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

Bike Xie, Mihaela van der Schaar and Richard D. Wesel
Department of Electrical Engineering, University of California, Los Angeles, CA 90095-1594
Email: xbk@ee.ucla.edu, mihaela@ee.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 delay (WSD) 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 network resource allocation remains static until all receivers finish downloading. This paper shows that the static scenario can be optimized in polynomial time by convex optimization, and the associated optimal static WSD 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-allocate the network resource during the file transfer. This paper proposes a dynamic rateless-coding-based scheme, which provides significantly smaller WSD than the optimal static scheme does.**

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

I. Introduction

P2P applications (e.g, BitTorrent [1]) 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 [2]. In a P2P file transfer application, the key performance metric from an end-user's point of view is the delay, or the time it takes for an end-user to download a file.

In [3], Li, Chou, and Zhang explore the problem of delivering the file to all receivers in the minimum amount of time assuming node uplinks are the only bottleneck in the network. They introduce a routing-based scheme, referred to as Mutualcast, which minimizes the maximum delay to all receivers with or without helpers.

In a recent paper, Wu et al. investigate the delay region for P2P file transfer in [4]. Given an order at which the receivers finish downloading, they demonstrate in [4] that the minimum weighted sum delay (WSD) 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 propose a routing-based scheme which has almost-optimal empirical performance and demonstrate how to significantly reduce the sum delay at the expense of a slight increase in the maximum delay.

This paper considers the problem of minimizing WSD for one-to-many file transfer in a P2P network. It is assumed that node uplinks and downlinks are the only bottlenecks in the network and that every node can connect to every other node through routing in the overlay. Most research in P2P considers 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 routing-based scheme proposed in [4]. For this reason, our paper considers heterogeneous peers with both uplink and downlink bandwidth constraints.

This paper is organized as follows. In Section II, definitions and notations for P2P networks are introduced. Section III studies static scenarios. Section III provides the optimal static scenario and a lower bound to the minimum WSD for static scenarios, proposes a static rateless-coding-based scheme, and simulates the performances. Section IV investigates the dynamic scenario, proposes a dynamic rateless-coding-based scheme and provides the simulation results. 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 that the source node has. 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 transmission rate from the source node to peer j as $r_{s\to j}$ and the 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 transmission rate 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

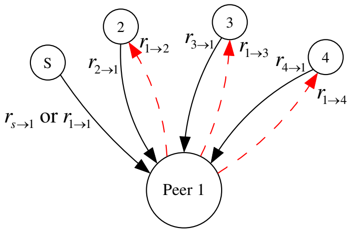


Fig. 1. The peer model

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.

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 delay to peer j for $j = 1, \cdots, N$. Given a static scenario $r_{i\to j}$, $(i, j = 1, \cdots, N)$, the maximum flow 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, $t_j \geq \frac{B}{r_j}, \forall j$. 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) delay 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 [5] [6] and the construction of the time-expanded graph [4], the set of the flow rates $\{r_j\}_{i=1}^{N}$ is achievable. Hence, the only Pareto optimal delay vector is $t_j = \frac{B}{r_j}$. ■

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:

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

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

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

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

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

By Lemma 1, the minimum WSD 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 delay 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 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 delays of $\frac{B}{\min(U_s, (U_s + \sum_{i=1}^{N} U_i)/N)}$ for all peers and obtains the minimum sum delay 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 [3], which minimizes the maximum delay to all peers. Hence, in this special case, the allocation of Mutualcast is the optimal static allocation which not only achieves the minimum sum delay but also minimizes the maximum delay of peers.

B. Bounding the Weighted Sum Delay 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

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

and

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

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

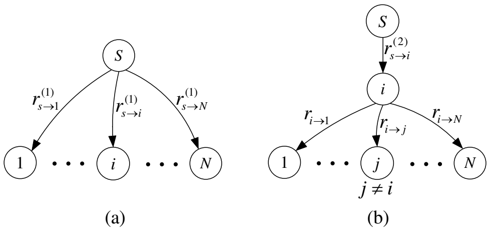
Consider the cut of $\{s\} \to \{1, \cdots, N\}$. r_i is also bounded by

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

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 WSD for static scenarios. The optimal solution to this relaxed problem is

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

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



(a) (b)

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

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 WSD 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 [3], can achieve the delay 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 Delay)** 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 delay, 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 which will be available on ArXiv. ■

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 [7]. 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 $B \simeq 10000$ [7]. This sub-section focuses on applying rateless erasure codes for P2P file transfer instead of designing rateless erasure codes. Hence, we 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

$$\tilde{r}_i = \left\{ \begin{array}{ll} \sqrt{W_i} \cdot R, & \text{if } \sqrt{W_i} \cdot R < \tilde{D}_i, \\ \tilde{D}_i & \text{if } \sqrt{W_i} \cdot R \geq \tilde{D}_i, \end{array} \right. \tag{11}$$

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

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

where c is chosen to be the largest possible value satisfying

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

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

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

$$r^{(1)}_{s\to i} = \left\{ \begin{array}{cl} \sqrt{W_i} \cdot R - \beta_i, & \text{if } \beta_i \leq \sqrt{W_i} \cdot R \leq \tilde{D}_i, \\ 0 & \text{if } \sqrt{W_i} \cdot R < \beta_i, \\ \tilde{D}_i - \beta_i, & \text{if } \sqrt{W_i} \cdot R > \tilde{D}_i, \end{array} \right. \tag{15}$$

and

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

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

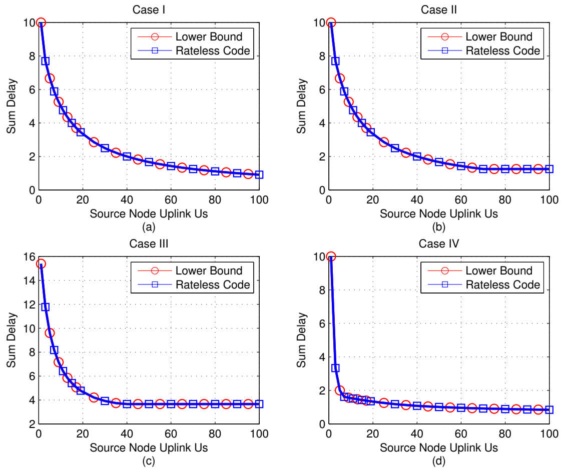


Fig. 3. Sum delay versus U_s for P2P networks with $N = 10$ peers.

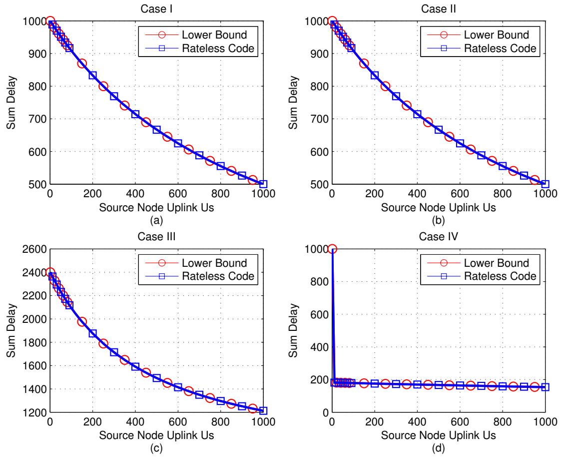


Fig. 4. Sum delay versus U_s for P2P networks with $N = 1000$ peers.

D. Simulations

This section provides the empirical WSD performance of the rateless-coding-based scheme, and compares it with the lower bound to the WSD. 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.

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

In all these simulations, the WSDs 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 all simulations, the rateless-coding-based scheme achieves or almost achieves the lower bound to the WSD. Hence, the lower bound to the WSD 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-allocate the network resource whenever a peer finishes downloading, joins into the network, or leaves from the network. Wu et al. show in [4] that dynamic scenarios can provide significantly smaller sum delay 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 [4]. 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

```
Initiate the P2P network. Peers join into the network.
while A peer finishes downloading, joins into the network or leaves from the network do
    Select a set of peers and reset peers' weights. (The peer selection and weight setting algorithm is provided in Algorithm 2)
    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.
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 partly supported, and the other peers are not supported.

Proof: The proof is deferred to our journal manuscript which will be available on ArXiv. ■

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.

---

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

Algorithm 2 Peer Selection and Weight Setting

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

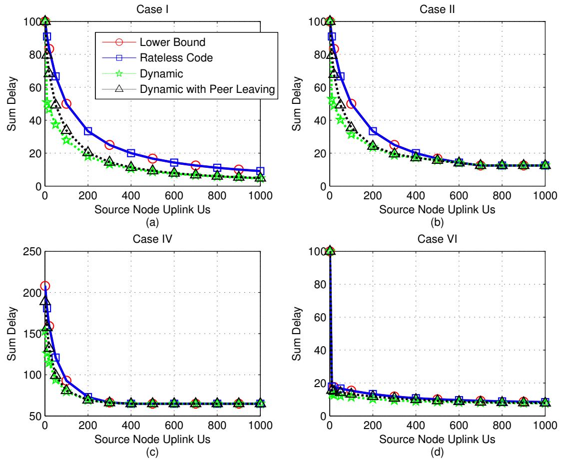


Fig. 5. Sum delay versus U_s for P2P networks with $N = 100$ peers.

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 WSD 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 delay versus U_s for the 4 cases are shown in Fig. 5. Fig. 6 shows the relative value of the sum delay by normalizing the minimum sum delay 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 delay of the dynamic rateless-coding-based scheme is almost half of the minimum sum delay for static scenarios for a broad range of the source node uplink U_s. This result matches the results in the previous work [4], which says that the minimum sum delay of dynamic scenarios is almost half of the minimum sum delay of static scenarios when peer uplink is the only bottleneck in the network. Our results also show that the sum delay of the dynamic rateless-coding-based scheme with peer leaving decreases to almost half of the minimum sum delay for static scenarios as U_s increases. For Cases II, III, and IV, the WSDs of the dynamic scheme and the dynamic scheme with peer leaving are also always smaller than the minimum WSD for static scenarios. In particular, the WSD of the dynamic scheme can be as small as 0.59, 0.70, and 0.73 of the minimum WSD for static scenarios for Cases II, III and IV, respectively. The WSD of the dynamic scheme with peer leaving can be as small as 0.71, 0.82, and 0.86 of the minimum WSD 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.

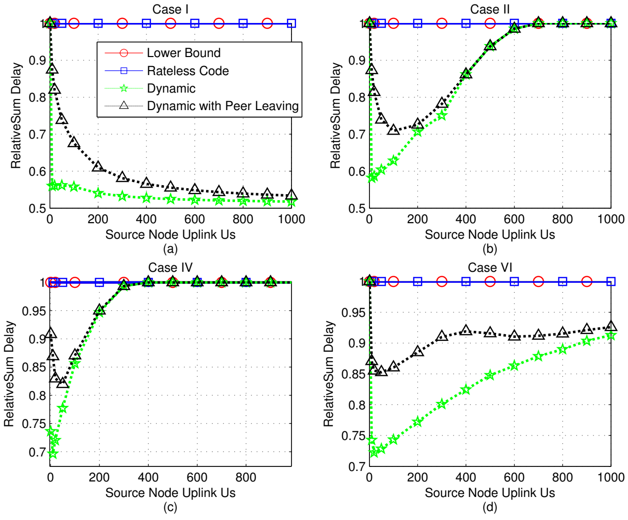


Fig. 6. Relative sum delay versus U_s for P2P networks with $N = 100$ peers.

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 peer uplink and downlink are the only bottlenecks. This paper shows that the static scenario can be optimized in polynomial time by convex optimization, and the associated optimal static WSD can be achieved by linear network coding. This paper proposes a static rateless-coding-based scheme which has almost-optimal empirical performances. This paper also proposes a dynamic rateless-coding-based scheme, which provides significantly smaller WSD than the optimal static scheme does.

REFERENCES

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

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

[3] 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.

[4] 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.

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

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

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