*) records the broadcast time when the cluster sizes are the same as their advertised sizes. (*Advertised*) records the broadcast time when the cluster sizes are either half or double their advertised sizes (this choice is made independently and randomly for each cluster). We also compare both these methods to the broadcast time if the algorithm were to use a random ordering of the clusters (*Random*). We also illustrate these times and compare them to the best lower bounds from Sect. 2.1.

B.1 Results

As shown in Fig. 5(b), we found that (*Actual*) performs the best. This is unsurprising because it has completely accurate information. Note that in all the experiments we ran, it performs at most 1.5 times the lower bound, which is smaller than the theoretical bound of 2. (However, we believe that the broadcasting time is a much closer

to the optimal solution even though our lower bounds are not strong enough to argue this.) Moreover, (*Advertised*) performs very close to (*Actual*) (it takes only one more time unit in all instances). This behavior shows that one needs not to have completely accurate information on the size of the clusters for our Largest Cluster First algorithm to perform well. On the other hand, if the algorithm uses a random ordering of the clusters, it takes, on average, at least 24% more than (*Actual*). In addition, as C increases, (*Actual*) performs closer and closer to the optimal.

References

1. Bar-Noy, A., Kipnis, S.: Designing broadcast algorithms in the postal model for message-passing systems. Math. Syst. Theory **27**(5), (1994)
2. Bar-Noy, A., Guha, S., Naor, J.S., Schieber, B.: Message multicasting in heterogeneous networks. SIAM J. Comput. **30**(2), 347–358 (2001)
3. Bhat, P.B., Raghavendra, C.S., Prasanna, V.K.: Efficient collective communication in distributed heterogeneous systems. J. Parallel Distrib. Comput. **63**(3), 251–263 (2003)
4. Bruck, J., Dolev, D., Ho, C., Rosu, M., Strong, R.: Efficient message passing interface(mpi) for parallel computing on clusters of workstations. Parallel Distrib. Comput. **40**, 19–34 (1997)
5. Culler, D.E., Karp, R.M., Patterson, D.A., Sahay, A., Schauser, K.E., Santos, E., Subramonian, R., von Eicken, T.: LogP: Towards a realistic model of parallel computation. In: Proceedings 4th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming, pp. 1–12 (1993)
6. Elkin, M., Kortsarz, G.: A combinatorial logarithmic approximation algorithm for the directed telephone broadcast problem. SIAM J. Comput. **35**(3), 672–689 (2005)
7. Elkin, M., Kortsarz, G.: Sublogarithmic approximation for telephone multicast. J. Comput. Syst. Sci. **72**(4), 648–659 (2006)
8. Foster, I., Kesselman, K.: The Grid: Blueprint for a New Computing Infrastructure. Morgan Kaufmann, San Mateo (1998)
9. Gropp, W., Lusk, E., Doss, N., Skjellum, A.: A high-performance, portable implementation of the mpi: a message passing interface standard. Parallel Comput. **22**, 789–828 (1996)
10. Hedetniemi, S.M., Hedetniemi, S.T., Liestman, A.L.: A survey of broadcasting and gossiping in communication networks. Networks **18**, 319–349 (1988)
11. Husbands, P., Hoe, J.C.: Mpi-start: delivering network performance to numerical applications. In: Supercomputing '98: Proceedings of the 1998 ACM/IEEE Conference on Supercomputing (CDROM), pp. 1–15. Washington, DC, USA, 1998. IEEE Computer Society, Los Alamitos (1998)
12. Karp, R., Sahay, A., Santos, E., Schauser, K.E.: Optimal broadcast and summation in the LogP model. In: Proceedings of 5th Annual Symposium on Parallel Algorithms and Architectures, pp. 142–153 (1993)
13. Kielmann, T., Bal, H., Gorlatch, S.: Bandwidth-efficient collective communication for clustered wide area systems. In: International Parallel and Distributed Processing Symposium, pp. 492–499. Washington, DC, USA, 2000. IEEE Computer Society, Los Alamitos (2000)
14. Kielmann, T., Hofman, R.F.H., Bal, H.E., Plaat, A., Bhoedjang, R.A.F.: Magpie: Mpiâs collective communication operations for clustered wide area systems. In: ACM SIGPLAN Notices, pp. 131–140 (1999)
15. Lowekamp, B.B., Beguelin, A.: Eco: Efficient collective operations for communication on heterogeneous networks. In: International Parallel Processing Symposium (IPPS), pp. 399–405. Honolulu, HI (1996)
16. Message passing interface forum. http://www.mpi-forum.org/index.html
17. Patterson, D.A., Culler, D.E., Anderson, T.E.: A case for NOWs (networks of workstations). IEEE Micro **15**(1), 54–64 (1995)
18. Pruyne, J., Livny, M.: Interfacing condor and pvm to harness the cycles of workstation clusters. J. Future Gener. Comput. Syst. **12**(1), 53–65 (1996)
19. Richards, D., Liestman, A.L.: Generalization of broadcasting and gossiping. Networks **18**, 125–138 (1988)
20. Williams, T., Parsons, R.: Exploiting hierarchy in heterogeneous environments. In: IEEE/ACM IPDPS 2001, pp. 140–147. IEEE Press, New York (2001)