Minimizing Weighted Sum Finish Time for One-to-Many File Transfer in Peer-to-Peer Networks

Bike Xie, Mihaela van der Schaar, Thomas Courtade and Richard D. Wesel
Department of Electrical Engineering, University of California, Los Angeles, CA 90095-1594
Email: bike@marvell.com, mihaela@ee.ucla.edu, tacourta@ucla.edu, wesel@ee.ucla.edu

Abstract— **This paper considers the problem of transferring a file from one source node to multiple receivers in a peer-to-peer (P2P) network. The objective is to minimize the weighted sum finish time (WSFT) for the one-to-many file transfer where peers have both uplink and downlink bandwidth constraints specified. The static scenario is a file-transfer scheme in which the constructed network topology and the network resource (link throughput) allocation remains static until all receivers finish downloading. This paper first shows that the static scenario can be optimized in polynomial time by convex optimization, and the associated optimal static WSFT can be achieved by linear network coding. This paper also proposes a static rateless-coding-based scheme which has almost-optimal empirical performance. The dynamic scenario is a file-transfer scheme which can re-construct the network topology and re-allocate the network resource during the file transfer. This paper proposes a dynamic rateless-coding-based scheme, which provides significantly smaller WSFT than the optimal static scheme does.**

Index Terms— **P2P network, network coding, rateless code, static scenario, dynamic scenario.**

I. Introduction

P2P applications (e.g, [1], [2], [3], [4]) are increasingly popular and represent the majority of the traffic currently transmitted over the Internet. A unique feature of P2P networks is their flexible and distributed nature, where each peer can act as both a server and a client [5]. Hence, P2P networks provide a cost-effective and easily deployable framework for disseminating large files without relying on a centralized infrastructure [6]. These features of P2P networks have made them popular for a variety of broadcasting and file-distribution applications [6] [7] [8] [9] [10] [11] [12]. In a P2P file distribution application, the key performance metric from an end-user's point of view is the finish time, or the time it takes for an end-user to download a file.

This paper considers the problem of transferring a file from one source to multiple receivers in a peer-to-peer (P2P) network. Specifically we are concerned with selecting the network connections that will minimize the finish time. While we use network coding in our algorithms, the main focus of this paper is distinct from network coding papers such as [13] [14] [15] [16] [17] [18] [19] [20] [21], which assume that the network topology has been established a-priori.

In [13], Ahlswede et. al. introduced the concept of network coding and demonstrated that the network coding outperforms network routing for multicast scenarios in a given network topology. The constructions and the capacity of network coding, especially linear network coding, over a pre-determined network topology are well studied in [14] [15] [17] [18] and [19]. In [21], the authors provided the capacity of network coding for multicast scenarios over a randomized connected network topology. All these papers assume that the network topology has been established a-priori, in a deterministic or a randomized manner.

This paper addresses the fundamental performance limit of the P2P file distribution applications and hence focuses on the centralized algorithms with full knowledge of the P2P network. While some other papers in the P2P area such as [22] [23] [24] [25] [26] and [27] study and optimize the performance of the distributed P2P protocols. In [27], the authors applied network coding in a gossip-based protocol and provided performance analysis over a homogeneous P2P network where each node sends one packet to a randomly selected node in each time slot.

The network-coding papers described above assume that the network topology has been established a-priori. In contrast, the primary concern of this paper is establishing the optimal network topology. Other papers such as [28] [29] [30] [31] [32] [33] [34] and [35] have addressed the problem of establishing an optimal network topology or an optimal scheduling policy for a given topology, but these papers do not take advantage of the tremendous benefit provided by network coding.

Furthermore, these papers usually consider or optimize the network topology under certain node-to-node communication models such as the unidirectional telephone model [30] [31], the bidirectional telephone model [32], the simultaneous send/receive model [33], or the uplink-

Bike Xie is currently with Marvell Semiconductor Inc. This work was done when Bike Xie was with University of California, Los Angeles.

sharing model [34] [35]. Our paper optimizes the network topology without such strong constraints on none-to-node communication.

In [28], Sanghavi et. al. considered the problem of distributing M messages from multiple nodes to all nodes over a homogeneous P2P network, and proposed scheduling protocols to approach the minimum last finish time. In this work, the network topology is randomly constructed rather than optimized such that in each time slot, users contact each other in a random uncoordinated manner and users upload one piece of file per time slot.

In [34] and [35], Mundinger et. al. investigated the problem of distributing M messages from one source node to all other nodes in a heterogeneous P2P network assuming node upload capacities are the only bottleneck in the network. They provided the centralized optimal network topology (by solving a mixed integer linear program) and the corresponding scheduling policy to minimize the last finish time. In these works, the uplink-sharing model is pre-assumed such that at each time, each node can only upload one packet to its directly-connected receivers and the upload capability has to be equally distributed to all of its directly connected receivers.

In order to explore the fundamental performance limits of one-to-many file distribution, some papers relax the constraints on the node-to-node communication model and allow to partition a file into as many pieces as possible. In [9], Li, Chou, and Zhang explored the problem of delivering the file, which is infinitely divisible, from one source node to all receivers in a heterogeneous P2P network constrained only by node upload capacities. No node-to-node communication model is pre-assumed, i.e., the nodes can transmit any number of pieces to any number of other nodes at each time as long as the constraints on the node upload capacity are satisfied. They introduce a routing-based scheme, referred to as Mutualcast, which minimizes the last finish time to all receivers with or without helpers.

In [34] and [35], Mundinger et. al. studied a more general P2P problem in which a distinct file from each node needs to be distributed to all other nodes in the heterogeneous P2P network constrained only by node upload capacities. The authors provided a routing-based scheme to achieve minimum last finish time. In [36], the authors investigated the multi-source multicast P2P scenarios only with node uplink constraints, and showed that even if network coding is applicable, routing is optimal to minimize last finish time when all multicast tasks have pairwise identical or disjoint receivers.

Some papers have addressed topology optimization in the context of network coding. In [37], Wu et. al. investigated the finish time region for P2P file transfer. Given an order at which the receivers finish downloading, they demonstrated in [37] that the minimum weighted sum finish time (WSFT) can be solved in polynomial time by convex optimization, and can be achieved by linear network coding, assuming that node uplinks are the only bottleneck in the network. They also proposed a routing-based scheme which empirically almost minimizes the average finish time over homogeneous P2P networks, and demonstrated how to significantly reduce the average finish time at the expense of a slight increase in the last finish time. However, the proposed scheme desires very large download capacities for certain nodes, especially the last finished node, and hence, might not work properly if some nodes have limited download capacities.

In [38], the authors investigated the one-to-all file distribution in heterogeneous networks and considered 3 different criteria: last finish time, average finish time, and min-min times, where"min-min times" refers to sequentially minimizing the finish times according to a specified order. The optimal network topology to achieve the minimum average finish time over a P2P network with 3 peers is provided based on a brute-force search.

In [39] and [40], the authors also considered the one-to-all file distribution in heterogeneous network assuming that node upload capacities are the only bottleneck in the network. These works proposed an optimal network topology and the associated scheduling policy to achieve the min-min times. Similar to the scheme in [37], this optimal solution also desires very large download capacities for certain nodes, and hence, it might not work properly if some nodes have limited download capacities. The authors also mentioned that the behavior of the optimal scheme needs to be investigated when nodes dynamically enter and leave upon completion. The authors claimed that the proposed scheme which achieves min-min times can also achieve the minimum average finish time. However, Chang et. al. [41] showed that this is not necessarily true when the number of peers is larger than 4.

Our paper considers the problem of minimizing weighted sum finish time (WSFT) from one source node to many receivers in a heterogeneous P2P network. First, this paper considers the criteria of WSFT, which is an generalization of the average finish time which is considered in most related works. Second, this paper considers the one-to-many file distribution which is an extension to the one-to-all file distribution investigated in some related research. (For peers which don't require the file, we can simply set their weights to be zero.) Last but not least, it is assumed that both upload and download capacities can be bottlenecks in the network and that every node can connect to every other node through routing in the overlay. Most research in P2P consider node uplinks as the only bottleneck because the uplink capacity is often several times smaller than the downlink capacity for typical residential connections (e.g., DSL and Cable). However, the downlink capacity can still be exceeded when a peer

downloads from many other peers simultaneously, as in the schemes proposed in [37] and [39]. For this reason, our paper also takes the download capacity constraints into account.

This paper is organized as follows: Section II introduces definitions and notations for P2P networks. Section III studies the scenario in which network resource allocations remain static. Section III-A provides the optimal resource allocation for the static scenario by solving a convex optimization problem to minimize the WSFT. Section III-B provides a lower bound to the minimum WSFT for static scenarios. Section III-C proposes a static rateless-coding-based scheme, and Section III-D provides simulations showing that the rateless-coding-based scheme closely approaches the lower bound for a variety of configurations. Section IV investigates the dynamic scenario in which network resource allocations need not remain static. Section IV-A proposes a dynamic rateless-coding-based scheme and Section IV-B provides simulation results showing how the dynamic solution can provide a reduction in WSFT as compared to the static solution. Section V delivers the conclusions.

II. Network Setup and Problem Definition

This paper focuses on content distribution applications (e.g, BitTorrent [1]) in which peers are only interested in content at full fidelity. The key issue for these P2P applications is to minimize download times (delays) to receivers. In order to understand the fundamental performance limit for one-to-many file transfer in P2P networks, it is assumed that all nodes cooperate, and a centralized algorithm provides the file-transfer scenario with the full knowledge of the P2P network including the source node's uplink capacity, and the weights, downlink capacities, and uplink capacities of peers.

This paper starts with static P2P networks in which the set of peers does not change over time. In a static P2P network, a source node s with uplink bandwidth U_s has a file of size B. There are N peers, denoted as $\{1, \cdots, N\}$, who want to download the file possessed by the source node. Each peer has weight W_i, downlink capacity D_i and uplink capacity U_i, for $i = 1, 2, \cdots, N$. It is reasonable to assume that $D_i \geq U_i$ for each $i = 1, \cdots, N$ since it holds for typical residential connections (e.g., Fiber, DSL and Cable).

Denote the allocated link throughput (transmission rate) from the source node to peer j as $r_{s\to j}$ and the link throughput (transmission rate) from peer i to peer j as $r_{i\to j}$. As a notational convenience, we also denote $r_{j\to j}$ as the link throughput from the source node to peer j. Since the total download rate is constrained by the downlink capacity, we have $\sum_{i=1}^{N} r_{i\to j} \leq D_j$ for all $j = 1, \cdots, N$. The total upload rate is constrained by the uplink capacity. Hence, $\sum_{i\neq j} r_{j\to i} \leq U_j$ for all $j = 1, \cdots, N$. One example of the peer model is shown in Fig. 1. The downlink capacity and uplink capacity of peer 1 are D_1 and U_1 respectively. Thus, the total download rate $r_{s\to 1} + \sum_{i=2}^{4} r_{i\to 1} = \sum_{i=1}^{4} r_{i\to 1}$ has to be less than or equal to D_1, and the total upload rate $\sum_{i=2}^{4} r_{1\to i}$ has to be less than or equal to U_1.

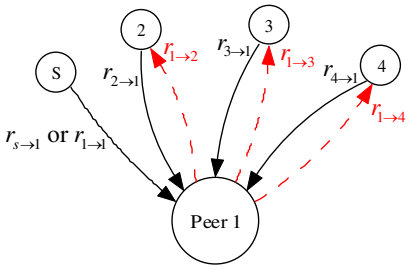

Fig. 1. The peer model

III. Static File-Transfer Scenarios

A static file-transfer scenario is a file-transfer scheme in which the network resource allocation remains static until all receivers finish downloading. The network resource allocation of a static scenario is determined by $r_{s\to j}$ (or $r_{j\to j}$) for $j = 1, \cdots, N$ and $r_{i\to j}$ for $j \neq i$.

A. Optimal Static Scenario

Let t_j denote the finish time for peer j for $j = 1, \cdots, N$. Given a static scenario $r_{i\to j}$, $(i, j = 1, \cdots, N)$, the maximum flow rate to peer j, denoted as r_j, is limited by the minimum cut from the source node s to peer j in the network by the Max-Flow-Min-Cut Theorem, and hence, $t_j \geq \frac{B}{r_j}, \forall j = 1, \cdots, N$. In fact, $t_j = \frac{B}{r_j}$ can be achieved simultaneously for all $j = 1, \cdots, N$.

Lemma 1: Given a static scenario $\{r_{i\to j}\}_{i,j=1}^{N}$ for a P2P network, the only Pareto optimal (smallest) finish time vector is $t_j = \frac{B}{r_j}$ for $j = 1, \cdots, N$, where r_j is the minimum cut from the source node s to peer j.

Proof: By Max-Flow-Min-Cut Theorem, the flow rate to peer j is less than or equal to r_j. By Network Coding Theorem [13] [14] and the construction of the time-expanded graph [37], the set of the flow rates $\{r_j\}_{i=1}^{N}$ is achievable. Hence, the only Pareto optimal finish time vector is $t_j = \frac{B}{r_j}, \forall j = 1, \cdots, N$. A detailed proof is deferred to our journal manuscript available on ArXiv [42]. ■

A set of flow rates $\{r_i\}_{i=1}^{N}$ is feasible if and only if there exists a solution to the following system of linear

inequalities:

$$\sum_{i=1}^{N} r_{i\to i} \le U_s; \quad \text{(recall that } r_{i\to i} \triangleq r_{s\to i}) \tag{1}$$

$$\sum_{j=1,j\neq i}^{N} r_{i\to j} \le U_i, \quad \forall i = 1,\cdots,N; \tag{2}$$

$$\sum_{j=1}^{N} r_{j\to i} \le D_i, \quad \forall i = 1,\cdots,N; \tag{3}$$

$$0 \le \boldsymbol{f}^{(i)} \le \boldsymbol{r}, \quad \forall i = 1,\cdots,N; \tag{4}$$

where vector $\boldsymbol{r}$ with elements $r_{i\to j}$ represents the allocated link throughput in the network, and $\boldsymbol{f}^{(i)}$ is a flow, constrained with the link throughput $\boldsymbol{r}$, from the source node s to peer i with flow rate r_i.

By Lemma 1, the minimum WSFT is the solution to the convex optimization of minimizing $\sum_{i=1}^{N} W_i B/r_i$ subject to (1-4). Thus, we can conclude the following theorem:

Theorem 1: Consider multicasting a file with size B from a source node s to peers $\{1,\cdots,N\}$ in a P2P network in which node uplinks and downlinks are the only bottlenecks. The minimum weighted sum finish time for static scenarios and the corresponding optimal static allocation can be found in polynomial time by solving the convex optimization of minimizing $\sum_{k=1}^{N} W_k B/r_i$ subject to constraints (1-4).

Theorem 1 can be extended by adding other linear network constraints (e.g. edge/link capacity constraints). For a special case where all peers have the same weight (normalized to 1) and infinite downlink capacities, the optimal static scenario achieves the finish times of $\frac{B}{\min(U_s,(U_s+\sum_{i=1}^{N} U_i)/N)}$ for all peers and obtains the minimum sum finish time of $\sum_{k=1}^{N} t_k = \frac{NB}{\min(U_s,(U_s+\sum_{i=1}^{N} U_i)/N)}$. In fact, this optimal static scenario is the same as the scenario of Mutualcast [9], which minimizes the last finish time to all peers. Hence, in this special case, the allocation of Mutualcast is the optimal static allocation which not only achieves the minimum sum finish time but also minimizes the last finish time of peers.

B. Bounding the Weighted Sum Finish Time for Static Scenarios

Consider the cut of $\{s,1,\cdots,i-1,i+1,\cdots,N\} \to \{i\}$ for any static scenario $r_{i\to j}$ $(i,j=1,\cdots,N)$. The maximum flow rate from the source node s to peer i, r_i, is limited by

$$r_i \le \sum_{j=1}^{N} r_{j\to i} \le D_i, \tag{5}$$

and

$$\sum_{i=1}^{N} r_i \le \sum_{i=1}^{N}\sum_{j=1}^{N} r_{j\to i} \le \sum_{j=1}^{N} r_{j\to j} + \sum_{j=1}^{N}\sum_{i=1,i\neq j}^{N} r_{j\to i} \tag{6}$$

$$\le U_s + \sum_{j=1}^{N} U_j. \tag{7}$$

Consider the cut of $\{s\} \to \{1,\cdots,N\}$. r_i is also bounded by

$$r_i \le \sum_{j=1}^{N} r_{j\to j} \le U_s. \tag{8}$$

Because all feasible sets of $\{r_i\}_{i=1}^{N}$ satisfy (5), (7) and (8), the solution to the optimization problem of minimizing $\sum_{i=1}^{N} W_i \frac{B}{r_i}$ subject to (5) (7) and (8) provides a lower bound to the minimum WSFT for static scenarios. The optimal solution to this relaxed problem is

$$r_i^* = \begin{cases} \sqrt{W_i}\cdot R, & \text{if } \sqrt{W_i}\cdot R < \tilde{D}_i, \\ \tilde{D}_i & \text{if } \sqrt{W_i}\cdot R \ge \tilde{D}_i, \end{cases} \tag{9}$$

where $\tilde{D}_i \triangleq \min(U_s, D_i)$ and R is chosen such that $\sum_{i=1}^{N} r_i^* = \min(U_s + \sum_{i=1}^{N} U_i, \sum_{i=1}^{N} \tilde{D}_i)$.

For the special case where $W_i = 1$ and $D_i = \infty$, the solution (9) is $r_i^* = \min(U_s, (U_s + \sum_{i=1}^{N} U_i)/N)$ and the lower bound to the minimum WSFT is $\frac{NB}{\min(U_s,(U_s+\sum_{i=1}^{N} U_i)/N)}$. As discussed in Section III-A, the routing-based scheme, Mutualcast [9], can achieve the finish time of $\frac{B}{\min(U_s,(U_s+\sum_{i=1}^{N} U_i)/N)}$ for all peers. Hence, the lower bound is attainable for this case.

Theorem 2: **(Minimum Sum Finish Time)** Consider multicasting a file with size B from a source node s to peers $\{1,\cdots,N\}$ in a P2P network in which peer uplink and downlink are the only bottlenecks. The lower bound to the minimum sum finish time, i.e. $\sum_{i=1}^{N} \frac{B}{r_i^*}$, is achievable, where r_i^* follows from (9) with $W_i = 1$.

Proof: The proof is deferred to our journal manuscript available on ArXiv [42]. ■

C. Rateless-Coding-Based Scheme

The rateless erasure code is rateless in the sense that the number of encoded packets that can be generated from the source message is potentially limitless [43]. Suppose the original file size is B packets. Once the receiver has received any B' packets, where B' is just slightly greater than B, the whole file can be recovered. The percentage of the overhead packets goes to zero as B goes to infinity. In practice, the overhead is about 5% for LT codes with file size $B \simeq 10000$ [43]. This sub-section focuses on applying rateless erasure codes for P2P file transfer instead of designing rateless erasure codes. Hence, we

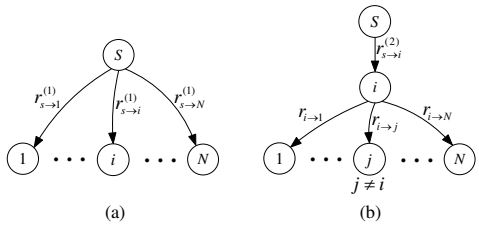

Fig. 2. (a) Depth-1 tree; (b) Depth-2 tree.

assume the overhead of the applied rateless erasure code is zero for simplicity.

The rateless-coding-based scheme constructs the two types of trees in Fig. 2 to distribute the content. The source node first partitions the whole file into B chunks and applies a rateless erasure code to these B chunks. For the depth-1 tree, the source node broadcasts independently rateless-coded chunks directly to peers. For the depth-2 trees, the source node sends independently rateless-coded chunks to a peer, and then the peer copies and forwards some of the rateless-coded chunks to other peers. This scheme requires that $0 \leq r_{i\to j} \leq r_{s\to i}$, and guarantees that all chunks received by a peer are independently generated. Hence, a peer can decode the whole file as long as it receives B coded chunks.

The rateless-coding-based scheme has a much simpler mechanism than that of routing-based schemes such as Mutualcast. First, the source node and peers don't need a chunk selection algorithm because all coded chunks transmitted from the source node are independently generated. For the same reason, peers don't need to feedback the index information of the received chunks to their neighbors. Second, the network resource allocation is more flexible than those for Mutualcast or other routing-based schemes because peers don't have to receive exactly the same chunks to decode the whole file. Third, this scheme is robust to the packet loss in the Internet since the rateless erasure codes are designed for erasure channels.

For the rateless-coding-based scheme, the optimal network resource allocation can be obtained by solving the following convex optimization problem.

$$\begin{array}{cc} \min & \sum_{i=1}^{N} W_i \frac{B}{r_i} \\ \text{subject to} & 0 \leq r_{i\to j} \leq r_{i\to i}, \forall i,j = 1,\cdots,N, \\ & \sum_{i=1}^{N} r_{i\to i} \leq U_s, \\ & \sum_{j=1,j\neq i}^{N} r_{i\to j} \leq U_i, \forall i = 1,\cdots,N, \\ & r_i = \sum_{j=1}^{N} r_{j\to i} \leq D_i, \forall i = 1,\cdots,N, \end{array} \tag{10}$$

where $r_{i\to i} \triangleq r_{s\to i}$. The complexity for the interior point method to solve this convex optimization is $O((N^2)^{3.5})$. For the case of $W_i = 1, D_i = \infty$, the optimal resource allocation is the same as that of Mutualcast. For general cases, we propose a suboptimal network resource allocation.

Consider a water-filling-type solution

$$\tilde{r}_i = \begin{cases} \sqrt{W_i}\cdot R, & \text{if } \sqrt{W_i}\cdot R < \tilde{D}_i, \\ \tilde{D}_i & \text{if } \sqrt{W_i}\cdot R \geq \tilde{D}_i, \end{cases} \tag{11}$$

where R is chosen such that

$$\sum_{i=1}^{N} \tilde{r}_i = U_s + \sum_{i=1}^{N} U_i - \max_k(\tilde{r}_k).$$

First construct the depth-2 trees with rates

$$r_{s\to i}^{(2)} = c\frac{U_i \max(\tilde{r}_k)}{\sum_{k=1}^{N}\tilde{r}_k - \tilde{r}_i}, \text{ and } r_{i\to j} = c\frac{U_i \tilde{r}_j}{\sum_{k=1}^{N}\tilde{r}_k - \tilde{r}_i}, \tag{12}$$

where c is chosen to be the largest possible value satisfying

$$\sum_{i=1}^{N} r_{s\to i}^{(2)} \leq U_s, \sum_{j=1,j\neq i}^{N} r_{i\to j} \leq U_i, \tag{13}$$

$$\beta_i \triangleq r_{s\to i}^{(2)} + \sum_{j=1,j\neq i}^{N} r_{j\to i} \leq \tilde{D}_i. \tag{14}$$

After constructing the depth-2 trees, the flow rate to peer i is β_i. The used source node's uplink is $c\alpha\max(\tilde{r}_k)$, where $\alpha = \sum_{i=1}^{N}\frac{U_i}{\sum_{k=1}^{N}\tilde{r}_k - \tilde{r}_i}$. If $c\alpha\max(\tilde{r}_k) < U_s$, we can further use the rest of the source node's uplink to distribute content through the depth-1 tree. The optimal resource allocation for the depth-1 tree is

$$r_{s\to i}^{(1)} = \begin{cases} \sqrt{W_i}\cdot R - \beta_i, & \text{if } \beta_i \leq \sqrt{W_i}\cdot R \leq \tilde{D}_i, \\ 0 & \text{if } \sqrt{W_i}\cdot R < \beta_i, \\ \tilde{D}_i - \beta_i, & \text{if } \sqrt{W_i}\cdot R > \tilde{D}_i, \end{cases} \tag{15}$$

and

$$r_i = r_{s\to i}^{(1)} + \beta_i, \tag{16}$$

where R is chosen such that $\sum_{i=1}^{N} r_{s\to i}^{(1)} = U_s - c\alpha\max(\tilde{r}_k)$. The complexity of calculating this resource allocation is $O(N^2)$.

D. Simulations

This section provides the empirical WSFT performance of the rateless-coding-based scheme, and compares it with the lower bound to the WSFT. In all simulations, the file size B is normalized to be 1. This section shows simulations for 4 cases of network settings as follows:

- Case I: $U_i = 1$, $D_i = \infty$ for $i = 1,\cdots,N$;
- Case II: $U_i = 1$, $D_i = 8$ for $i = 1,\cdots,N$;
- Case III: $U_i = i/N$, $D_i = 8i/N$ for $i = 1,\cdots,N$;
- Case IV: $U_i = 1 + 9\delta(i > N/2)$, $D_i = 8i/N$, $i = 1,\cdots,N$;

where $\delta(\cdot)$ is the indicate function.

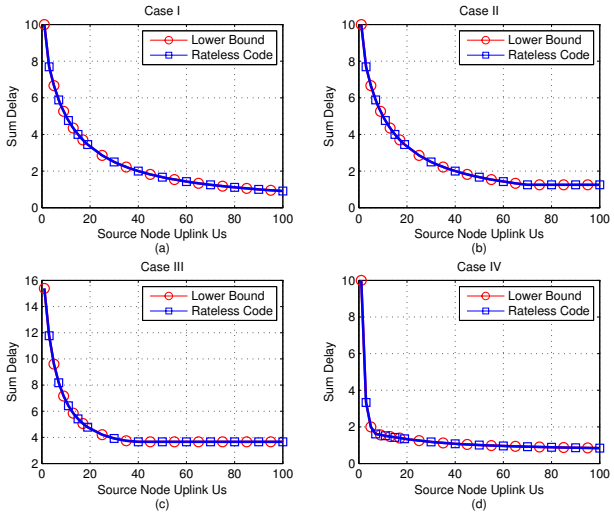

Fig. 3. Sum finish time versus U_s for P2P networks with $N = 10$ peers.

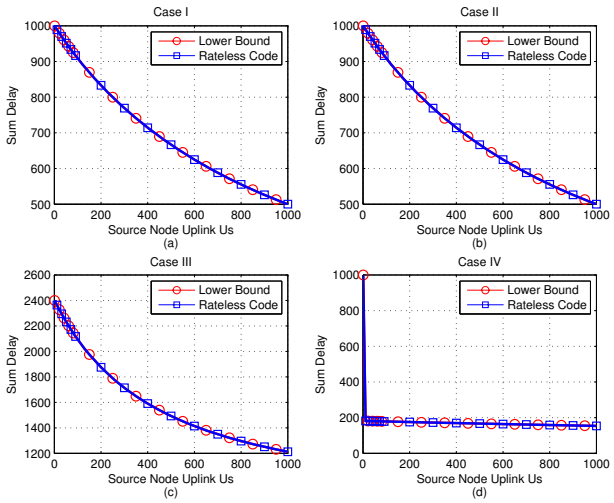

Fig. 4. Sum finish time versus U_s for P2P networks with $N = 1000$ peers.

The performances of sum finish time versus U_s for small P2P networks with $N = 10$ are shown in Fig. 3. The performances of sum finish time versus U_s for large P2P networks with $N = 1000$ are shown in Fig. 4.

In all these simulations, the WSFTs of the rateless-coding-based scheme achieve or almost achieve the lower bound. We also simulated for many other network settings and weight settings in [42]. In all simulations, the rateless-coding-based scheme achieves or almost achieves the lower bound to the WSFT. Hence, the lower bound to the WSFT is empirically tight, and the rateless-coding-based scheme has almost-optimal empirical performance.

IV. Dynamic File-Transfer Scenarios

The dynamic scenario is a file-transfer scheme which can re-construct the network topology and re-allocate the network resource whenever a peer finishes downloading, joins into the network, or leaves from the network. Wu et. al. showed in [37] that dynamic scenarios can provide significantly smaller sum finish time than static scenarios do for static P2P networks in which peer uplink is the only bottleneck. We propose a dynamic rateless-coding-based scheme for P2P network in which node uplinks and downlinks are the only bottlenecks. This scheme is applicable for not only static P2P networks but also dynamic P2P network which peers can join in or leave from.

A. Dynamic Rateless-Coding-Based Scheme

The key idea of this dynamic rateless-coding-based scheme is similar to that of the dynamic routing-based scheme in [37]. In particular, in each epoch, the scheme deploys all uplink resource to fully support several chosen peers. The details of the dynamic rateless-coding-based scheme is provided in Algorithm 1.

Algorithm 1 Dynamic Rateless-Coding-Based Scheme

1: Initiate the P2P network. Peers join into the network.
2: **while** A peer finishes downloading, joins into the network or leaves from the network **do**
3: Select a set of peers and reset peers' weights. (The peer selection and weight setting algorithm is provided in Algorithm 2)
4: Apply the static rateless-coding-based scheme based on the set weights until a peer finishes downloading, joins into the network or leaves from the network.
5: **end while**

Algorithm 1 provides the structure of the dynamic rateless-coding-based scheme. Because the peers always receive independently generated rateless coded chunks in the static rateless-code scheme, the dynamic rateless-coding-based scheme is also applicable for dynamic P2P network. As long as a peer receives enough rateless coded chunks [1], it can decode the whole file. The key issue is how to set the peer weights in each epoch. Since the weight setting and the static rateless-coding-based scheme in the current epoch will influence the dynamic scheme in the following epoches, the problem of setting weights is complex.

Theorem 3: The optimal network resource allocation in each epoch of a dynamic scenario is only obtained when some peers are fully supported, at most one peer is partially supported, and the other peers are not supported.

[1]The number of coded chunks needed to decode the whole file is only slightly larger than the total number of the original chunks.

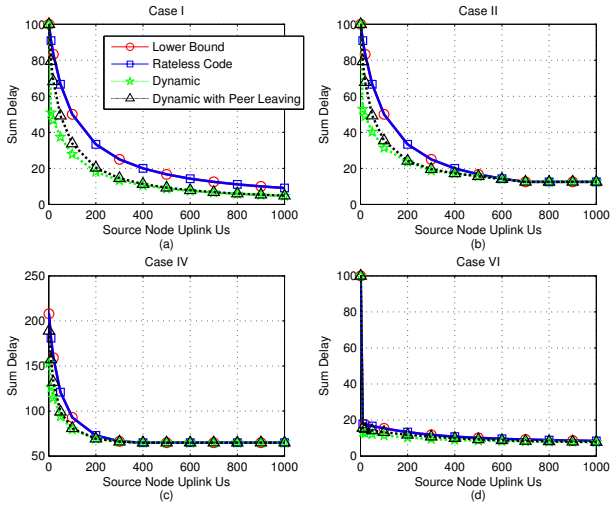

Fig. 5. Sum finish time versus U_s for P2P networks with $N = 100$ peers.

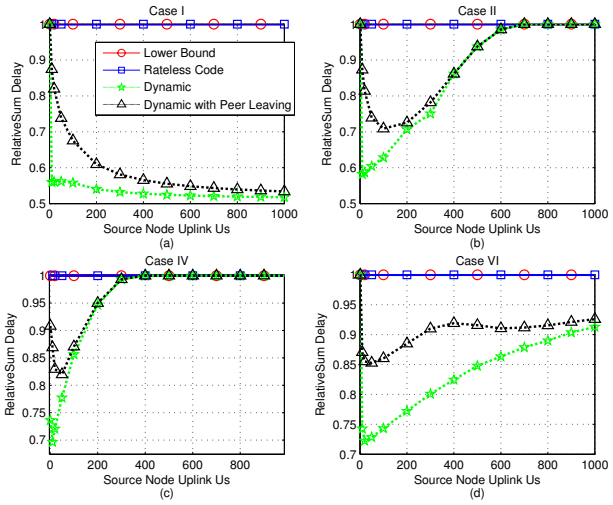

Fig. 6. Relative sum finish time versus U_s for P2P networks with $N = 100$ peers.

Proof: The proof is deferred to our journal manuscript available on ArXiv [42]. ■

Theorem 3 indicates that the dynamic scheme should deploy all uplink resource to fully support several peers in each epoch and partly support at most one peer. A sub-optimal peer selection algorithm and the corresponding weight setting is given in Algorithm 2.

Algorithm 2 Peer Selection and Weight Setting

1: Suppose N peers are downloading in the current epoch.
2: Let $B - q_i B$ $(0 < q_i \leq 1)$be the number of chunks that peer i has received for $i = 1, \cdots, N$.
3: Sort $\{\frac{W_i}{q_i}\}_{i=1}^{N}$ in descending order and get $(k_1, \cdots, k_N)$.
4: Find the smallest M such that $\sum_{i=1}^{M} \tilde{D}_{k_i} \geq U_s + \sum_{i=1}^{N} U_i$.
5: Select peers $\{k_i\}_{i=1}^{M}$ to fully support.
6: Set $W_j = 1$ if $j \in \{k_i\}_{i=1}^{M}$, or $W_j = 0$ otherwise.

B. Simulations

The dynamic rateless-coding-based scheme is applicable to both static P2P networks and dynamic P2P networks. Consider a type of dynamic P2P networks in which any peer leaves as soon as it finishes downloading, and no peer joins. This section provides the empirical WSFT performances of the dynamic rateless-coding-based scheme for static P2P networks and dynamic P2P networks with peer leaving, and compares them with those of the static scenarios for static P2P networks. In all simulations, the file size B is normalized to be 1.

Consider median-size P2P networks with $N = 100$ peers. The performances of sum finish time versus U_s for the 4 cases are shown in Fig. 5. Fig. 6 shows the relative value of the sum finish time by normalizing the minimum sum finish time for static scenarios to be 1 in order to explicitly compare the performances of the dynamic rateless-coding-based scheme and static scenarios. For Case I where peers have infinite downlink capacities, the sum finish time of the dynamic rateless-coding-based scheme is almost half of the minimum sum finish time for static scenarios for a broad range of the source node uplink U_s. This result matches the results in the previous work [37], which says that the minimum sum finish time of dynamic scenarios is almost half of the minimum sum finish time of static scenarios when peer uplink is the only bottleneck in the network. Our results also show that the sum finish time of the dynamic rateless-coding-based scheme with peer leaving decreases to almost half of the minimum sum finish time for static scenarios as U_s increases. For Cases II, III, and IV, the WSFTs of the dynamic scheme and the dynamic scheme with peer leaving are also always smaller than the minimum WSFT for static scenarios. In particular, the WSFT of the dynamic scheme can be as small as 0.59, 0.70, and 0.73 of the minimum WSFT for static scenarios for Cases II, III and IV, respectively. The WSFT of the dynamic scheme with peer leaving can be as small as 0.71, 0.82, and 0.86 of the minimum WSFT for static scenarios for Cases II, III and IV, respectively. These largest improvements in percentage of deploying the dynamic scheme is obtained when the source node can directly support tens of the peers. More simulations on the dynamic rateless-coding-based scheme are available in our journal manuscript [42].

V. Conclusions

This paper considers the problem of transferring a file from one source node to multiple receivers in a peer-to-peer (P2P) network in which both peer uplink and down-

link capacities are considered as possible bottlenecks. This paper shows that the static scenario can be optimized in polynomial time by convex optimization, and the associated optimal static WSFT can be achieved by linear network coding. This paper also proposes a static rateless-coding-based scheme which has almost-optimal empirical performance. Additionally, this paper proposes a dynamic rateless-coding-based scheme which provides significantly smaller WSFT than the optimal static scheme does.

References

[1] "BitTorrent.". *[Online]. Available: http://www.bittorrent.com.*

[2] "Napster.". *[Online]. Available: http://www.napster.com.*

[3] "Gnutella.". *[Online]. Available: http://www.gnutella.com.*

[4] "KaZaA.". *[Online]. Available: http://www.kazaa.com.*

[5] S. Androutsellis-Theotokis and D. Spinellis. "A survey of peer-to-peer content distribution technologies". *ACM Compl Surveys*, 36(4):335–371, Dec. 2004.

[6] J. Liu, S. G. Rao, B. Li, and H. Zhang. "Opportunities and challenges of peer-to-peer internet video broadcast". *Proceedings of the IEEE, Special Issue on Recent Advances in Distributed Multimedia Communications*, 2007.

[7] X. Zhang, J. Liu, B. Li, and T. S. P. Yum. "Coolstreaming/donet: A data-driven overlay network for efficient live media streaming". *in Proc. INFOCOM'05*, 2005.

[8] V. Pai, K. Kumar, K. Tamilmani, V. Sambamurthy, and A. E. Mohr. "Chainsaw: Eliminating trees from overlay multicast". *in Proc. 4th Int. Workshop on Peer-to-Peer Systems (IPTPS)*, Feb. 2005.

[9] J. Li, P. A. Chou, and C. Zhang. "Mutualcast: An Efficient Mechanism for Content Distribution in a Peer-to-Peer (P2P) Network". *Microsoft Research, MSR-TR-2004-100*, 2004.

[10] J. Li. "PeerStreaming: A practical receiver-driven peer-to-peer media streaming system". *Microsoft, Tech. Rep. MSR-TR-2004-101*, Sep. 2004.

[11] Z. Xiang, Q. Zhang, W. Zhu, Z. Zhang, and Y.-Q. Zhang. "Peer-to-peer based multimedia distribution service". *IEEE Trans. Multimedia*, 6(2):343–355, Apr. 2004.

[12] J. Jannotti, D. K. Gifford, K. L. Johnson, M. F. Kaashoek, and J. W. O'Toole. "Overcast: Reliable multicasting with an overlay network". *in Proc. of the Fourth Symposium of Operating System Design and Implementation (OSDI)*, pages 197–212, Oct. 2000.

[13] R. Ahlswede, N. Cai, S.-Y. R. Li and R. W. Yeung. "Network information flow". *IEEE Trans. on Information Theory*, 2000.

[14] S.-Y. R. Li, R. W. Yeung, and N. Cai. "Linear network coding". *IEEE Trans. on Information Theory*, 2003.

[15] R. Koetter, M. Medard. "An Algebraic Approach to Network Coding". *IEEE Trans. on Networking*, 2003.

[16] R. Koetter, M. Medard. "Beyond Routing: An Algebraic Approach to Network Coding". *Infocom*, 2003.

[17] T. Ho, D. Karger, M. Medard and R. Koetter. "Network Coding from a Network Flow Perspective". *ISIT*, 2003.

[18] P. A. Chou, Y. Wu, and K. Jain. "Practical network coding". *Allerton Conference on Communication, Control, and Computing*, 2003.

[19] S. Jaggi et al. "Polynomial time algorithms for multicast network code construction". *IEEE Transactions on Information Theory*, 51(6):1973–1982, 2005.

[20] T. Ho et al. "The Benefits of Coding over Routing in a Randomized Setting". *ISIT*, 2003.

[21] A. Ramamoorthy, J. Shi and R. D. Wesel. "On the Capacity of Network Coding for Random Networks". *IEEE Transactions on Information Theory*, 51(8):2878.

[22] A. Demers et. al. "Epidemic algorithms for replicated database maintenance". *In Proc. ACM Symposium on Principles of Distributed Computing*, 1987.

[23] R. Karp, C. Schindelhauer, S. Shenker, and B. Vocking. "Randomized rumor spreading". *In Proc. Foundations of Computer Science*, 2000.

[24] S. Boyd, A. Ghosh, B. Prabhakar, and D.Shah. "Gossip and mixing times of random walks on random graphs". *Preprint available at http://www.stanford.edu/?boyd/gossip gnr.html.*, 2004.

[25] D. Kempe, J. Kleinberg, and A. J. Demers. "Spatial gossip and resource location protocols". *In Proc. ACM Symposium on Theory of Computing*, 2001.

[26] D. Kempe and J. Kleinberg. "Protocols and impossibility results for gossip-based communication mechanisms". *In Proc. 43rd IEEE Symposium on Foundations of Computer Science*, 2002.

[27] S. Deb, M. Medard, C. Choute. "Algebraic gossip: a network coding approach to optimal multiple rumor mongering". *IEEE Transactions on Information Theory*, 52(6):2486.

[28] S. Sanghavi, B. Hajek, L. Massoulie. "Gossiping with multiple messages". *IEEE Transactions on Information Theory*, 53(12):4640.

[29] X. Yang and G. de Veciana. "Service capacity of peer to peer networks". *IEEE INFOCOM*, 2004.

[30] E. J. Cockayne and A. G. Thomason. "Optimal multimessage broadcasting in complete graphs". *Utilitas Math.*, 18:181.

[31] A. M. Farley. "Broadcast time in communication networks". *SIAM Journal on Applied Mathematics*, 39(2):385.

[32] A. Bar-Noy, S. Kipnis, and B. Schieber. "Optimal multiple message broadcasting in telephone-like communication systems". *Discrete Applied Mathematics*, 100:1.

[33] C.-H. Kwon and K.-Y. Chwa. "Multiple message broadcasting in communication networks". *Networks*, 26:253.

[34] J. Mundinger, R. Weber, and G. Weiss. "Optimal scheduling of Peer-to-Peer file dissemination". *Journal of Scheduling*, 2007.

[35] J. Mundinger and R. Weber. "Efficient file dissemination using peer-to-peer technology". *Technical Report 2004C01, Statistical Laboratory Research Reports*, 2004.

[36] S. Sengupta, M. Chen, P. A. Chou, and J. Li. "On optimality of routing for multi-source multicast communication scenarios with node uplink constraints". *IEEE ISIT*, 2008.

[37] Y. Wu, Y. C. Hu, J. Li, and P. A. Chou. "The Delay Region for P2P File Transfer". *in International Symposium of Information Theory 2009, Seoul Korea*, July 2009.

[38] M. Mehyar, W. Gu, S. H. Low, M. Effros, and T. Ho. "Optimal strategies for efficient peer-to-peer file sharing ". *ICASSP*, 2007.

[39] G. M. Ezovski, A. Tang, and L. L.H. Andrew. "Minimizing average finish time in P2P networks". *IEEE Infocom*, 2009.

[40] G. M. Ezovski, A. Anandkumar, A. Tang, and L. L.H. Andrew. "Min-Min times in peer-to-peer file sharing networks". *Allerton Conference on Communication, Control, and Computing*, 2008.

[41] C. S. Chang, T. Ho, M. Effros, M. Medard, B. Lenong . "Issues in peer-to-peer networking: a coding optimization approach". *IEEE NetCod*, 2010.

[42] B. Xie, M. van der Schaar, and R. D. Wesel . "Minimizing weighted sum download time for one-to-many file transfer in peer-to-peer networks". *arXiv:1002.3449v2 [cs.IT]*, 2011.

[43] D.J.C. MacKay. "Fountain codes". *in IEEE Proc.-Commun.*, (6), 2005.