

On Generalized Gossiping and Broadcasting^{*}

(Extended Abstract)

Samir Khuller, Yoo-Ah Kim, and Yung-Chun (Justin) Wan

Department of Computer Science and Institute for Advanced Computer Studies,
University of Maryland, College Park, MD 20742.
{samir,ykim,ycwan}@cs.umd.edu

Abstract. The problems of gossiping and broadcasting have been widely studied. The basic gossip problem is defined as follows: there are n individuals, with each individual having an item of gossip. The goal is to communicate each item of gossip to every other individual. Communication typically proceeds in rounds, with the objective of minimizing the number of rounds. One popular model, called the telephone call model, allows for communication to take place on any chosen matching between the individuals in each round. Each individual may send (receive) a single item of gossip in a round to (from) another individual. In the broadcasting problem, one individual wishes to broadcast an item of gossip to everyone else. In this paper, we study generalizations of gossiping and broadcasting. The basic extensions are: (a) each item of gossip needs to be broadcast to a specified subset of individuals and (b) several items of gossip may be known to a single individual. We study several problems in this framework that generalize gossiping and broadcasting. Our study of these generalizations was motivated by the problem of managing data on storage devices, typically a set of parallel disks. For initial data distribution, or for creating an initial data layout we may need to distribute data from a single server or from a collection of sources.

1 Introduction

The problems of Gossiping and Broadcasting have been the subject of extensive study [21,15,17,3,4,18]. These play an important role in the design of communication protocols in various kinds of networks. The *gossip problem* is defined as follows: there are n individuals. Each individual has an item of gossip that they wish to communicate to everyone else. Communication is typically done in rounds, where in each round an individual may communicate with at most one other individual (also called the telephone model). There are different models that allow for the full exchange of all items of gossip known to each individual in a single round, or allow the sending of only one item of gossip from one to

^{*} Full paper is available at

<http://www.cs.umd.edu/projects/smart/papers/multicast.pdf>. This research was supported by NSF Awards CCR-9820965 and CCR-0113192.

the other (half-duplex) or allow each individual to send an item to the individual they are communicating with in this round (full-duplex). In addition, there may be a communication graph whose edges indicate which pairs of individuals are allowed to communicate in each round. (In the classic gossip problem, communication may take place between any pair of individuals; in other words, the communication graph is the complete graph.) In the *broadcast problem*, one individual needs to convey an item of gossip to every other individual. The two parameters typically used to evaluate the algorithms for this problem are: the number of communication rounds, and the total number of telephone calls placed.

The problems we study are generalizations of the above mentioned gossiping and broadcasting problems. The basic generalizations we are interested in are of two kinds (a) each item of gossip needs to be communicated to only a subset of individuals, and (b) several items of gossip may be known to one individual. Similar generalizations have been considered before [23,25]. (In Section 1.2 we discuss in more detail the relationships between our problem and the ones considered in those papers.)

There are four basic problems that we are interested in. Before we define the problems formally, we discuss their applications to the problem of creating data layouts in parallel disk systems. The communication model we use is the half-duplex telephone model, where only one item of gossip may be communicated between two communicating individuals during a single round. Each individual may communicate (either send or receive an item of data) with at most one other individual in a round. This model best captures the connection of parallel storage devices that are connected on a network and is most appropriate for our application.

We now briefly discuss applications for these problems, as well as prior related work on data migration. To deal with high demand, data is usually stored on a parallel disk system. Data objects are often replicated within the disk system, both for fault tolerance as well as to cope with demand for popular data [29, 5]. Disks typically have constraints on storage as well as the number of clients that can simultaneously access data from it. Approximation algorithms have been developed [26,27,12,19] to map known demand for data to a specific data layout pattern to maximize utilization¹. In the layout, we not only compute how many copies of each item we need, but also a layout pattern that *specifies the precise subset of items on each disk*. The problem is *NP-hard*, but there is a polynomial time approximation scheme [12]. Hence given the relative demand for data, the algorithm computes an almost optimal layout. For example, we may wish to create this layout by copying data from a single source that has all the data initially. Or the data may be stored at different locations initially—these considerations lead to the different problems that we consider.

In our situation, each individual models a **disk** in the system. Each item of gossip is a **data item** that needs to be transferred to a set of disks. If each disk

¹ Utilization refers to the total number of clients that can be assigned to a disk that contains the data they want.

had exactly one data item, and needs to copy this data item to every other disk, then it is exactly the problem of gossiping.

Different communication models can be considered based on how the disks are connected. We use the same model as in the work by [13,1] where the disks may communicate on any matching; in other words, the underlying communication graph is complete. For example, *Storage Area Networks* support a communication pattern that allows for devices to communicate on a specified matching.

Suppose we have N disks and Δ data items. The problems we are interested in are:

1. **Single-source broadcast.** There are Δ data items stored on a single disk (the source). We need to broadcast all items to all $N - 1$ remaining disks.
2. **Single-source multicast.** There are Δ data items stored on a single disk (the source). We need to send data item i to a specified subset D_i of disks. Figure 1 gives an example when Δ is 4.
3. **Multi-source broadcast.** There are Δ data items, each stored separately at a single disk. These need to be broadcast to all disks. We assume that data item i is stored on disk i , for $i = 1 \dots \Delta$.
4. **Multi-source multicast.** There are Δ data items, each stored separately at a single disk. Data item i needs to be sent to a specified subset D_i of disks. We assume that data item i is stored on disk i , for $i = 1 \dots \Delta$.

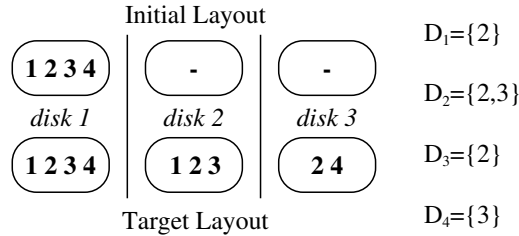


Fig. 1. An initial and target layouts, and their corresponding D_i 's of a single-source multicast instance.

We do not discuss the first problem in any detail since this was solved by [8, 10]. For the multi-source problems, there is a sub-case of interest, namely when the source disks are not in any subset D_i . For this case we can develop better bounds (details omitted).

1.1 Contributions

In Section 2 we define the basic model of communication and the notation used in the paper. Let N be the number of disks and Δ be the number of items. The main results that we show in this paper are:

Theorem 1.1. *For the single-source multicast problem we design a polynomial time algorithm that outputs a solution where the number of rounds is at most $OPT + \Delta$.*

Theorem 1.2. *For the multi-source broadcast problem we design a polynomial time algorithm that outputs a solution where the number of rounds is at most $OPT + 3$.*

Theorem 1.3. *For the multi-source multicast problem we design a polynomial time algorithm that outputs a solution where the number of rounds is at most $4OPT + 3$. We also show that this problem is NP -hard.*

For all the above algorithms, we move data only to disks that need the data. Thus we use no bypass (intermediate) nodes as holding points for the data. If bypass nodes are allowed, we have this result:

Theorem 1.4. *For the multi-source multicast problem allowing bypass nodes we design a polynomial time algorithm that outputs a solution where the number of rounds is at most $3OPT + 6$.*

1.2 Related Work

One general problem of interest is the **data migration problem** when data item i resides in a specified (source) subset S_i of disks, and needs to be moved to a (destination) subset D_i . This problem is more general than the Multi-Source multicast problem where we assumed that $|S_i| = 1$ and that all the S_i 's are disjoint. For the data migration problem we have developed a 9.5-approximation algorithm [20]. While this problem is a generalization of all the problems we study in this paper (and clearly also NP -hard since even the special case of multi-source multicast is NP -hard), the bounds in [20] are not as good. The methods used for single-source multicast and multi-source broadcast are completely different from the algorithm in [20]. Using the methods in [20] one cannot obtain additive bounds from the optimal solution. The algorithm for multi-source multicast presented here is a simplification of the algorithm developed in [20], and we also obtain a much better approximation factor of 4.

Many generalizations of gossiping and broadcasting have been studied before. For example, the paper by Liben-Nowell [23] considers a problem very similar to multi-source multicast with $\Delta = N$. However, the model that he uses is different than the one that we use. In his model, in each telephone call, a pair of users can exchange all the items of gossip that they know. The objective is to simply minimize the total number of phone calls required to convey item i of gossip to set D_i of users. In our case, since each item of gossip is a data item that might take considerable time to transfer between two disks, we cannot assume that an arbitrary number of data items can be exchanged in a single round. Several other papers use the same telephone call model [2,7,14,18,30]. Liben-Nowell [23] gives an exponential time exact algorithm for the problem.

Other related problems that have been studied are the set-to-set gossiping problem [22,25] where we are given two possibly intersecting sets A and B of gossipers and the goal is to minimize the number of calls required to inform all gossipers in A of all the gossip known to members in B . The work by [22] considers minimizing both the number of rounds as well as the total number of

calls placed. The main difference is that in a single round, an arbitrary number of items may be exchanged. For a complete communication graph they provide an exact algorithm for the minimum number of calls required. For a tree communication graph they minimize the number of calls or number of rounds required. Liben-Nowell [23] generalizes this work by defining for each gossip i the set of relevant gossip that they need to learn. This is just like our multi-source multicast problem with $\Delta = N$, except that the communication model is different, as well as the objective function. The work by [9] also studies a set to set broadcast type problem, but the cost is measured as the total cost of the broadcast trees (each edge has a cost). The goal is not to minimize the number of rounds, but the total cost of the broadcast trees. In [11] they also define a problem called scattering which involves one node broadcasting distinct messages to all the other nodes (very much like our single source multicast, where the mutlicast groups all have size one and are disjoint).

As mentioned earlier, the single source broadcast problem using the same communication model as in our paper was solved by [8,10].

2 Models and Definitions

We have N disks and Δ data items. Note that after a disk receives item i , it can be a source of item i for other disks that have not received the item as yet. Our goal is to find a schedule using the minimum number of rounds, that is, to minimize the total amount of time to finish the schedule. We assume that the underlying network is connected and the data items are all the same size, in other words, it takes the same amount of time to migrate an item from one disk to another. The crucial constraint is that each disk can participate in the transfer of only one item—either as a sender or receiver. Moreover, as we do not use any bypass nodes, all data is only sent to disks that desire it.

Our algorithms make use of a known result on edge coloring of multi-graphs. Given a graph G with max degree Δ_G and multiplicity μ the following result is known (see [6] for example). Let χ' be the edge chromatic number of G .

Theorem 2.1. (*Vizing [31]*) *If G has no self-loops then $\chi' \leq \Delta_G + \mu$.*

3 Single-Source Multicasting

In this section, we consider the case where there is one source disk s that has all Δ items and others do not have any item in the beginning. For the case of *broadcasting* all items, it is known that there is a schedule which needs $2\Delta - 1 + \lfloor \log N \rfloor$ rounds for odd N and $\lceil \frac{\Delta(N-1) - 2^{\lfloor \log_2 N \rfloor} + 1}{\lfloor N/2 \rfloor} \rceil + \lfloor \log N \rfloor$ rounds for even N [8,10] and this is optimal. We develop an algorithm that can be applied when D_i is an arbitrary subset of disks. The number of rounds required by our algorithm is at most $\Delta + OPT$ where OPT is the minimum number of rounds required for this problem. Our algorithm is obviously a 2-approximation for the problem, since Δ is a lower bound on the number of rounds required by the optimal solution.

3.1 Outline of the Algorithm

Without loss of generality, we assume that $|D_1| \geq |D_2| \geq \dots \geq |D_\Delta|$ (otherwise renumber the items). Let $|D_i| = 2^{d_i^1} + 2^{d_i^2} + \dots + 2^{d_i^{m_i}}$ where $d_i^j (j = 1, 2, \dots, m_i)$ are integers and $d_i^j > d_i^{j+1}$. (In other words, we consider the bit representation of each $|D_i|$ value.)

Our algorithm consists of two phases.

Phase I. In the first phase, we want to make exactly $\lfloor |D_i|/2 \rfloor$ copies for all items i . At the t -th round, we do the following:

1. If $t \leq \Delta$, copy item t from source s to a disk in D_t .
2. For items $j (j < t)$, double the number of copies unless the number of copies reaches $\lfloor |D_j|/2 \rfloor$. In other words, every disk having an item j makes another copy of it if the number of copies of item j is no greater than $2^{d_j^1-2}$, and when it becomes $2^{d_j^1-1}$, then only $\lfloor |D_j|/2 \rfloor - 2^{d_j^1-1}$ disks make copies, and thus the number of copies of item i becomes $\lfloor |D_i|/2 \rfloor$.

Phase II. At t -th round, we finish the migration of item t . Each item j has $\lfloor |D_j|/2 \rfloor$ copies. We finish migrating item t by copying from the current copies to the remaining $\lfloor |D_t|/2 \rfloor$ disks in D_t which did not receive item t as yet, and we use the source disk if $|D_t|$ is odd.

Figure 2 shows an example of data transfers taken in Phase 1.

where $|D_1|, |D_2|$ and $|D_3|$ are 8, 6 and 4, respectively. It is easy to see that Phase II can be scheduled without conflicts because we deal with only one item each round. But in Phase I, migration of several items happen at the same time and D_i 's can overlap. Therefore, we may not be able to satisfy the requirement of each round if we arbitrarily choose the disks to receive items. We show that we can finish Phase I successfully without conflicts by carefully choosing disks.

3.2 Details of Phase I

Let D_i^p be the disks in D_i that participate in either sending or receiving item i at the $(i + p)$ -th round. D_i^0 is the first disk receiving i from the source s and

$$|D_i^p| = \begin{cases} 2^p & \text{if } p \leq d_i^1 - 1 \\ 2^{\lfloor \frac{|D_i|}{2} \rfloor} - 2^{d_i^1} & \text{if } p = d_i^1 \end{cases}$$

At $(i + p)$ -th round, disks in $D_j^{i+p-j} (i + p - d_j^1 \leq j \leq \min(i + p, \Delta))$ either send or receive item j at the same time. To avoid conflicts, we decide which disks belong to D_i^p before starting migration. If we choose disks from $D_i \cap D_j$ for $D_i^p (j > i)$, it may interfere with the migration of D_j . Therefore, when we build D_i^p , we consider $D_j^{p'}$ where $j > i$ and $p' \leq p$. Also note that since each disk receiving an item should have its corresponding sender, the half of D_i^p should have item i as senders and another half should not have item i as receivers.

We build D_Δ^p first. Choose $2\lfloor |D_\Delta|/2 \rfloor - 2^{d_\Delta^1}$ disks for $D_\Delta^{d_\Delta^1}$ and $2^{d_\Delta^1-1}$ disks for $D_\Delta^{d_\Delta^1-1}$ from D_Δ . When we choose $D_\Delta^{d_\Delta^1-1}$, we should include the half of $D_\Delta^{d_\Delta^1}$

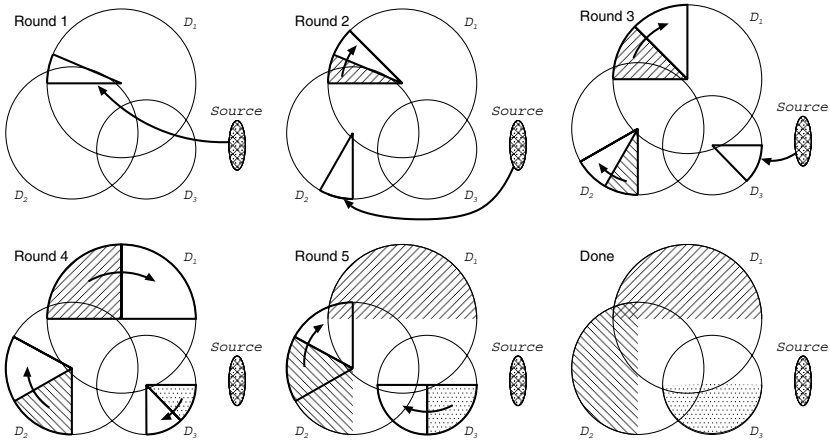


Fig. 2. An example of Phase I when all $|D_i|$ are even

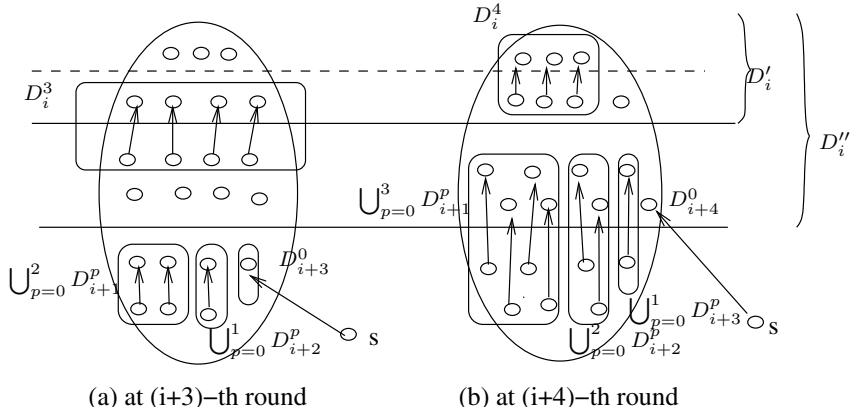


Fig. 3. How disks in D_i behave in Phase I where $|D_i| = 2^4 + 2^2 + 2^1$

(that will be senders at $(\Delta + d_\Delta^1)$ -th round) and exclude the remaining half of $D_\Delta^{d_\Delta^1}$ (that will be receivers at $(\Delta + d_\Delta^1)$ -th round). And then build D_Δ^p ($p < d_\Delta^1 - 1$) by taking any subset of D_Δ^{p+1} .

Now given $D_j^{p'}$ ($i < j \leq \Delta$), we decide D_i^p as follows: Define D_i' to be disks in D_i which do not have any item $j (> i)$ after $(i + d_i^1)$ -th round. In the same way, define D_i'' to be disks in D_i which do not have any item $j (> i)$ after $(i + d_i^1 - 1)$ -th round. Formally, since all disks in $\bigcup_{p=0}^{p'} D_j^p$ have item j after $(j + p')$ -th rounds, $D_i' = D_i - \bigcup_{j=i+1}^\Delta (\bigcup_{p=0}^{i+d_i^1-j} D_j^p)$ and $D_i'' = D_i - \bigcup_{j=i+1}^\Delta (\bigcup_{p=0}^{i+d_i^1-1-j} D_j^p)$. As shown in Figure 3, we choose $D_i^{d_i^1}$ from D_i' and also $D_i^{d_i^1-1}$ from D_i'' , by which we can avoid conflicts. Also, half of $D_i^{d_i^1}$ should be included in $D_i^{d_i^1-1}$

(to be senders) and the remaining half should be excluded from $D_i^{d_i^1-1}$ (to be receivers). We make D_i^p ($p < d_i^1 - 1$) by choosing any subset of disks from D_i^{p+1} .

Lemma 3.1. *We can find a migration schedule by which we perform every round in phase I without conflicts.*

3.3 Analysis

We prove that our algorithm uses at most Δ more rounds than the optimal solution for single-source multicasting. Let us denote the optimal makespan of an migration instance I as $C(I)$.

Lemma 3.2. *For any migration instance I , $C(I) \geq \max_{1 \leq i \leq \Delta} (i + \lfloor \log |D_i| \rfloor)$.*

Lemma 3.3. *The total makespan of our algorithm is at most $\max_{1 \leq i \leq \Delta} (i + \lfloor \log |D_i| \rfloor) + \Delta$.*

Theorem 3.4. *The total makespan of our algorithm is at most the optimal makespan plus Δ .*

Corollary 3.5. *We have a 2-approximation algorithm for the single-source multicasting problem.*

4 Multi-source Broadcasting

We assume that we have N disks. Disk i , $1 \leq i \leq \Delta$, has an item numbered i . The goal is to send each item i to all N disks, for all i . We present an algorithm which performs no more than 3 extra rounds than the optimal solution.

4.1 Algorithm Multi-source Broadcast

1. We divide N disks into Δ disjoint sets G_i such that disk $i \in G_i$, for all $i = 1 \dots \Delta$. Let q be $\lfloor \frac{N}{\Delta} \rfloor$ and r be $N - q\Delta$. $|G_i| = q + 1$ for $i = 1 \dots r$, and $|G_i| = q$ for $i = r+1 \dots \Delta$. Every disk in G_i can receive item i using $\lceil \log |G_i| \rceil$ rounds by doubling the item in each round. Since the sets G_i are disjoint, every disk can receive an item belongs to its group in $\lceil \log \frac{N}{\Delta} \rceil$ rounds.
2. We divide all N disks into $q - 1$ groups of size Δ by picking one disk from each G_i , and one group of size $\Delta + r$ which consists of all remaining disks.
3. Consider the first $q - 1$ gossiping groups; each group consists of Δ disks, with each having a distinct item. Using gossiping algorithm in [4], every disk in the first $q - 1$ groups can receive all Δ items in 2Δ rounds².
4. Consider the last gossiping group, there are exactly two disks having items $1, \dots, r$, while there is exactly one disk having item $r + 1, \dots, \Delta$. If r is zero, we can finishes all transfers in 2Δ rounds using algorithm in [4]. For non-zero r , we claim that all disks in this gossiping group can receive all items in 2Δ rounds.

² The number of rounds required is 2Δ if Δ is odd, otherwise it is $2(\Delta - 1)$

We divide the disks in this gossiping group into 2 groups, G_X and G_Y of size $\Delta - \lfloor \frac{\Delta-r}{2} \rfloor$ and $r + \lfloor \frac{\Delta-r}{2} \rfloor$ respectively. Note that $|G_Y| + 1 \geq |G_X| \geq |G_Y|$. Exactly one disk having items $1, \dots, r$ appear in each group, disks having item $r + 1, \dots, \Delta - \lfloor \frac{\Delta-r}{2} \rfloor$ appear in G_X , and the remaining disks (having items $\Delta - \lfloor \frac{\Delta-r}{2} \rfloor + 1, \dots, \Delta$) appear in G_Y . Note that the size of the two groups differ by at most 1. The general idea of the algorithm is as follows (The details of these step are non-trivial and covered in the proof of Lemma 4.1):

- a) Algorithm in [4] is applied to each group in parallel. After this step, each disk has all items belong to its group.
- b) In each round, disks in G_Y send item i to disks in G_X , where i is $\Delta - \lfloor \frac{\Delta-r}{2} \rfloor + 1, \dots, \Delta$. Note that only disks in G_Y have these items, but not the disks in G_X . Since the group sizes diff by at most 1, the number of rounds required is about the same as the number of items transferred.
- c) The step is similar to the above step but in different direction. Item i , where i is $r + 1, \dots, \Delta - \lfloor \frac{\Delta-r}{2} \rfloor$, are copied to G_Y .

Thus, our algorithm takes $\lceil \log \frac{N}{\Delta} \rceil + 2\Delta$ rounds.

4.2 Analysis

Lemma 4.1. *For a group of disks of size $\Delta + r$, where $1 \leq r < \Delta$, if every disk has one item, exactly 2 disks have item $1, \dots, r$, and exactly 1 disk has item $r + 1, \dots, \Delta$, all disks can receive all Δ items in 2Δ rounds.*

Theorem 4.2. *The makespan time of any migration instance of multi-source broadcasting is at least $\lceil \log \frac{N}{\Delta} \rceil + 2(\Delta - 1)$.*

Thus, our solution takes no more than 3 rounds than the optimal.

5 Multi-source Multicasting

We assume that we have N disks. Disk i , $1 \leq i \leq \Delta \leq N$, has data item i . The goal is to copy item i to a subset D_i of disks that do not have item i . (Hence $i \notin D_i$). We could show that finding a schedule with the minimum number of rounds is NP -hard. In this section we present a polynomial time approximation algorithm for this problem. The approximation factor of this algorithm is 4.

We first define β as $\max_{i=1 \dots N} |\{j | i \in D_j\}|$. In other words, β is an upper bound on the number of different sets D_j , that a disk i may belong to. Note that β is a lower bound on the optimal number of rounds, since the disk that attains the max, needs at least β rounds to receive all the items j such that $i \in D_j$, since it can receive at most one item in each round.

The algorithm will first create a small number of copies of each data item j (the exact number of copies will be dependent on $|D_j|$). We then assign each newly created copy to a set of disks in D_j , such that it will be responsible for providing item j to those disks. This will be used to construct a transfer graph,

where each directed edge labeled j from v to w indicates that disk v must send item j to disk w . We will then use an edge-coloring of this graph to obtain a valid schedule [6]. The main difficulty here is that a disk containing an item is its source, is also the destination for several other data items.

Algorithm Multi-source Multicast

1. We first compute a disjoint collection of subsets $G_i, i = 1 \dots \Delta$. Moreover, $G_i \subseteq D_i$ and $|G_i| = \lfloor \frac{|D_i|}{\beta} \rfloor$. (In Lemma 5.1, we will show how such G_i 's can be obtained.)
2. Since the G_i 's are disjoint, we have the source for item i (namely disk i) send the data to the set G_i using $\lceil \log |D_i| \rceil + 1$ rounds as shown in Lemma 5.2. Note that disk i may itself belong to some set G_j . Let $G'_i = \{i\} \cup G_i$. In other words, G'_i is the set of disks that have item i at the end of this step.
3. We now create a transfer graph as follows. Each disk is a node in the graph. We add directed edges from each disk in G'_i to disks in $D_i \setminus G_i$ such that the out-degree of each node in G'_i is at most $\beta - 1$ and the in-degree of each node in $D_i \setminus G_i$ is 1. (In Lemma 5.3 we show how that this can be done.) This ensures that each disk in D_i receives item i , and that each disk in G'_i does not send out item i to more than $\beta - 1$ disks.
4. We now find an edge coloring of the transfer graph (which is actually a multigraph) and the number of colors used is an upper bound on the number of rounds required to ensure that each disk in D_j gets item j . (In Lemma 5.4 we derive an upper bound on the degree of each vertex in this graph.)

Lemma 5.1. [20] (Step 1) *There is a way to choose disjoint sets G_i for each $i = 1 \dots \Delta$, such that $|G_i| = \lfloor \frac{|D_i|}{\beta} \rfloor$ and $G_i \subseteq D_i$.*

Lemma 5.2. *Step 2 can be done in $\lceil \log |D_i| \rceil + 1$ rounds.*

Lemma 5.3. *We can construct a transfer graph as described in Step 3 with in-degree at most 1 and out-degree at most $\beta - 1$.*

Lemma 5.4. *The in-degree of any disk in the transfer graph is at most β . The out-degree of any disk in the transfer graph is at most $2\beta - 2$. Moreover, the multiplicity of the graph is at most 4.*

Theorem 5.5. *The total number of rounds required for the multi-source multicast is $\max_i \lceil \log |D_i| \rceil + 3\beta + 3$.*

As the lower bound on the optimal number of rounds is $\max(\max_i \lceil \log |D_i| \rceil, \beta)$, we have a 4-approximation algorithm.

5.1 Allowing Bypass Nodes

The main idea is that without bypass nodes, only a small fraction of N disks is included in G_i for some i , if one disk requests many items while, on average, each disk requests few items. If we allow bypass nodes and hence G_i is not necessary a subset of D_i , we can make G_i very big so that each of almost all N disks belongs to some G_i . Bigger G_i reduces the out-degree of the transfer graphs and thus reduces the total number of rounds.

Algorithm Multi-source Multicast Allowing Bypass Nodes

1. We define $\bar{\beta}$ as $\frac{1}{N} \sum_{i=1 \dots N} |\{j | i \in D_j\}|$. In other words, $\bar{\beta}$ is the number of items a disk could receive, averaging over all disks. We arbitrarily choose a disjoint collection of subsets G_i , $i = 1 \dots \Delta$ with a constraint that $|G_i| = \lfloor \frac{|D_i|}{\bar{\beta}} \rfloor$. By allowing bypass nodes, G_i is not necessary a subset of D_i .
2. This is the same as Step 2 in the Multi-Source Multicast Algorithm, except that the source for item i (namely disk i) may belong to G_i for some i .
3. This step is similar to Step 3 in the Multi-Source Multicast Algorithm. We add $\lceil \bar{\beta} \rceil$ edges from each disk in G_i to satisfy $\lceil \bar{\beta} \rceil \cdot \lfloor \frac{|D_i|}{\bar{\beta}} \rfloor$ disks in D_i , and add at most another $\lceil \bar{\beta} \rceil - 1$ edges from disk i to satisfy the remaining disks in D_i .
4. This is the same as Step 4 in the Multi-Source Multicast Algorithm.

Theorem 5.6. *The total number of rounds required for the multi-source multicast algorithm, by allowing bypass nodes, is $\max_i \lceil \log |D_i| \rceil + \beta + \lceil 2\bar{\beta} \rceil + 6$.*

We now argue that $\lceil 2\bar{\beta} \rceil$ is a lower bound on the optimal number of rounds. Intuitively, on average, every disk has to spend $\bar{\beta}$ rounds to send data, and another $\bar{\beta}$ rounds to receive data. As a result, the total number of rounds cannot be smaller than $\lceil 2\bar{\beta} \rceil$. Allowing bypass node does not change the fact that $\max(\max_i \lceil \log |D_i| \rceil, \beta)$ is the other lower bound. Therefore, we have a 3-approximation algorithm.

References

1. E. Anderson, J. Hall, J. Hartline, M. Hobbes, A. Karlin, J. Saia, R. Swaminathan and J. Wilkes. An Experimental Study of Data Migration Algorithms. *Workshop on Algorithm Engineering*, 2001
2. B. Baker and R. Shostak. Gossips and Telephones. *Discrete Mathematics*, 2:191–193, 1972.
3. J. Bermond, L. Gargano and S. Perennes. Optimal Sequential Gossiping by Short Messages. *DAMATH: Discrete Applied Mathematics and Combinatorial Operations Research and Computer Science*, Vol 86, 1998.
4. J. Bermond, L. Gargano, A. A. Rescigno and U. Vaccaro. Fast gossiping by short messages. *International Colloquium on Automata, Languages and Programming*, 1995.
5. S. Berson, S. Ghandeharizadeh, R. R. Muntz, and X. Ju. Staggered Striping in Multimedia Information Systems. *SIGMOD*, 1994.
6. J. A. Bondy and U. S. R. Murty. Graph Theory with applications. *American Elsevier*, New York, 1977.
7. R. T. Bumby. A Problem with Telephones. *SIAM Journal on Algebraic and Discrete Methods*, 2(1):13–18, March 1981.
8. E. J. Cockayne, A. G. Thomason. Optimal Multi-message Broadcasting in Complete Graphs. *Utilitas Mathematica*, 18:181–199, 1980.
9. G. De Marco, L. Gargano and U. Vaccaro. Concurrent Multicast in Weighted Networks. *SWAT*, 193–204, 1998.

10. A. M. Farley. Broadcast Time in Communication Networks. *SIAM Journal on Applied Mathematics*, 39(2):385–390, 1980.
11. P. Fraigniaud and E. Lazard. Methods and problems of communication in usual networks. *Discrete Applied Mathematics*, 53:79–133, 1994.
12. L. Golubchik, S. Khanna, S. Khuller, R. Thurimella and A. Zhu. Approximation Algorithms for Data Placement on Parallel Disks. *Proc. of ACM-SIAM SODA*, 2000.
13. J. Hall, J. Hartline, A. Karlin, J. Saia and J. Wilkes. On Algorithms for Efficient Data Migration. *Proc. of ACM-SIAM SODA*, 620–629, 2001.
14. A. Hajnal, E. C. Milner and E. Szemerédi. A Cure for the Telephone Disease. *Canadian Mathematical Bulletin*, 15(3):447–450, 1972.
15. S. M. Hedetniemi, S. T. Hedetniemi and A. Liestman. A Survey of Gossiping and Broadcasting in Communication Networks. *Networks*, 18:129–134, 1988.
16. I. Holyer. The NP-Completeness of Edge-Coloring. *SIAM J. on Computing*, 10(4):718–720, 1981.
17. J. Hromkovic and R. Klasing and B. Monien and R. Peine. Dissemination of Information in Interconnection Networks (Broadcasting and Gossiping). *Combinatorial Network Theory*, pp. 125–212, D.-Z. Du and D.F. Hsu (Eds.), Kluwer Academic Publishers, Netherlands, 1996.
18. C. A. J. Hurkens. Spreading Gossip Efficiently. *Nieuw Archief voor Wiskunde*, 5(1):208–210, 2000.
19. S. Kashyap and S. Khuller. Algorithms for Non-Uniform Size Data Placement on Parallel Disks. *Manuscript*, 2003.
20. S. Khuller, Y. A. Kim and Y. C. Wan. Algorithms for Data Migration with Cloning. ACM Symp. on Principles of Database Systems (2003).
21. W. Knodel. New gossips and telephones. *Discrete Mathematics*, 13:95, 1975.
22. H. M. Lee and G. J. Chang. Set to Set Broadcasting in Communication Networks. *Discrete Applied Mathematics*, 40:411–421, 1992.
23. D. Liben-Nowell. Gossip is Synteny: Incomplete Gossip and an Exact Algorithm for Syntenic Distance. *Proc. of ACM-SIAM SODA*, 177–185, 2001.
24. C. H. Papadimitriou. Computational complexity. *Addison-Wesley*, 1994.
25. D. Richards and A. L. Liestman. Generalizations of Broadcasting and Gossiping. *Networks*, 18:125–138, 1988.
26. H. Shachnai and T. Tamir. On two class-constrained versions of the multiple knapsack problem. *Algorithmica*, 29:442–467, 2001.
27. H. Shachnai and T. Tamir. Polynomial time approximation schemes for class-constrained packing problems. *Proc. of Workshop on Approximation Algorithms*, 2000.
28. C.E. Shannon. A theorem on colouring lines of a network. *J. Math. Phys.*, 28:148–151, 1949.
29. M. Stonebraker. A Case for Shared Nothing. *Database Engineering*, 9(1), 1986.
30. R. Tijdeman. On a Telephone Problem. *Nieuw Archief voor Wiskunde*, 19(3):188–192, 1971.
31. V. G. Vizing. On an estimate of the chromatic class of a p-graph (Russian). *Diskret. Analiz.* 3:25–30, 1964.