

# Notes On the Peer-to-Peer Data Dissemination Problem

Jian Li

Department of Computer Science, University of Maryland, College Park.

## Abstract

We consider the problem to disseminate a file to all peers on a peer-to-peer network. We prove the file dissemination problem is NP-hard for arbitrary networks (modeled as undirected graphs) while polynomial time solvable if the network topology is a tree. We also provide an optimal algorithm for disseminate a file on Chord-like graphs. One salient feature of this algorithm is that it fully makes use of Chords routing table and can be complete decentralized while the optimality is preserved.

**Keywords:** Peer-to-Peer, Data Dissemination

## 1 Introduction

Recently, quite a few file dissemination systems in peer-to-peer networks are proposed to utilize peers' upload bandwidth to achieve a better system performance, such as BitTorrent[5], Bullet[6], FOX[7], etc. We assume the information source holds a file which is equally divided into  $M$  pieces and need to distribute it to all  $N$  nodes in the network.

### 1.1 Related Work

Considerable work on designing and analyzing file dissemination systems has been done recently. We just discuss a few closely related ones. The coupon replication system [9] analyzes the asymptotic behavior of the piece distribution assuming uniform piece exchanging rate and a randomly uploading strategy. Arthur et al. models the system by an undirected graph such that data transmission can only happen on edges [8]. They show the file dissemination can be finished in  $O(M \log N)$  rounds in the BitTorrent-like graph. Our work uses the same model. In the FOX system [7], all peers have same upload bandwidth and each peer can upload to at most  $k$  peers at the same time. Peers are arranged to form a  $k$ -ary tree and an asymptotically completion time  $O(\log_k N + \frac{M}{B})$  can be achieved. Mundinger et al. propose a centralized optimal scheduling algorithm for file dissemination problem[4] on complete graphs. All the above model assume that

each piece can be transmitted in equal time for each peer. Therefore, they can use the number of “rounds” to measure the makespan (time for the whole system to complete the file distribution) Most of these work assume a limited upload bandwidth but an infinite download bandwidth. We note here this assumption conforms to some extent to the real scenario that the download bandwidth is often several times larger than the upload bandwidth.

## 1.2 Our Results

Our results are summarized as follows:

- We prove the file dissemination problem is NP-hard in general graph even there is only one piece of file.
- We solve the file dissemination problem on trees optimally by dynamic programming.
- We generalize the result in [4] by proposing an decentralized and easy-to-implemented algorithm for Chord-like graph(defined later).

## 2 Preliminary

### 2.1 Model

We use the same model as Arthur et al. do[8]. We assume the communication network is a undirected graph. The file required to be disseminated through the network is divided into a number of equal-sized parts, which we call pieces. We assume each node has the same upload bandwidth and unlimited download bandwidth. It has been shown that an optimal centralized solution can always been achieved by only uploading one piece to only one peer at a time[4]. Therefore we can assume the time can be discretized into rounds. Uploading one piece can be done in one round. We require that at each round, one peer can only upload one piece it already has to one of its neighbors.

It is nature to consider the following two objectives:

- minimize the maximum completion time.
- minimize the average completion time.

In this note, we mainly focus on the objective (1).

### 2.2 Binomial Trees

A binomial tree is a tree with a very special topology: see Figure 2.2

**Definition 1** [1] (Binomial Tree of order  $k$ ) *The binomial tree of order  $k$  with root  $r$  is the tree  $T_k$  defined as follows:*

- If  $k = 0$ ,  $T_k = \{r\}$  .
- If  $k > 0$ ,  $T_k$  comprises the root  $r$  and  $k$  binomial subtrees,  $T_0, T_2, \dots, T_{k-1}$  with each of their root connected to  $r$ .

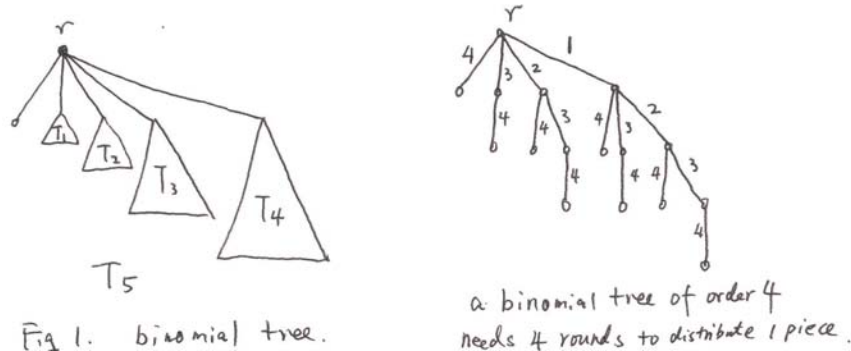


Figure 1: Binomial Trees.

## 2.3 Chord-like Graph

**Definition 2** (Chord-like graph) *A Chord-like graph is a graph where the edges are defined by the routing table in the Chord protocol[10]. Namely, if a node's routing table contains another node's address, there is an edge between them. A Chord-like graph is perfect if all nodes are exactly evenly distributed over the ID space.*

I recently found a perfect Chord-like graph of size a power of 2 is actually a hypercube.

## 3 NP-Completeness

It is easy to see  $\lceil \log n \rceil$  is a lower bound of rounds to distribute one piece of file in a graph of size  $n$ . It is also not hard to see we can do it within  $i$  rounds in a graph with  $2^i$  vertices if and only if the dissemination tree is a binomial tree of order  $i$ , see the right hand side of Figure 2.2. So if we can show the NP-Completeness of deciding whether a given graph of size  $2^i$  contains a binomial spanning tree rooted at a specific vertex (we call it rooted binomial spanning tree (RBST) problem), the NP-hardness of the data dissemination problem follows.

Before we show the hardness of RBST problem, we prove a closely related problem, the binomial spanning tree (BST) problem, is NP-Complete.

**Lemma 3** *It is NP-complete to decide whether a given graph  $G(V, E)$  with  $2^i$  nodes contains a binomial spanning tree.*

**Proof:** Our proof bears a resemblance to the the NP-Completeness proof of the bisectable-tree problem[3]. To show the problem is NP-Complete, we reduce from the known NP-Complete Problem  $H$ -matching.  $H$ -matching problem asks whether all vertices of a given graph can be covered by vertex disjoint copies of a given graph  $H$ . If  $H$  is any connected graph with at least 3 vertices, the problem is NP-Complete [2]. In our proof,  $H$  is a path of length 4 (also a binomial tree of order 2).

Given an  $H$ -matching instance, say  $G(V, E)$ , we build a new larger graph  $G'(V', E')$ . W.l.o.g, we assume  $|V|$  is a power of 2 since we can add a sufficient number of paths

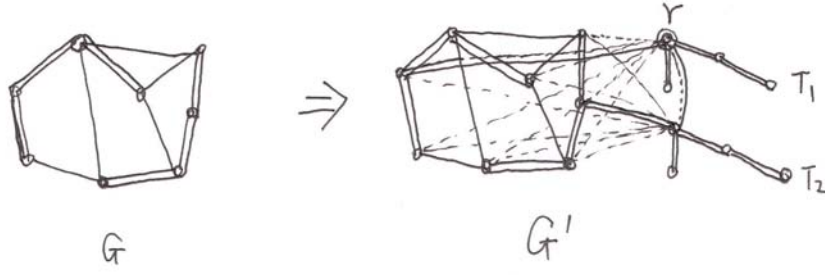


Figure 2: The construction in the proof of Theorem 3.

of length 4 such that the number of vertices is a power of 2 and the new graph has a  $H$ -matching if and only if  $G$  does.  $G'$  is constructed as follows. Add  $\frac{|V|}{4}$  binomial trees of order 2  $T_i$  to  $G$  with all their roots interconnect to each other. Suppose  $s_i$  is  $T_i$ 's root for  $1 \leq i \leq \frac{|V|}{4}$ . Add all edges of form  $(s_i, v)$  for  $1 \leq i \leq \frac{|V|}{4}$  and  $v \in V$ . An example of the construction is depicted in Figure 3.

Now, we briefly argue  $G$  has a  $H$ -matching if and only if  $G'$  has a binomial spanning tree. Suppose  $G'$  has a binomial spanning tree  $T$ . It is easy to see any binomial tree has a unique  $H$ -matching. So each  $T_i$  in  $G'$  corresponds to exactly one copy of  $H$  in the  $H$ -matching of  $T$ . Therefore, each other copies of  $H$  is fully contained in  $G$  which implies  $G$  has an  $H$ -matching. Now, suppose  $G$  has a  $H$ -matching. Since we have all edges between  $G'$  and  $s_i$ s, we can link each copy of  $H$  in  $G$  to one  $T_i$  to form a binomial tree of order 3 rooted at  $s_i$ . Because all  $s_i$ s are fully connected, it is always possible to further link these binomial trees to a larger binomial spanning tree.  $\square$

**Lemma 4** *Given a graph  $G(V, E)$  with a specific node  $s \in V$  and  $|V| = 2^i$ , it is NP-complete to determine  $G$  has a binomial spanning tree rooted at  $s$ , i.e. RBST problem is NP-complete.*

**Proof:** We prove the theorem by reducing it to the BST problem. Given an instance of a BST problem, say  $G(V, E)$  with  $|V| = 2^i$ , we construct a new graph  $G'(V', E')$ .  $G'$  is formed by the union of  $G$  and a binomial tree of order  $i$  rooted at  $s$ . We add edges  $(s, v)$  for all  $v \in V$ . It is easy to see  $G$  has a binomial spanning tree of order  $i$  if and only if  $G'$  has a binomial spanning tree of order  $i + 1$  and rooted at  $s$ . Therefore, the NP-Completeness follows from Theorem 3.  $\square$

Now, our main theorem follows.

**Theorem 5** *The file dissemination problem is NP-hard even if there is only one piece of file.*

## 4 Algorithm for Trees

In this section, we show a simply dynamic program which optimally solves the data dissemination problems on trees.

We first assume the file only contains one data pieces. We denote the communication tree by  $T$  and its root  $r$ . We use  $T_v$  to denote the subtree rooted at vertex  $v$ . We define  $OPT(T_v)$  be the optimal number of rounds to distribute the file to all nodes in  $T_v$  assuming the file is only contained in  $v$  initially. Let  $v_1, v_2, \dots, v_l$  be the children of  $v$ , already sorted in an nonincreasing order of  $OPT(T_{v_i})$ .

The dynamic program is simply

$$OPT(v) = \begin{cases} 0, & \text{if } v \text{ is a leaf;} \\ \max_{1 \leq i \leq l} \{OPT(v_i) + i\}, & \text{Otherwise.} \end{cases} \quad (1)$$

$OPT$  values are computed bottom-up in the tree. The idea is a child with higher  $OPT$  value has to get the piece earlier to minimize the system makespan. The proof of the optimality is straightforward and omitted here.

To generalize the dynamic program to multiply pieces cases is an ongoing work. We note here trivially apply the above algorithm for each piece one by one is not an optimal solution anymore.

## 5 Algorithm for Chord-like Graph

For simplicity of discussion, we first assume there are  $N = 2^n$  nodes in the system and one of them has all  $M$  pieces of the file. It has been show that the minimum number of rounds is  $n - 1 + M$  if the network is a complete graph[4]. In this section, we generalized this result to show that the same result holds for Chord-like Graph. Our algorithm is much easier to state and implemented than the one in [4]. One salient feature of our algorithm is that it can be fully decentralized and no bitmap exchanging is needed.

We number the node in the Chord ring from 0 to  $2^n - 1$  and the file pieces from 1 to  $M$  and assume node 0 is the file source. Recall the routing table stored in a Chord node  $v$  contains all neighbors whose (clockwise) ring distance to  $v$  is a power of 2. Formally, node  $i$ 's neighbor set is  $\{(i+1) \bmod N, (i+2) \bmod N, \dots, (i+2^{n-1}) \bmod N\}$ . We call  $(i+2^j) \bmod N$  the  $n-1-j$ th neighbor of  $i$ .

Basically, each node uploads to its neighbors in a round-rabin manner. See Algorithm 1 for the pseudocode of the algorithm.

---

### Algorithm 1 CHORD-DISS

---

```

for r=1 to n-1+M do
  if r ≤ M then
    Peers 0 upload piece r to its (r mod n)th neighbor;
  end if
  For all i ≠ 0, peer i uploads the piece with highest piece number to its (r mod n)th neighbor;
end for

```

---

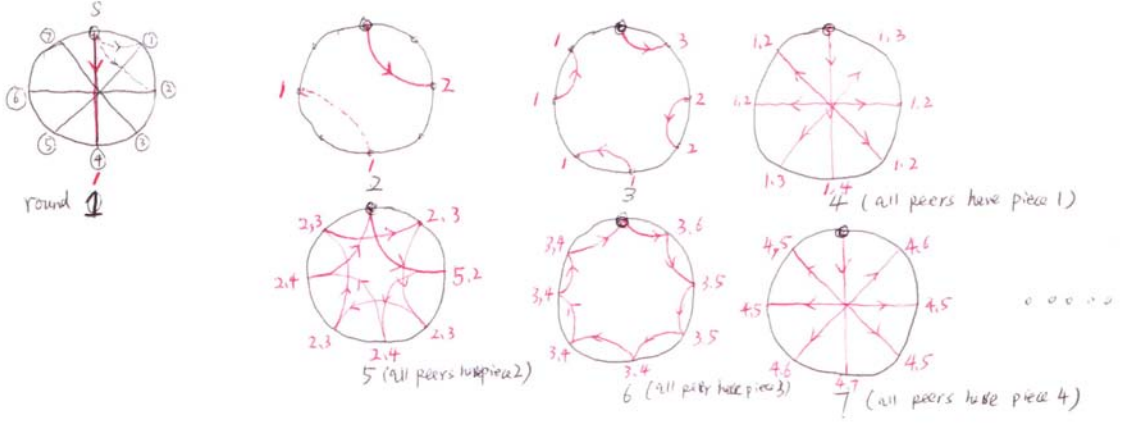


Figure 3: A running example of CHORD-DISS.

See Figure 5 for a small but instructive running example of Algorithm 1. Now, we formally prove the correctness of the algorithm.

**Theorem 6** *The file dissemination can be completed after round  $n - 1 + M$  by Algorithm CHORD-DISS.*

**Proof:** We prove a slightly generalization of the theorem by induction. For ease of description, we assume there are infinite number of pieces number from  $\{-\infty, \dots, -1, 0, 1, \dots, \infty\}$  and before round 1, we have

1. All peers have piece  $\{-\infty, \dots, -n + 1\}$ .
2. All peers whose binary representations of their number labels sharing the same digits  $1, 0, 0, \dots, 0$  on bits  $(j \bmod n), (j + 1 \bmod n), (j + 2 \bmod n), \dots, (1 \bmod n)$ , respectively, has the piece  $j$  for  $-n + 1 \leq j \leq 0$ .

This is without loss of generality since if all peers have pieces from 1 to  $M$  after round  $n - 1 + M$  in this case, they will do the same in the case where there are only  $M$  pieces. We inductively prove the following claim: We assume at the time right before round  $i$ ,

1. All peers have piece  $\{-\infty, \dots, i - n - 1\}$ .
2. All peers whose binary representations of their number labels sharing the same digits  $1, 0, 0, \dots, 0$  on bits  $(j \bmod n), (j + 1 \bmod n), (j + 2 \bmod n), \dots, (i \bmod n)$ , respectively, has the piece  $j$ .

Now, we prove the claim holds before round  $i + 1$ .

1. Consider the peers which contain piece  $i - n$  before round  $i$ . By induction hypothesis 2 (let  $k=0$ ), those are peers whose binary representations share the same digits 1 on bits  $i \bmod n$ . At round  $i$ , each of those peer upload this piece to its  $i$ th neighbor which is the one with the same binary representation except on bit  $i$ . After the execution of this round, it is clear each peer has piece  $i - n$ .

2. Similarly, all peers which contain piece  $j$  have binary representation with digits  $0, 0, 0, \dots, 1$  on bits  $(i \bmod n), (i+1 \bmod n), (i+2 \bmod n), \dots, (j \bmod n)$  before round  $i$ . Then after the execution of round  $i$ , the set of peers becomes the ones who have binary representation with digits  $0, 0, 0, \dots, 1$  on bits  $(i+1 \bmod n), (i+2 \bmod n), \dots, ((i+1) + k - n \bmod n)$ .

The theorem trivially follows from claim 1. □

To generalize the algorithm to graph of size not a power of 2 is an ongoing work.

## 6 Algorithm under the same assumption as FOX's

Ongoing work...

## 7 Concluding Remark

This note serves for the course project of CMSC711 (Network course). I recently found exactly the same problem had already been extensively studied in theory community under the name "broadcast problem". I am pretty surprised that most recent work try to design scheduling algorithms for P2P network, at least those I listed in reference, even didn't mention these work. I obtained these results independently. I haven't really read these papers and hopefully my approaches or descriptions are different.

## References

- [1] T. H. Cormen, C. E. Leiserson, and R. L. Rivest. Introduction to Algorithms. *MIT press*, pages 498–513, 1990.
- [2] M. R. Garey and D. S. Johnson. Computers and Intractability: a Guide to the Theory of NP-Completeness. W. H. Freeman, 1979.
- [3] David Eppstein. Squarepants in a Tree: Sum of Subtree Clustering and Hyperbolic Pants Decomposition. In *Proc of SODA '07*.
- [4] Jochen Munding, Richard Weber, Gideon Weiss. Optimal scheduling of peer-to-peer file dissemination. In *Journal of Scheduling*, Vol 11, Issue 2, 2008.
- [5] D. Qiu, R. Srikant. Modeling and performance analysis of BitTorrent-like peer-to-peer networks. In *Proc. of SIGCOMM*, 2004
- [6] Dejan Kostić, Alex C. Snoeren, Amin Vahdat, Ryan Braud, Charles Killian, James W. Anderson, Jeannie Albrecht, Adolfo Rodriguez, Erik Vandekieft. High-bandwidth data dissemination for large-scale distributed systems In *ACM Transactions on Computer Systems (TOCS)*, Vol 26, Issue 1, 2008
- [7] Dave Levin, Rob Sherwood, Bobby Bhattacharjee. Fair File Swarming with FOX. In *International Workshop on Peer-to-Peer Systems*, 2006.

- [8] David Arthur and Rina Panigrahy. Analyzing BitTorrent and related peer-to-peer networks. In *Proc. of SODA*, 05.
- [9] L.Massoulié, M.Vojnović. Coupon Replication Systems. In *Proc of SIGMETRICS*, 2005
- [10] I.Stoica, R. Morris, D.Liben-Nowell, D.R.Karger, M.F.Kaashoek, F.Dabek, H.Balakrishnan. Chord: a scalable peer-to-peer lookup protocol for Internet applications. In *Proc of SIGCOMM*, 2001