

# The Streaming Capacity of Sparsely Connected P2P Systems With Distributed Control

Can Zhao, *Associate Member, IEEE*, Xiaojun Lin, *Senior Member, IEEE*, and Chuan Wu, *Member, IEEE, ACM*

**Abstract**—Peer-to-peer (P2P) streaming technologies can take advantage of the upload capacity of clients, and hence can scale to large content distribution networks with lower cost. A fundamental question for P2P streaming systems is the maximum streaming rate that all users can sustain. Prior works have studied the optimal streaming rate for a complete network, where every peer is assumed to be able to communicate with all other peers. This is, however, an impractical assumption in real systems. In this paper, we are interested in the achievable streaming rate when each peer can only connect to a small number of neighbors. We show that even with a random peer-selection algorithm and uniform rate allocation, as long as each peer maintains  $\Omega(\log N)$  downstream neighbors, where  $N$  is the total number of peers in the system, the system can asymptotically achieve a streaming rate that is close to the optimal streaming rate of a complete network. These results reveal a number of important insights into the dynamics of the system, based on which we then design simple improved algorithms that can reduce the constant factor in front of the  $\Omega(\log N)$  term, yet can achieve the same level of performance guarantee. Simulation results are provided to verify our analysis.

**Index Terms**—Distributed control, live streaming, peer-to-peer (P2P) networks, sparse connectivity, streaming capacity.

## I. INTRODUCTION

WITH the proliferation of high-speed broadband services, the demand for rich multimedia content over the Internet, in particular high-quality video delivery over the Internet, has kept increasing. Streaming video directly from the server requires a large amount of upload bandwidth at the server, which can be very costly. The service quality can also be poor when the clients are far away from the server. In addition, it may be difficult for the server bandwidth to keep up when the demand is exceedingly high. There have been different approaches to offload traffic from the server, using either content distribution network (CDN) or peer-to-peer (P2P) technologies. Deploying a large CDN can introduce a

high fixed cost. In contrast, P2P technologies are particularly attractive because they take advantage of the upload bandwidth of the clients, which does not incur additional cost to the video service provider. Several well-known commercial P2P live streaming systems have been successfully deployed, include CoolStreaming [2], PPLIVE [3], TVAnts [4], UUSEE [5], and PPStream [6]. A typical P2P streaming system can now offer thousands of TV channels or movies for viewing and may serve hundreds of thousands of users simultaneously [5].

In contrast to the practical success of these P2P streaming systems, the theoretical understanding of the performance of P2P streaming seems to be lagging behind, which may impede further improvement of P2P live streaming. A basic question that can be asked is what the maximum streaming rate that all users can sustain for all possible policies is. This question has been studied under the assumption of a complete network, where each peer can connect to all other peers simultaneously. Under this assumption, the maximum streaming capacity has been found in [7], and both centralized and distributed rate allocation algorithms to achieve this maximum streaming capacity have been developed [7]–[10]. However, the assumption of a complete network is impractical for any large-scale P2P streaming systems. In a real P2P streaming system, typically each peer is only given a small list of other peers (which we refer to as neighbors) chosen from the entire population, and each peer can only connect to this subset of neighboring peers (neighbors may not be close in terms of physical distance). The number of neighboring peers is often much smaller than the total population, in order to limit the control overhead.

When each peer only has a small number of neighbors, the P2P network can be modeled as an incomplete graph with node-degree constraints. In this case, the streaming capacity of P2P systems becomes more complicated to characterize. Liu *et al.* [11] investigate the case when the number of downstream peers in a single substream tree is bounded. However, the number of neighbors that each peer could have over all substreams can still be very large (in the worse case, it can be connected to all the other peers simultaneously). Some approximated and centralized solutions to solve the optimal streaming capacity problem on a given incomplete network have been proposed in [12]. However, for large-scale P2P streaming systems, such a centralized approach will be difficult to scale. Liu *et al.* [13] proposed a Cluster-Tree algorithm to construct a topology subject to a bounded node-degree constraint, which could achieve a streaming rate that is close to the optimal streaming capacity of a complete network. This result gives us hope that, even with node-degree constraints, a P2P network may achieve almost the same streaming rate as that of a complete network. However, the Cluster-Tree

Manuscript received April 06, 2012; revised November 19, 2012; March 18, 2013; July 21, 2013; and February 03, 2014; accepted August 19, 2014; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor J. Wang. This work was supported in part by the NSF under Grants CNS-0643145, CNS-0721484, and CNS-0831999 and a grant from Hong Kong RGC under Contract HKU 717812. An earlier version of this paper has appeared in the 30th IEEE International Conference on Computer Communications (INFOCOM), Shanghai, China, April 10–15, 2011.

C. Zhao and X. Lin are with the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907 USA (e-mail: zhao43@purdue.edu; linx@purdue.edu).

C. Wu is with the Department of Computer Science, The University of Hong Kong, Hong Kong (e-mail: cwu@cs.hku.hk).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TNET.2014.2359963

algorithm is not a completely decentralized algorithm because it requires the tracker (a central entity) to apply the Bubble algorithm at the cluster level. The Bubble algorithm is a centralized algorithm. Some other works such as SplitStream [14] and Chainsaw [15] have also studied the problem of how to improve the streaming capacity when there is a node-degree constraint. However, these works did not provide theoretical results on the achievable streaming rate. To the best of our knowledge, there is no fully distributed algorithm in the literature that can achieve close-to-optimal P2P streaming capacity in incomplete networks.

In this paper, we are interested in the following question: Without centralized control, how many neighbors does a peer in a large P2P network need to maintain in order to achieve a streaming capacity that is close to the optimal streaming capacity of an otherwise complete network? Furthermore, can we develop fully distributed algorithms for peer selection and rate allocation to achieve the close-to-optimal streaming capacity? This paper provides some interesting and positive answers to these questions. We first show that, if each peer has  $\Omega(\log N)$  neighbors, where  $N$  is the total number of peers in the system, close-to-optimal streaming rate can be achieved with probability approaching 1 as  $N$  goes to infinity. Furthermore, in order to achieve this goal, each peer only needs to choose  $\Omega(\log N)$  downstream neighbors uniformly and randomly from the entire population, and simply allocates its upload capacity evenly among all downstream peers. Only the server needs a slightly different peer-selection policy (see Section II-B for details).

The results that we obtain have a similar flavor as scaling-law results in wireless ad hoc networks [16]. Although such results only hold when the size of the network  $N$  is large, they do provide important insights into the dynamics of the system. For example, our analysis indicates that, with a random peer selection strategy, for each user the most likely bottle neck for its streaming capacity is at the “last hop,” i.e., the sum of the upload capacity allocated to this user by its immediate upstream neighbors. This insight suggests that we could focus on balancing the capacity at the *last* hop when designing new distributed resource allocation algorithms for P2P streaming. Based on this insight, we then design an alternative algorithm that can substantially reduce the number of neighbors required to achieve the same probability of attaining the near-optimal streaming rate. This improved algorithm is still very simple and can be implemented in a distributed fashion. Hence, we believe that the insights from these results can be very helpful for designing more efficient control algorithms for P2P streaming. Finally, although due to space constraints we focus in this paper on single-channel P2P systems (i.e., only one video is served), we believe that the results and insights obtained here can also be generalized to multi-channel P2P systems [17]. Readers can refer to [1] for examples.

## II. SYSTEM MODEL AND MAIN RESULT

In this section, we will show that even without centralized control,  $\Omega(\log N)$  neighbors are sufficient for large P2P streaming networks. Specifically, we will show that just by letting each peer select its  $\Omega(\log N)$  neighbors randomly and do uniform rate allocation among these neighbors, the close-to-optimal streaming rate could be achieved with high probability when the network size  $N$  is large.

### A. System Model

We consider a peer-to-peer live streaming network with  $N$  peers and one source  $s$ . In the rest of the paper, we will use the terms “source” and “server” interchangeably. Similarly, we will use the terms “peer,” “node,” and “user” interchangeably. Denote the set of all peers and the source as  $V$  (thus,  $|V| = N + 1$ ). We assume that the source has an infinitely long video stream to be streamed to all peers and it has a fixed upload capacity  $u_s$ . Let  $U_i$  denote the upload capacity of peer  $i$ . For ease of exposition, we use a simple ON-OFF model to model the heterogeneity and random variation of the upload capacity: each peer has an upload capacity of  $U_i = u$  with probability  $p$  and an upload capacity of  $U_i = 0$  with probability  $1 - p$ , i.i.d. across peers. Thus, an ON peer represents a user with large upload capacity, while an OFF peer represents a user with low upload capacity. We assume that  $u_s \geq u$ . Like other works [7], [12], [13], [18], we assume that the download capacity and the core network capacity are sufficiently large, and hence the only capacity constraints are on the upload capacity. Each peer  $i \in V \setminus \{s\}$  has a fixed set  $\mathcal{E}_i$  of  $M$  downstream neighbors. Similarly, the source has a set  $\mathcal{E}_s$  of  $M$  downstream peers. We can then model the P2P network as a directed and capacitated random graph [19]. If  $j \in \mathcal{E}_i$ , assign a directed edge  $(i, j)$  from  $i$  to  $j$ . Let the set of all edges be  $E$ . Note that there may be multiple peers that have a common downstream neighbor. Define  $C_{ij}$  and  $C_{sj}$  be the streaming rate from peer  $i$  and source  $s$ , respectively, to peer  $j$ .

*Remark:* The above model seems to assume that each peer’s upload capacity is fixed in time. Nonetheless, we note that the results in the paper can also be applied to the case when the upload capacity is time-varying. Specifically, assume that the upload capacity follows a time-varying but stationary stochastic process. Then, the above model can be viewed as a snapshot of such a system at any given time instant. Hence, the results reported in the rest of the paper will also hold for each snapshot in time for such a system with a stationary marginal distribution. (Note that a similar “snapshot” assumption has also been used in other prior work, e.g., [7], [12], [13], and [18].) In addition, we note that the ON-OFF model can be viewed as the most extreme case of heterogeneous upload capacity. In fact, among all possible distributions of the peers’ upload capacity that are between  $[0, U_{\max}]$  and that have the same mean  $\mu$ , the ON-OFF model has the largest variance. Hence, the uncertainty/variability of the ON-OFF model will be the largest, and the performance of the system will also likely be the worst. Based on this relationship, we can also generalize the main conclusions of this paper to other distributions for the upload capacity (see also the numerical results in Section IV). However, due to space constraints, we have to omit the details. Interested readers can refer to our online technical report [20].

The values of  $\mathcal{E}_i$ ,  $\mathcal{E}_s$ ,  $C_{ij}$ , and  $C_{sj}$  depend on the peer-selection and rate-allocation algorithm. Given such an algorithm, we can define the “streaming capacity” of the system as the maximum rate that the source can distribute the streaming content to all peers. For example, for a complete network, we have  $\mathcal{E}_i = V \setminus \{i, s\}$  and  $\mathcal{E}_s = V \setminus \{s\}$ . Under such an idealized setting, [7] shows that the optimal streaming capacity is  $\min \left\{ u_s, \frac{u_s + \sum_{i \in V} U_i}{N} \right\}$ , and it can be achieved by setting

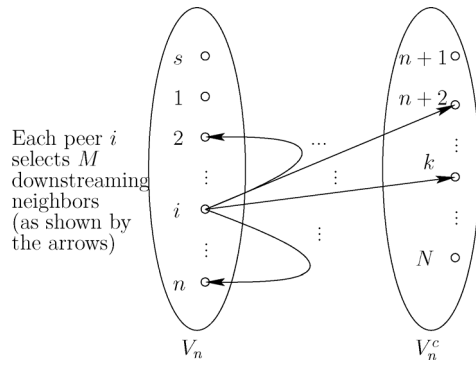


Fig. 1. Illustration of the neighbor selection and a cut.

$C_{ij} = U_i/(N - 1)$  and  $C_{sj} = U_s/N$  for all  $i, j$ . Note that the  $\min(\cdot)$  function is a concave function. Therefore, the expectation of the above optimal streaming capacity satisfies

$$\mathbf{E} \left[ \min \left\{ u_s, \frac{u_s + \sum_{i \in V} U_i}{N} \right\} \right] \leq \min \left\{ u_s, \frac{u_s + \sum_{i \in V} \mathbf{E}[U_i]}{N} \right\} \triangleq C_f. \quad (1)$$

For ease of exposition, we refer to  $C_f$  as “the optimal streaming capacity” throughout the rest of this paper. For our ON-OFF model of upload capacity, this optimal streaming capacity is equal to  $C_f = \min \{u_s, \frac{u_s}{N} + up\}$ . However, as we discussed in Section I, the assumption of a complete network is impractical. In this paper, we are interested in the streaming capacity of an incomplete network, which can be calculated by the minimum cuts. Specifically note that for a given user  $t$ , a cut that separates  $s$  and  $t$  is defined by dividing the peers in  $V$  into a set  $V_n$  of size  $(n + 1)$  that contains the server, and the complementary set  $V_n^c$  of size  $(N - n)$  that contains the peer  $t$ , i.e.,

$$s \in V_n, |V_n| = n + 1, t \in V_n^c \text{ and } |V_n^c| = N - n.$$

The capacity of the cut  $C_n$  is defined as  $C_n = \sum_{i \in V_n} \sum_{j \in V_n^c} C_{ij}$ . See Fig. 1 for illustration.

Let  $C_{\min}(s \rightarrow t)$  denote the minimum-cut capacity, which is the minimum capacity of all cuts that separate the source  $s$  and the destination  $t$ . It is well known that this min-cut capacity is equal to the maximum rate from  $s$  to  $t$ . Let  $C_{\min - \min}(s \rightarrow \mathcal{T})$  denote the min-min-cut that is the minimum cut of all individual min-cut capacities from the source to each destination  $t$  within a set  $\mathcal{T}$ , i.e.,

$$C_{\min - \min}(s \rightarrow \mathcal{T}) = \min_{t \in \mathcal{T}} C_{\min}(s \rightarrow t).$$

The streaming capacity of the network is then equal to  $C_{\min - \min}(s \rightarrow V \setminus \{s\})$  [21]. Note that given the graph and the capacity of each edge, this streaming capacity can be achieved with simple transmission schemes, e.g., with network coding [22], [23] or with a latest-useful-chunk policy [8]. However, it may require global knowledge and centralized control in order to optimally construct the network graph and allocate the upload capacity. A natural question is then the following: Without centralized control, can the streaming capacity over an

incomplete network approach the optimal streaming capacity  $C_f$  of a complete network? In Section II-B, we will provide a simple and distributed peer-selection and rate-allocation algorithm that can achieve this with high probability when the network size is large.

## B. Algorithms

We will now give explicit description of our simple control algorithm. First, we use a random peer-selection algorithm. Specifically, each peer randomly selects  $M$  downstream neighbors uniformly from all other peers. On the other hand, the server selects  $M$  downstream neighbors uniformly and randomly among the ON peers. We note that uniformly random peer-selection is very easy to implement in practice, even with dynamic peer arrivals and departures. Specifically, note that the number of upstream neighbors of a peer will be a binomial random variable  $X$  (sum of  $N$  Bernoulli random variables with mean  $\frac{M}{N}$ ). Note that the mean of  $X$  is  $M$ . Thus, when a new peer joins the system, it simply contacts  $X$  peers chosen uniformly randomly among the existing peers. Then, each contacted peer will choose one of its current downstream neighbor uniformly randomly, break this downstream connection, and take the new peer as the downstream neighbor. Furthermore, the new peer selects  $M$  downstream neighbors uniformly randomly among the existing peers. On the other hand, when a peer leaves the system, all of its upstream neighbors simply reselect a new downstream neighbor randomly. With this mechanism, it is easy to verify that, at any point in time, the set of  $M$  downstream neighbors of each peer is uniformly distributed among the current set of active peer.

Second, we use a uniform rate-allocation algorithm, i.e., each peer  $i$  simply divides its upload capacity equally among all of its downstream neighbors in  $\mathcal{E}_i$ . Therefore, each peer in the set  $\mathcal{E}_i$  receives a streaming rate  $U_i/M$  from peer  $i$ . Similarly, each downstream peer of the server receives  $U_s/M$  from the server. Under the above scheme, the link capacity  $C_{ij}$  is given by

$$C_{ij} = \begin{cases} U_i/M, & \text{if } j \in \mathcal{E}_i, i \neq s \\ U_s/M, & \text{if } j \in \mathcal{E}_s, i = s \\ 0, & \text{otherwise.} \end{cases}$$

Note that since  $\mathcal{E}_i$  and  $\mathcal{E}_s$  are chosen randomly,  $C_{ij}$ 's are also random variables. We define another import parameter for the total capacity that each peer  $i$  directly receives from its upstream neighbors, which is given by  $C_i^R = \sum_{j \in V} C_{ji}$ . We will see that this value is the main factor that determines the streaming capacity from the source to each node.

*Remark:* Since an ON peer represents a user with low upload capacity, the above scheme implies that, regardless of each user's upload capacity, it will choose the same number  $M$  of downstream neighbors uniformly and divide its capacity evenly among these downstream neighbors. In [20], we use this model and show that, even with a general distribution of upload capacity,  $\Omega(\log N)$  neighbors are still sufficient to attain a close-to-optimal streaming capacity. For details, please refer to [20].

Somewhat surprisingly, we will show that, as long as  $M = \Omega(\log N)$ , the algorithm achieves close-to-optimal streaming capacity, with probability approaching 1 as  $N \rightarrow \infty$  (Theorem 1).

*Remark:* Note that the server only chooses ON peers as its downstream neighbors. This is essential for achieving the close-to-optimal streaming capacity. To see this, note that the optimal streaming capacity  $C_f$  of a complete network is also constrained by the server capacity [see (1)]. If the server had used a substantial fraction of its upload capacity to serve ON peers, intuitively the rest of the peers would then suffer a lower streaming rate. With the same intuition, one would think that the peers directly connected to the server also need to be careful in choosing their downstream neighbors. However, this turns out to be unnecessary. For our main result (Theorem 1) to hold, no other peers (except the server) are required to differentiate their downstream neighbors. As readers will see, this is because those cuts with  $V_n$  only containing the downstream neighbors of  $s$  play a small role in the overall probability of attaining the close-to-optimal streaming capacity.

We also note that the above algorithm uses the “push” model, where upstream peers choose downstream neighbors. An alternate model is the “pull” model, where downstream peers choose upstream neighbors. Note that both models create a mesh-topology, and there is considerable symmetry between the two models. We use the push model in this paper because it is easier to analyze, although we believe that the main results of the paper can be generalized to the pull model, which we leave as future work.

### C. Main Result

*Theorem 1:* For any  $\epsilon \in (0, 1)$  and  $d > 1$ , there exist  $\alpha$  and  $N_0$  such that for any  $M = \alpha \log(N)$  and  $N > N_0$  the probability for the min-min-cut under the algorithm in Section II-B to be smaller than  $(1 - \epsilon)C_f$  is bounded by

$$\mathbf{P}(C_{\min-\min}(s \rightarrow V) \leq (1 - \epsilon)C_f) \leq O\left(\frac{1}{N^{2d-1}}\right).$$

Recall that the min-min-cut is equal to the streaming rate to all peers. Hence, Theorem 1 shows that as long as the number of downstream neighbors  $M$  is  $\Omega(\log N)$ , for any  $\epsilon \in (0, 1)$  the streaming rate of our algorithm will be larger than  $(1 - \epsilon)$  times the optimal streaming capacity with probability approaching 1 as the network size  $N$  increases.

### D. Proof of Theorem 1

We first find the min-cut for any fixed peer  $t$ . We will use a similar approach as the one in [19]. We will show that the probability for the capacity of a cut to be smaller than  $(1 - \epsilon)$  times its mean is very small, as  $N$  becomes large. Then, we will take the union bound over all cuts and show that overall probability is also very small. However, the techniques in [19] do not directly apply to our model due to the following two reasons. First, due to the ON-OFF model, there are fewer “ON” peers and hence the probability for each cut to fall below its expected value is larger than the case when all peers’ upload capacity is the same. However, there are still the same number of cuts we need to account for, which may cause the union bound in [19] to diverge. Second, the link capacity  $C_{ij}$  in [19] is assumed to be independent across  $j$ , which is not the case in our model. To address the first difficulty, we will first consider the subgraph that only contains the ON users, and hence the number of cuts is also reduced

correspondingly. To address the second difficulty, we will show that the joint distribution of  $C_{ij}$  can be approximated by i.i.d. random variables, which significantly simplifies the analysis.

We first introduce the following general relationship between the min-cut from the server  $s$  to the peer  $t$  in a random graph  $G$  and the min-cut from the server  $s$  to the peer  $t$  in the any subgraph  $H_t$  of  $G$  that contains  $s$  and  $t$ .

*Proposition 2:* Let  $G$  be a random graph defined on some probability space  $\Omega$  that has a fixed source  $s$  and a fixed destination  $t$ . Let  $H_t$  be another random graph defined on the same probability space such that  $H_t(\omega) \subseteq G(\omega)$  for all  $\omega \in \Omega$  and  $H_t$  contains  $s$  and  $t$ . Then, for any given positive value  $C$ , the following holds:

$$\mathbf{P}(C_{\min,G}(s \rightarrow t) \leq C) \leq \mathbf{P}(C_{\min,H_t}(s \rightarrow t) \leq C) \quad (2)$$

where  $C_{\min,G}(s \rightarrow t)$  is the min-cut in  $G$  from  $s$  to  $t$ , and  $C_{\min,H_t}(s \rightarrow t)$  is the min-cut in  $H_t$  from  $s$  to  $t$ .

*Proof:* Let  $\mathcal{A} = \{\omega : C_{\min,G(\omega)}(s \rightarrow t) \leq C\}$  and  $\mathcal{B} = \{\omega : C_{\min,H_t(\omega)}(s \rightarrow t) \leq C\}$ . For any  $\omega \in \mathcal{A}$ , the min-cut from  $s$  to  $t$  in the graph  $G(\omega)$  is less than  $C$ . Since  $H_t$  is a subgraph of  $G(\omega)$ , the min-cut from  $s$  to  $t$  in  $H_t(\omega)$  is smaller than the min-cut in  $G$ , i.e.,  $C_{\min,H_t(\omega)}(s \rightarrow t) \leq C_{\min,G(\omega)}(s \rightarrow t) \leq C$ . Hence,  $\omega \in \mathcal{B}$ . We then have  $\mathcal{A} \subseteq \mathcal{B}$ , and (2) holds consequently.  $\square$

Proposition 2 is intuitive because every cut in  $G(\omega)$  has a larger capacity than the corresponding cut in the subgraph  $H_t(\omega)$ . For a given destination  $t$ , let  $H_t(W, F)$  be the subgraph of  $G(V, E)$  such that  $W$  contains the peer  $t$ , the server, and all of the nodes whose channel condition is ON, and  $F \subset E$  are those edges between nodes in  $W$ . The capacity of the edges in  $F$  is the same as the capacity of the edges in  $E$ . Proposition 2 allows us to focus on the subnetwork  $H_t$  instead of the entire network  $G$ . Assume that there are  $Y$  ON peers in the network excluding peer  $t$ , and thus  $|W| = Y + 2$ . Clearly,  $Y$  is a random variable with binomial distribution with parameter  $N - 1$  and  $p$ . For ease of exposition, we assume that  $Y$  is fixed during the following discussion for one given cut, and we will consider the randomness of  $Y$  later when we take the union bound over all cuts. We define a cut on  $H_t$  by dividing the peers in  $W$  into a set  $W_m$  of size  $m + 1$  that contains the server, and the complementary set  $W_m^c$  of size  $Y - m + 1$  that contains peer  $t$ . The capacity of the cut  $D_m$  is then given by

$$D_m = \sum_{k \in W_m^c} C_{sk} + \sum_{i \in W_m} \sum_{k \in W_m^c} C_{ik}. \quad (3)$$

Note that for each peer  $i \in W_m$  (and  $i \neq s$ ), we have  $\sum_{k \in W_m^c} C_{ik} = L_i u / M$ , where  $L_i$  is the number of downstream neighbors of peer  $i$  that are in the set  $W_m^c$ . Note that the value of  $L_i$  must satisfy  $\max\{0, M - (N - Y + m - 2)\} \leq L_i \leq \min\{M, Y - m + 1\}$ . Since downstream neighbors of peer  $i$  are uniformly chosen from other peers, we have

$$\mathbf{P}\left(\sum_{k \in W_m^c} C_{ik} = l \cdot \frac{u}{M}\right) = \frac{\binom{Y-m+1}{l} \binom{N-Y+m-2}{M-l}}{\binom{N-1}{M}}.$$

This is the probability that  $l$  out of  $M$  downstream neighbors of peer  $i$  are in  $W_m^c$  (of size  $Y - m + 1$ ) and  $M - l$  of them are in

the set  $V \setminus W_m$ . The distribution of  $L_i$  is known as a hyper-geometric distribution with expectation  $\frac{(Y-m+1)M}{N-1}$  [24, p. 167]. We can get a similar expression for the source  $s$ , i.e.,

$$\mathbf{P} \left( \sum_{i \in W_m^c} C_{si} = l \cdot \frac{u_s}{M} \right) = \begin{cases} \frac{\binom{Y-m}{l} \binom{M}{M-l}}{\binom{M}{M}}, & \text{if } t \text{ is OFF} \\ \frac{\binom{Y-m+1}{l} \binom{M-l}{M-l}}{\binom{Y+1}{M}}, & \text{if } t \text{ is ON.} \end{cases}$$

$$\mathbf{E} \left[ \sum_{i \in W_m^c} C_{si} \right] = \begin{cases} \frac{u_s(Y-m)}{Y}, & \text{if } t \text{ is OFF} \\ \frac{u_s(Y+1-m)}{Y+1}, & \text{if } t \text{ is ON.} \end{cases}$$

Hence, we obtain the expectation of  $D_m$  as

$$\mathbf{E}[D_m] = \mathbf{E} \left[ \sum_{k \in W_m^c} C_{sk} \right] + \sum_{i \in W_m} \mathbf{E} \left[ \sum_{k \in W_m^c} C_{ik} \right]$$

$$= \begin{cases} \frac{u_s(Y-m)}{Y} + \frac{u}{N-1}m(Y-m+1), & \text{if } t \text{ is OFF} \\ \frac{u_s(Y+1-m)}{Y+1} + \frac{u}{N-1}m(Y-m+1), & \text{if } t \text{ is ON.} \end{cases} \quad (4)$$

Next, we are interested in the probability that  $D_m \geq (1 - \epsilon)\mathbf{E}[D_m]$  for all  $m$  for a given constant  $\epsilon \in (0, 1)$ . In other words, this is the probability that the min-cut value is no less than  $(1 - \epsilon)$  times its average. For all  $m$ , it is not hard to see

$$\mathbf{E}[D_m] \geq \min\{\mathbf{E}[D_0], \mathbf{E}[D_Y]\} = \min \left\{ u_s, \frac{u_s}{Y} + \frac{Y}{N-1}u \right\}.$$

If we have  $Y \geq (1 - \epsilon)p(N - 1)$ , we will get

$$\mathbf{E}[D_m] \geq (1 - \epsilon) \min \left\{ u_s, \frac{u_s}{N} + pu \right\}.$$

Recall that  $C_f = \min\{u_s, \frac{u_s}{N} + pu\}$  is the optimal streaming capacity assuming a complete network [7]. Hence,  $D_m \geq (1 - \epsilon)\mathbf{E}[D_m]$  then implies that  $D_m \geq (1 - \epsilon)^2 C_f$ . In other words, the probability that  $D_m \geq (1 - \epsilon)\mathbf{E}[D_m]$  for all  $m$  becomes a lower bound for the probability that the min-cut is no less than  $(1 - \epsilon)^2 C_f$ . In the following, we will derive  $\mathbf{P}(D_m \geq (1 - \epsilon)\mathbf{E}[D_m])$ . We will use the moment generating function for  $D_m$ . Before we go further, we need to address the second difficulty we mentioned above, i.e., the  $C_{ij}$ 's are correlated across  $j$ . To remove the coupling, we need to introduce the notion of negatively related for Bernoulli random variables [25], [26].

*Definition 3:* The Bernoulli random variables  $\tilde{I}_i, i = 1, \dots, n$ , are said to be negatively related if for each  $i \leq n$  there exists random variables  $J_{ij}$ , such that the distribution of the random vector  $[J_{i1}, J_{i2}, \dots, J_{in}]$  is equal to the conditional distribution of the random vector  $[I_1, I_2, \dots, I_n]$  given that  $I_i = 1$ , and  $J_{ij} \leq I_j$  for  $j \neq i$ .

For negatively related random variables, the following theorem holds ([26, Theorem 4]).

*Theorem 4:* Suppose  $I_i$ 's are negatively related Bernoulli random variables with identical distribution,  $i = 1, 2, \dots, n$ . Let  $\tilde{I}_i, i = 1, 2, \dots, n$ , be i.i.d. random variables, where  $\tilde{I}_i$  has the same distribution as  $I_i$  for all  $i$ . Then, for any real  $t$

$$\mathbf{E} \left[ e^{t \sum_{i=1}^n I_i} \right] \leq \mathbf{E} \left[ e^{t \sum_{i=1}^n \tilde{I}_i} \right].$$

Theorem 4 thus allows us to bound the moment generating function of negatively related random variables by that of independent random variables. Its intuition can be explained as follows. Roughly speaking, for negatively related Bernoulli random variables, conditioned on the event that one of them is 1, the others are more likely to be small. Correspondingly, conditioned on the event that one of them is 0, the others are more likely to be large. Therefore, when  $t > 0$ , the moment generating function is mainly determined by the probability of the sum of all indicator random variables achieving the larger value. The sum of negatively related random variables is less likely to achieve a larger value, and hence the value of the moment generation function is smaller. For  $t < 0$ , the moment generating function is mainly determined by the probability of the sum of all indicator random variables achieving the smaller value. The sum of negatively related random variables is also less likely to achieve a smaller value, and hence the value of the moment generation function is smaller.

One can show that hyper-geometric random variables can be viewed as the sum of negatively related Bernoulli random variables (see [26, Example 1]). Specifically, we first construct  $I_i$  by choosing  $M$  neighbors out of  $N - 1$  peers. For each peer  $i$  on the right, let  $I_i = 1$  if peer  $i$  is chosen as a neighbor, and let  $I_i = 0$  otherwise (note that  $I_i$  is not defined for peers on the left). We can then construct  $J_{ij}$  as follows. First, set  $J_{ij} = I_j$  for all  $j$ . Then, if  $J_{ii} = 0$ , in order to make  $J_{ii} = 1$ , we choose one neighbor  $k$  randomly (either from the left or the right) and exchange that neighbor with peer  $i$ . If  $k$  was on the left, we then let  $J_{ii} = 1$ . If  $k$  was on the right, we then let  $J_{ii} = 1$  and  $J_{ik} = 0$ . Clearly,  $\mathbf{J}_i$  has the same distribution as  $\mathbf{I}$  given that  $I_i = 1$ . However, by our construction,  $J_{ij} \leq I_j$  for all  $j \neq i$ . Hence,  $I_i, i = 1, \dots, M$ , are negatively related. We can now use Theorem 4 to bound the moment generation function of  $\sum_{k \in W_m^d} C_{ik}$  by the moment generating functions of the sum of i.i.d. random variables. Toward this end, we have the following proposition.

*Proposition 5:* For any given cut  $V_k$  and  $V_k^c$  of a network  $G(V, E)$ , let  $\tilde{W}_1$  and  $\tilde{W}_2$  be subsets of  $V_k$  and  $V_k^c$ , respectively. Assume that  $|\tilde{W}_1| = q \leq k + 1$  and  $|\tilde{W}_2| = r \leq N - k$ . Let the upload capacity of each peer  $i \in \tilde{W}_1$  be  $u$ . For each peer in  $\tilde{W}_1$ , it chooses  $M$  downstream neighbors uniformly and randomly from a given subset  $\tilde{V}$  of  $V$  that is a superset of  $\tilde{W}_2$ . Let  $\tilde{N} = |\tilde{V}|$ . Then the moment generating function of  $\sum_{i \in \tilde{W}_1} \sum_{j \in \tilde{W}_2} C_{ij}$  satisfies

$$\mathbf{E} \left[ e^{-\theta \sum_{i \in \tilde{W}_1} \sum_{j \in \tilde{W}_2} C_{ij}} \right] \leq \exp \left[ Mq \frac{r}{\tilde{N}} \left( e^{-\theta \frac{u}{M}} - 1 \right) \right]. \quad (5)$$

Note that the right-hand side of (5) can be viewed as the moment generating function of  $\sum_{i \in \tilde{W}_1} \sum_{j \in \tilde{W}_2} C_{ij}$  assuming that  $C_{ij}$ 's are independent. Proposition 5 then follows from Theorem 4 and the negatively related property discussed above. The detailed proof of Proposition 5 is available in [20]. Proposition 5 combined with the Chernoff bound will be frequently used to estimate the probability for a cut to "fail," i.e., the capacity of a cut being less than  $(1 - \epsilon)$  times its expected capacity. Recall that the capacity  $D_m$  of the cut  $W_m$  is given by (3). Then, by taking  $\tilde{W}_1$  and  $\tilde{W}_2$  in Proposition 5 to be  $W_m$  and

$W_m^c$ , respectively, we can show the following result for the cut  $W_m$  in  $H_t$  under the assumption of ON-OFF upload capacities.

*Lemma 6:* Let  $\epsilon \in (0, 1)$ . Given that the total number of ON peers in the entire network  $Y$  is equal to  $y$ , the probability that the capacity  $D_m$  of the cut  $W_m$  in  $H_t$  is less than  $(1 - \epsilon)\mathbf{E}[D_m]$  can be bounded by the following:

$$\begin{aligned} \mathbf{P}(D_m \leq (1 - \epsilon)\mathbf{E}[D_m] | Y = y) \\ \leq \exp \left[ - \left( Mm \frac{y - m + 1}{N - 1} + M \frac{y - m}{y} \right) \frac{u}{u_s} \frac{\epsilon'^2}{2} \right]. \end{aligned}$$

The proof of Lemma 6 can be found in Appendix A. Lemma 6 gives us an upper bound on the probability that the capacity  $D_m$  of a cut  $W_m$  is less than  $1 - \epsilon$  times its mean conditioned on the event that the total number of ON peers  $Y$  is equal to  $y$ . Note that  $Mm \frac{y - m + 1}{N}$  is the average number of edges from peers in  $W_m$  to peers in  $W_m^c$ , while  $M \frac{y - m}{y}$  is a lower bound on the average number of edges from the server to peers in  $W_m^c$ . Hence, the upper bound in Lemma 6 decreases exponentially if the average number of edges increases. Furthermore, since the average number of edges is proportional to  $M$ , the upper bound also decreases exponentially if  $M$  increases. We will use Lemma 6 for each  $m = 1, 2, \dots, Y$ . The following lemma then bounds the effect of all cuts separating  $s$  and  $t$ . Note that for each value of  $m$ , there are  $\binom{Y}{m}$  possible cuts  $W_m$ . Due to symmetry, the capacity of all  $\binom{Y}{m}$  cuts has the same distribution.

*Lemma 7:* Define  $\tilde{B}_m$  to be the event  $\{D_m \leq (1 - \epsilon)C_f \text{ for any cut } W_m \text{ among the } \binom{Y}{m} \text{ cuts}\}$ . Suppose that there exists  $\eta \in (0, 1)$  such that for any  $y \geq \eta pN$  and any integer  $m$  between 0 and  $y$ , the following holds for  $\beta = \exp\left(-M \frac{u}{u_s} \frac{\epsilon'^2}{2}\right)$  and  $\gamma = \eta p$ :

$$\mathbf{P}(D_m \leq (1 - \epsilon)C_f | Y = y) \leq \beta^m \frac{y - m + 1}{N - 1} + \frac{y - m}{y}.$$

Then, the probability of the union of all  $\tilde{B}_m$ 's is bounded by

$$\mathbf{P} \left( \bigcup_{m=0}^Y \tilde{B}_m \right) \leq O(e^{-(1-\eta)^2 p^2 N}) + \beta^\gamma \left[ \left(1 + p\beta^{\frac{\gamma}{2}}\right)^{N-1} \right].$$

In addition, we can separate the union bound into two parts

$$\begin{aligned} \mathbf{P} \left( \bigcup_{m=0}^{Y-1} \tilde{B}_m \right) &\leq O(\exp(-(1-\eta)^2 p^2 N)) \\ &\quad + \beta^\gamma \left[ \left(1 + p\beta^{\frac{\gamma}{2}}\right)^{N-1} - 1 \right] \quad (6) \end{aligned}$$

$$\mathbf{P}(\tilde{B}_Y) \leq O(\exp(-(1-\eta)^2 p^2 N)) + \beta^\gamma. \quad (7)$$

Lemma 7 is obtained by taking the union bound over all cuts. The detailed proof of Lemma 7 is in [20]. Combing Lemmas 6 and 7, we can now prove Theorem 1.

*Proof of Theorem 1:* According to Proposition 2 and Lemma 7, for any peer  $t$ , the minimum cut from the source  $s$  to  $t$  can be bounded by

$$\begin{aligned} \mathbf{P}(C_{\min}(s \rightarrow t) \leq (1 - \epsilon)C_f) \\ \leq \mathbf{P}(C_{\min, H_t}(s \rightarrow t) \leq (1 - \epsilon)C_f) = \mathbf{P} \left( \bigcup_{m=0}^Y \tilde{B}_m \right). \quad (8) \end{aligned}$$

Recall that if  $Y \geq \sqrt{1 - \epsilon}pN$ ,  $D_m \geq \sqrt{1 - \epsilon}\mathbf{E}[D_m]$  implies  $D_m \geq (1 - \epsilon)C_f$ . By Lemma 6, letting  $\epsilon' = 1 - \sqrt{1 - \epsilon}$  and  $\beta = \exp\left(-M \frac{u}{u_s} \frac{\epsilon'^2}{2}\right)$ , we have if  $Y \geq (1 - \epsilon')pN$

$$\begin{aligned} \mathbf{P}(D_m \geq (1 - \epsilon)C_f) \\ \leq \mathbf{P}(D_m \geq (1 - \epsilon')\mathbf{E}[D_m]) \\ \leq \exp \left[ - \left( Mm \frac{y - m + 1}{N - 1} + M \frac{y - m}{y} \right) \frac{u}{u_s} \frac{\epsilon'^2}{2} \right] \\ = \beta^m \frac{y - m + 1}{N - 1} + \frac{y - m}{y}. \end{aligned}$$

Now let  $\eta = 1 - \epsilon'$  and apply Lemma 7 to (8). We get

$$\begin{aligned} \mathbf{P}(C_{\min}(s \rightarrow t) \leq (1 - \epsilon)C_f) \\ \leq \mathbf{P} \left( \bigcup_{m=0}^Y \tilde{B}_m \right) \\ \leq 2\beta^\gamma \left(1 + p\beta^{\frac{\gamma}{2}}\right)^{N-1} + O(\exp(-\epsilon'^2 p^2 N)). \end{aligned}$$

Note that by assumption,  $M = \alpha \log(N)$ . For any  $\epsilon > 0$  and  $\epsilon' = 1 - \sqrt{1 - \epsilon}$ , choose a sufficiently large  $\alpha$  such that  $\alpha \geq \frac{4du_s}{\gamma u \epsilon'^2}$ . We then have, for large  $N$

$$\beta^\gamma = \exp\left(-M\gamma \frac{u}{u_s} \frac{\epsilon'^2}{2}\right) = \exp(-2d \log(N)) = 1/N^{2d}.$$

Hence, the minimum cut satisfies

$$\begin{aligned} \mathbf{P}(C_{\min}(s \rightarrow t) \leq (1 - \epsilon)C_f) \\ \leq \frac{2}{N^{2d}} \left(1 + pO\left(\frac{1}{N^d}\right)\right)^{N-1} = O\left(\frac{1}{N^{2d}}\right). \end{aligned}$$

Thus, the min-min cut satisfies

$$\begin{aligned} \mathbf{P}(C_{\min - \min} \leq (1 - \epsilon)C_f) \\ \leq \sum_{t=1}^N \mathbf{P}(C_{\min}(s \rightarrow t) \leq (1 - \epsilon)C_f) \\ \leq O\left(\frac{1}{N^{2d}}\right) \cdot N = O\left(\frac{1}{N^{2d-1}}\right). \end{aligned}$$

□

We remark on several implications of Theorem 1. First, Theorem 1 not only shows that pure random selection is sufficient to achieve close-to-optimal streaming capacity as long as each peer has  $\Omega(\log N)$  downstream neighbors; it also reveals important insights on the significance of different types of cuts. To see this, note that if we choose  $\alpha$  as in the proof such that  $\beta^\gamma = O(1/N^{2d})$ , we have [from (6)]

$$\begin{aligned} \mathbf{P} \left( \bigcup_{m=0}^{Y-1} \tilde{B}_m \right) &\leq 2\beta^\gamma \left[ \left(1 + p\beta^{\frac{\gamma}{2}}\right)^{N-1} - 1 \right] \\ &= O(1/N^{2d})O(e^{1/N^{d-1}} - 1) = o(1/N^{2d}). \end{aligned}$$

On the other hand, we have  $\mathbf{P}(\tilde{B}_Y) = O(1/N^{2d})$ . Hence, the probability that the last cut (the  $W_Y$  and  $W_Y^c$  cut) fails is much larger than the probability that any other cut fails. Thus, for each peer  $t$ , the min-cut from the source to  $t$  is mainly determined by  $C_t^R$  (recall that  $C_t^R$  is the total capacity received by peer  $t$  directly from its upstream neighbors, which is also the capacity of the last cut).

The above insight suggests that if we want to design improved distributed control algorithms for P2P streaming systems, we may want to focus on improving the capacity  $C_t^R$  at the last hop. Note that one of the main reasons for  $C_t^R$  to fall below its mean value is the imbalance of  $C_t^R$  across  $t$ . More specifically, some peers  $t$  may have a larger number of upstream peers, and hence have a larger-than-average value of  $C_t^R$ , while other peers may have a smaller-than-average value of  $C_t^R$ . Such imbalance will lead to an increase in the probability that some peers have low streaming rates. Based on this intuition, we can design a slightly more sophisticated scheme to balance the value of  $C_t^R$  of different peers, which will be discussed explicitly in Section III.

Theorem 1 also reveals important relationships between the number of neighbors required and key system parameters. For example, if we require a better performance (smaller  $\epsilon$  or larger  $d$ ) or have fewer ON peers (smaller  $p$ ), the number of downstream neighbors needed by each peer will increase. Specifically, according to the proof, we need  $\alpha \geq \frac{4du_s}{\gamma u \epsilon^{1/2}}$ . If we require a higher streaming rate or a faster convergence rate, i.e.,  $\epsilon$  is smaller (consequently  $\epsilon'$  is smaller) or  $d$  is larger, we will need a larger  $\alpha$ . If the probability that a peer is ON is reduced, i.e.,  $p$  is reduced, we will also need a larger  $\alpha$ .

### III. IMPROVED HYBRID ALGORITHM

In Section II, we proposed a simple scheme with random neighbor selection and uniform rate allocation that can sustain a close-to-optimal streaming rate for all users. Our scheme only requires  $O(\log N)$  neighbors for each peer. However, our simulation results (see Section IV) indicate that the number of neighbors that each peer needs may still be quite large. This is because the actual number of neighbors required also depends on the constant factor  $\alpha$  before the  $\log N$  term. As in the remarks following Theorem 1, for uniform rate-allocation schemes, we need  $\alpha \geq \frac{4du_s}{\gamma u \epsilon^{1/2}}$ , which increases inversely proportional to the square of  $\epsilon'$ . The goal of this section is to study whether we can design a slightly more sophisticated scheme for neighbor selection and/or rate allocation that can significantly reduce the constant factor  $\alpha$ . Specifically, our strategy is to retain the random peer-selection algorithm but focus on improving the rate allocation algorithm. One may argue that random peer-selection may still be suboptimal. However, as we explain in Section II-B, random peer selection has the advantage that it is very easy to implement and robust to peer dynamics. In contrast, other peer-selection algorithms (e.g., based on forming tree [13]) will likely be more costly in the presence of peer dynamics. Since our goal in this paper is both to attain a close-to-optimal streaming capacity and to use simple, robust, and distributed control, we believe that the choice of using random peer selection strikes a reasonable tradeoff. In fact, as we will show, even by improving the rate allocation alone, significant performance improvement can be attained.

As we observed in earlier sections, with high probability, the bottleneck for uniform rate allocation lies in the last hop, i.e., the total upload capacity allocated to some peers from their immediate upstream neighbors is smaller than average. Hence, a natural idea is to design a more sophisticated rate-allocation scheme such that the capacity of the last hop is more balanced, and therefore we may be able to reduce the number of neighbors that each user needs in order to achieve a close-to-optimal

streaming rate. More specifically, we may find  $C_{ij} \geq 0$ ,  $i, j \in V$ , such that with as few neighbors as possible, the following holds:

$$\begin{aligned} \sum_{j \in \mathcal{E}_i} C_{ij} &\leq U_i && \text{for all } i \\ \sum_{i \in \mathcal{U}_j} C_{ij} &\geq R_j && \text{for all } j \end{aligned} \quad (9)$$

where  $\mathcal{U}_j$  denotes the set of all the upstream neighbors of peer  $j$ . Such a rate-allocation scheme is in general not difficult to complete: It can be found by solving a linear optimization problem. Wu and Li [27] have proposed a fully distributed rate-allocation algorithm to solve a similar linear program. However, a potential limitation of this approach is the following: such a rate-allocation scheme may only guarantee the capacity for the last hop. There may be another cut with smaller capacity, which still constrains the overall streaming rate of the system. To the best of our knowledge, we are not aware of an existing result that can rigorously prove or disprove that guaranteeing the last-cut capacity is sufficient for guaranteeing the end-to-end streaming rate with high probability in a random topology. On the other hand, if we were to formulate the rate-allocation problem as another linear program for the minimum cut, the complexity would be much higher than (9). Hence, it remains a challenging question to develop low-complexity rate-allocation algorithms that can provably outperform the uniform rate-allocation scheme.

Recall that in Section II, using uniform rate allocation among the downstream neighbors, we show that all the other cuts have a much higher probability (than the last-hop cut) for achieving a rate larger than the required streaming rate. A natural question is then whether we can design a scheme that combines the advantages of both the more sophisticated rate allocation in (9) for improving the last cut and the uniform rate allocation for maintaining the high values at other cuts. This question leads us to the following hybrid algorithm that is simple to implement and provably reduces the number of neighbors required.

We consider the following class of hybrid algorithms  $\pi_\theta$  for rate allocation: Each peer reserves a fraction  $\theta \in (0, 1)$  of its upload capacity for the more sophisticated rate allocation similar to (9) and uses the remaining  $(1 - \theta)$  fraction of its upload capacity for uniform rate allocation. Specifically, let  $C_{ij}^S$  be the allocated capacity to  $j$  from  $i$ 's  $\theta$  fraction of upload capacity using the more sophisticated rate-allocation scheme, and let  $C_{ij}^U$  be the uniformly allocated capacity to peer  $j$  from peer  $i$ 's remaining  $(1 - \theta)$  fraction of upload capacity. Note that each peer still randomly selects  $M$  downstream neighbors. Hence,  $C_{ij}^U = \frac{(1-\theta)U_i}{M}$  if  $j \in \mathcal{E}_i$ . Then, the total allocated capacity from  $i$  to  $j$  is  $C_{ij} = C_{ij}^U + C_{ij}^S = \frac{(1-\theta)U_i}{M} + C_{ij}^S$ . We now formulate a linear feasibility problem to control  $C_{ij}^S$ . As we did before, we wish our algorithm could achieve a close-to-optimal streaming capacity. Hence, we set the target streaming rate of each user  $j$  to be  $R_j = (1 - \epsilon)C_f$ . Recall that  $C_f$  is the optimal streaming capacity. Therefore, the goal of the more sophisticated rate-allocation algorithm is to find  $C_{ij}^S$ 's such that

$$\begin{aligned} \sum_{j \in \mathcal{E}_i} C_{ij}^S &\leq \theta U_i, && \text{for all } i \\ \sum_{i \in \mathcal{U}_j} (C_{ij}^U + C_{ij}^S) &\geq (1 - \epsilon)C_f, && \text{for all } j. \end{aligned} \quad (10)$$

Note that as long as a feasible solution to (10) exists, a sufficient condition to (10) can be produced by the following modified optimization problem:

$$\begin{aligned} & \max \quad r \\ & \text{subject to} \quad \sum_{j \in \mathcal{E}_i} C_{ij}^S \leq \theta U_i, \quad \text{for all } i \\ & \quad \quad \quad \sum_{i \in \mathcal{U}_j} (C_{ij}^U + C_{ij}^S) \geq r, \quad \text{for all } j. \quad (11) \end{aligned}$$

Like the case with uniform rate allocation, in (11) we do not even need to know the optimal streaming rate  $C_f$  beforehand. Hence, the control of the peers (such as peer selection and rate allocation) is decoupled from the problem of choosing the streaming rate.<sup>1</sup> The distributed algorithm proposed in [27] is still suitable for solving this problem. Therefore, this hybrid algorithm still preserves the feature of being fully distributed and simple to implement. Next, we will show that it can achieve a close-to-optimal streaming capacity with a significantly lower number of neighbors.

#### A. Performance Analysis

Next, we will show that this hybrid algorithm can achieve a streaming capacity of  $(1 - \epsilon)C_f$  with a much smaller number of downstream neighbors of each peer. The following theorem states the performance of this hybrid algorithm more clearly.

*Theorem 8:* For any  $\epsilon \in (0, 1)$ ,  $\theta < 1/2$  and  $d > 1$ , there exist

$$\alpha \geq \max \left\{ \frac{(2d)u_s}{pu \max \left\{ \frac{\epsilon^2}{2}, \frac{(2\theta-1)^2}{8\theta^2} \right\}}, \frac{2 + \frac{p+\epsilon}{\theta} + d}{\left[ p - \frac{(p+\epsilon)\delta}{\theta} \right] (1 - \epsilon)} \right\}$$

and  $N_0$  such that for any  $N > N_0$  and  $M = \alpha \log(N)$ , the probability that for the capacity of the min-min-cut under the algorithm  $\pi_\theta$  is smaller than  $(1 - \epsilon)C_f$  is bounded by

$$\mathbf{P}(C_{\min - \min}(s \rightarrow V) \leq (1 - \epsilon)C_f) \leq O\left(\frac{1}{N^d}\right).$$

This result shows that the hybrid algorithm indeed reduces the lower bound on the number of required neighbors of each peer. Note that for small  $\epsilon$ , the factor  $\alpha$  does not depend on  $\epsilon$  at all. In contrast, the factor  $\alpha$  for the uniform rate-allocation scheme must increase proportional to  $1/\epsilon^2$ . As a numerical example, suppose that we want to sustain at least 90% of the optimal streaming capacity, which means that  $\epsilon = 0.1$ . The uniform rate-allocation scheme requires  $\alpha \geq \frac{400du_s}{up}$ . In contrast, if we use the hybrid algorithm  $\pi_\theta$  and choose  $\theta = 0.3$ , then we only need  $\alpha \geq \frac{10du_s}{up}$ . The number of neighbors of each peers is reduced by 40 times.

We separate the proof of Theorem 8 into two parts. First, since the allocation of  $C_{ij}^S$  is based on (10), we need to show that, given the uniform rate allocation of  $C_{ij}^U = \frac{(1-\theta)U_i}{M}$ , there exists a feasible solution to (10) with high probability. Hence, all last cuts should be able to exceed the required streaming rate with high probability. Second, we need to show that, based on the

<sup>1</sup>We note that this decoupling property may also be exploited to help the server to find the optimal streaming rate. For example, the server can use a simple probing mechanism to estimate the largest possible streaming rate based on the peers' feedback.

uniform rate allocation  $C_{ij}^S$  alone, the values of all other cuts should also exceed the required streaming rate with high probability. Theorem 8 would then follow.

For the first step, we will use the following results, which state an equivalent characterization to (9) and (10). Specifically, there exists a rate allocation such that the sum of the upload capacity allocated to each user from its immediate upstream neighbors is larger than its required streaming rate  $R_j$  if and only if, for any group of peers in the network, the total upload capacity from their upstream neighbors is larger than the sum of the streaming rates of this group of users.

*Lemma 9:* There exist  $C_{ij} \geq 0$ ,  $i, j \in V$ , such that (9) holds if and only if for any subset  $S \subseteq V$ , the following holds:

$$\sum_{i \in \mathcal{U}(S)} U_i \geq \sum_{j \in S} R_j \quad (12)$$

where  $\mathcal{U}(S) = \cup_{j \in S} \mathcal{U}_j$ .

*Corollary 10:* There exist  $C_{ij}^S \geq 0$ ,  $i, j \in V$  such that (10) holds if and only if for any subset  $S \subseteq V$ , the following holds:

$$\sum_{i \in \mathcal{U}(S)} \theta U_i \geq \sum_{j \in S} \left( (1 - \epsilon)C_f - \sum_{i \in \mathcal{U}_j} C_{ij}^U \right) \quad (13)$$

where  $C_{ij}^U = \frac{(1-\theta)U_i}{M}$ .

The proof of Lemma 9 follows a similar line of the argument as the Hall's Theorem [28]. The complete proof using the min-cut max-flow theorem is provided in [20]. Note that for the hybrid schemes, the reserved upload capacity of each user for the more sophisticated rate allocation is  $\theta U_i$ . In addition, each user receives a capacity of  $\sum_{i \in \mathcal{U}_j} C_{ij}^U$  from the uniform rate allocation. Thus, since the required streaming rate for each user  $j$  is  $(1 - \epsilon)C_f$ , the target downloading rate for the more sophisticated rate allocation should be  $(1 - \epsilon)C_f - \sum_{i \in \mathcal{U}_j} C_{ij}^U$ . Therefore, Corollary 10 follows from Lemma 9 immediately by letting the upload capacity of each user in Lemma 9 be  $\theta U_i$ , and letting  $R_j$  in Lemma 9 be  $(1 - \epsilon)C_f - \sum_{i \in \mathcal{U}_j} C_{ij}^U$ . Corollary 10 states that if (13) holds, then we can find a proper hybrid rate-allocation scheme such that the capacity of the last hop of each user is enough for its streaming rate. Next, we will show that (13) holds with high probability.

*Lemma 11:* Fix  $\theta \in (0, 1)$ . For any  $\epsilon \in (0, 1)$  and  $d > 1$ , there exist  $N_0$  and  $\alpha_0$  such that if  $N \geq N_0$  and

$$\alpha \geq \frac{2 + (p + \epsilon)/\theta + d}{[p - (p + \epsilon)\delta/\theta](1 - \epsilon)} \quad (14)$$

the following holds for the hybrid algorithm  $\pi_\theta$ :

$$\mathbf{P}\left(\theta \sum_{i \in \mathcal{U}(S)} U_i \leq \sum_{j \in S} \left[ (1 - \epsilon)C_f - \sum_{i \in V} C_{ij}^U \right] \text{ for some } S \subset V\right) \leq O\left(\frac{1}{N^d}\right).$$

Lemma 11 and Corollary 10 together imply that the probability with which (10) has no solution converges to 0 as the network size  $N$  grows. Therefore, with high probability, we can find a rate allocation such that (10) holds, i.e., the capacities of all last-hop cuts are greater than  $(1 - \epsilon)C_f$  with high probability. For others cuts, our random graph approach in Section II still applies (it is here that we need  $\theta < 0.5$ ). Theorem 8 then follows. Readers can refer to [20] for the detailed proof.



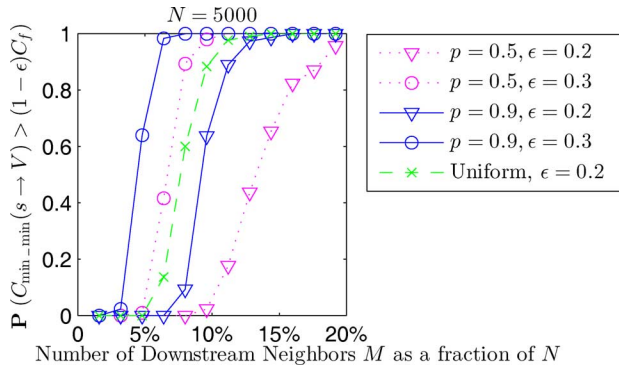


Fig. 2. Success probability versus the number of downstream neighbors  $M$  under uniform rate allocation.

#### IV. SIMULATION

In this section, we provide simulation results to verify the analytical results in previous sections. We simulate a P2P network with  $N = 5000$  peers and one server. Although the analytical results in this paper focus on the ON-OFF model for peers' upload capacity, here we provide simulation results both for the ON-OFF model and a uniform distribution model. In the ON-OFF model, each user has an ON probability of  $p$ . When a user is ON, it contributes an upload capacity  $u = 10$ . On the other hand, in the uniform distribution model, the upload capacity of each peer is uniformly distributed between  $[0, 10]$ . Furthermore, each peer chooses the same number of downstream neighbors and divides its upload capacity evenly among these neighbors, regardless of its upload capacity. In both cases, the server has a capacity of  $u_s = 20$ . The optimal streaming capacity is thus  $C_f = 9.004$  for the ON-OFF model with  $p = 0.9$ , and  $C_f = 5.004$  both for the ON-OFF model with  $p = 0.5$  and for the uniform distribution model. We vary the number of downstream neighbors of each user from 80 ( $\approx 9.4 \log N$ ) to 960 ( $\approx 113 \log N$ ), which correspond to 1.6% and 19.2% of the total number of peers  $N$ . For each choice of the number of downstream neighbors, we generate random networks for 200 times. For each iteration, all users select their downstream neighbors randomly as described in Section II-B, and we use the algorithm in [29] (a modified push-relabel algorithm) to find the min-min cut from the source to all the users and compare it to  $(1 - \epsilon)C_f$ . We count the number of times that the min-min cut of the network is larger than  $(1 - \epsilon)C_f$  and plot the probability for that to happen as the number of downstream neighbors of each peer varies. The result is shown in Fig. 2, where we simulate four different combinations of  $p$  (for the ON-OFF model) and  $\epsilon$ .

First, let us focus on the two curves marked with a triangle. They correspond to  $\epsilon = 0.2$ , i.e., the targeted streaming rate is 80% of  $C_f$ . We can observe that, using random peer-selection, when  $p = 0.5$  for the ON-OFF model and when the number of downstream neighbors of each peer is more than 960  $\approx 113 \log N$  (19.2% of  $N$ ), the success probability that the system could sustain a streaming rate higher than 80% of the optimal streaming capacity is greater than 0.9. If  $p = 0.9$  for the ON-OFF model, the number of downstream neighbors needed by each peer to achieve the same success probability of 0.9 reduces to 640  $\approx 75 \log N$  (12.8% of  $N$ ). Furthermore, we can observe that with the same ON probability  $p$ , when we increase

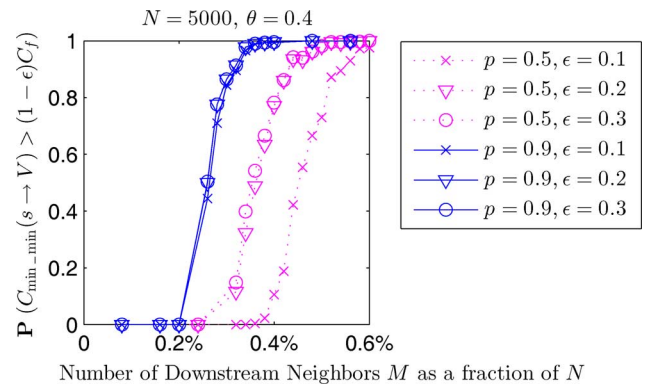


Fig. 3. Success probability versus the number of downstream neighbors  $M$  under hybrid rate allocation ( $\theta = 0.4$ ).

$\epsilon$  from  $\epsilon = 0.2$  to  $\epsilon = 0.3$ , the required number of downstream neighbors to achieve the same success probability of 0.9 decreases to 400  $\approx 47 \log N$  (for  $p = 0.5$ ) and 320  $\approx 38 \log N$  (for  $p = 0.9$ ). These observations verify our remarks following Theorem 1 that  $M$  needs to be larger if  $\epsilon$  is smaller or  $p$  is smaller. We also observe that, when the upload capacity of each peer follows the uniform distribution and when the number of downstream neighbors of each peer is more than 480  $\approx 56 \log N$  (9.6% of  $N$ ), the success probability of sustaining more than 80% of the optimal streaming capacity is almost 1. This suggests that our analytical result is still valid for other models of peer upload capacity.

We note that in the above simulation results, the number of neighbors required to achieve a high success probability is still quite large. Using a similar set of configurations, we next simulate the hybrid algorithm proposed in Section III, which is designed to further improve the performance. We first choose the parameter  $\theta$  to be 0.4 (i.e., each user performs the more sophisticated rate allocation with 40% of its upload capacity as described in Section III and allocates the remaining upload capacity uniformly among its downstream neighbors). The result is shown in Fig. 3. We notice that the number of neighbors required is reduced by an order of magnitude. For example, focus on the curve for  $p = 0.9$  and  $\epsilon = 0.2$ . In Fig. 3, when the number of downstream neighbors of each peer is more than 15 (0.3% of  $N$ ), the probability that the system can sustain a streaming rate higher than 80% of the optimal streaming capacity is already almost 1. In contrast, recall that for the corresponding curve in Fig. 2 with  $p = 0.9$  and  $\epsilon = 0.2$ , if we use uniform rate allocation, each peer needs more than 640  $\approx 75 \log N$  (12.8% of  $N$ ) downstream neighbors to achieve the same performance. Hence, the hybrid algorithm reduces the required number of downstream neighbors of each peer by more than 40 times, while still retaining the simplicity and robustness of the random peer-selection scheme.

In order to further understand how the value of  $\theta$  affects the performance of the hybrid algorithm, we conduct the following simulations. We vary the value of  $\theta$  from 0.1 to 1.0. For each  $\theta$ , we run the simulation in the same way as we generated Fig. 3. Note that although our analytical result in Theorem 8 requires that  $\theta < 0.5$ , here we experiment with an even larger range of  $\theta$ . In Fig. 4, we plot the success probability versus the number of downstream numbers for different values of  $\theta$ . The two subfigures correspond to two configurations of  $(p, \epsilon)$ . We observe that

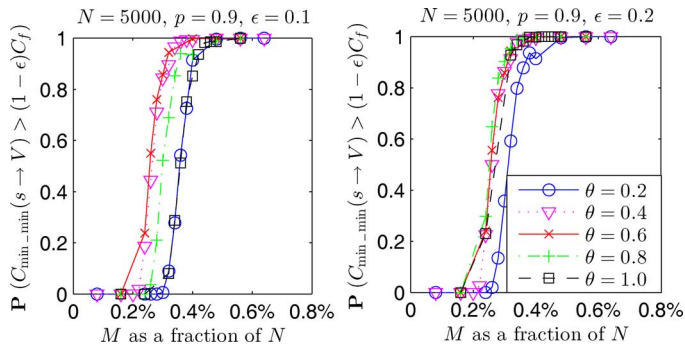


Fig. 4. Success probability versus the number of downstream neighbors  $M$  under hybrid rate allocation for different values of  $\theta$ .

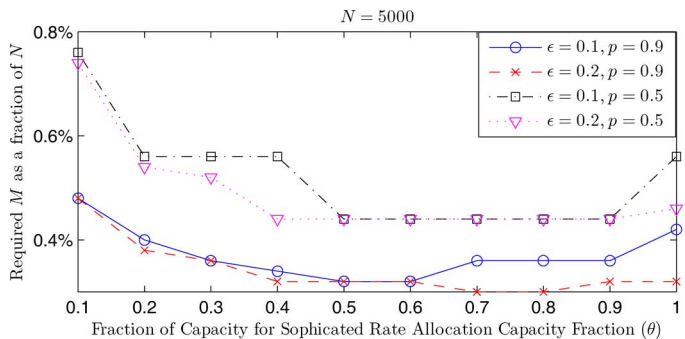


Fig. 5. Required number of downstream neighbors  $M$  versus the fraction of capacity for sophisticated rate allocation.

the performance of the hybrid algorithm is fairly insensitive to the value of  $\theta$  in the range  $[0.2, 1]$ .

To more clearly observe the trend, in Fig. 5 we plot, for each value of  $\theta$ , the smallest number of downstream neighbors  $M$  that is required for the system to reach a success probability of 0.9. Each of the four curves corresponds to a different combination of  $p$  and  $\epsilon$ . We recall that the point  $\theta = 0$  corresponds to uniform rate allocation. We can observe from Fig. 5 that when  $\theta$  is small (e.g.,  $\theta = 0.1$ ), the required number of downstream neighbors is significantly larger for all curves. As we explained toward the end of Section II-D, this behavior is due to the difficulty for uniform rate allocation to guarantee the capacities of the last-hop cuts. On the other hand, the point  $\theta = 1$  corresponds to the “pure” sophisticated rate allocation. For all curves, we observe that there is a large range of  $\theta$  where the required number of downstream neighbors is less than that required when  $\theta = 1$ . As we conjecture in Section III, this may have something to do with the difficulty for the “pure” sophisticated rate allocation to guarantee the capacities of cuts other than the last-hop cuts. Although in our simulations this performance degradation for  $\theta = 1$  does not appear to be very large, we are not aware of a theoretical result that can rigorously prove or disprove the performance of the “pure” sophisticated algorithm. From Fig. 5, we observe that a value of  $\theta$  between 0.3 to 0.5 appears to be a reasonable choice: it provides both theoretical performance guarantees (recall that Theorem 8 requires  $\theta < 0.5$ ) and good empirical performance.

We next simulate the performance of both the uniform rate-allocation algorithm and the hybrid rate-allocation algorithm when the total number of users  $N$  changes. We vary the total number of users in the systems from  $N = 100$  to  $N = 6400$ .

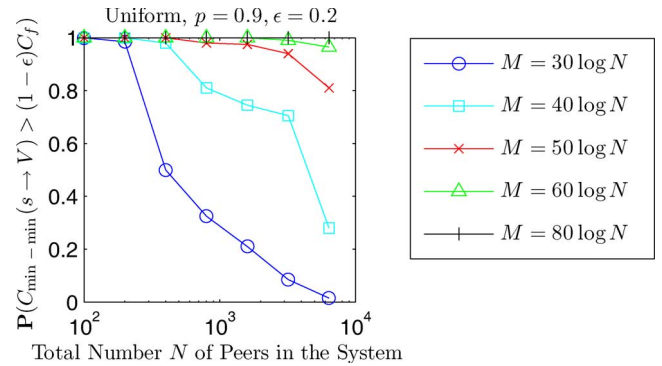


Fig. 6. Success probability versus  $N$  under uniform rate allocation.

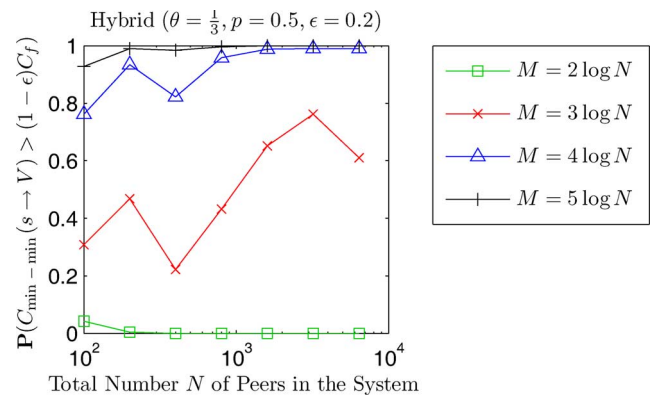


Fig. 7. Success probability versus  $N$  under hybrid rate allocation.

The results are shown in Figs. 6 and 7. For the results of the uniform rate-allocation algorithm in Fig. 6, we choose the parameters  $p = 0.9$  and  $\epsilon = 0.2$ . Each curve corresponds to a different choice of  $M$  from  $M = 30 \log N$  to  $M = 80 \log N$ . An interesting observation is that when  $M$  is small (e.g.,  $M = 30 \log N$ ), the performance in fact degrades as  $N$  increases. The reason is that when  $N$  is small,  $M$  may be even larger than  $N$ , in which case we use  $M = N$  and the network becomes fully connected. However, as  $N$  increases, the sparse connectivity and the negative effect of low  $M$  will eventually kick in. On the other hand, when  $M$  is sufficiently large ( $M = 80 \log N$ ), the success probabilities under all different values of  $N$  are always 1. For the results of the hybrid rate-allocation algorithm in Fig. 7, we choose the parameters  $p = 0.5$  and  $\epsilon = 0.2$ . Each curve corresponds to a different choice of  $M$  from  $M = 2 \log N$  to  $M = 5 \log N$ . We observe that the performance of the hybrid rate-allocation algorithm is less sensitive to the total number of users  $N$ . Under the same value of  $M$ , the success probability remains on the same level as  $N$  varies. On the other hand, we can still see that when  $M$  is sufficiently large, the success probability becomes 1 for all different values of  $N$ .

## V. CONCLUSION

In this paper, we study the streaming capacity of sparsely connected P2P networks. We show that even with a random peer-selection algorithm and uniform rate allocation, as long as each peer maintains  $\Omega(\log N)$  downstream neighbors, the system can achieve close-to-optimal streaming capacity with high probability when the network size is large. These results provide important new insights on the streaming capacity of large P2P

network with a sparse topology. One such insight is that the capacity of the last cut (i.e., the capacity from direct upstream neighbors) is often the bottleneck. We then use this insight to improve the peer-selection and rate-allocation algorithm to further optimize the achievable streaming capacity. Specifically, we design a hybrid algorithm that uses a slightly more sophisticated rate-allocation algorithm to improve the capacity and to reduce the constant factor in the  $\Omega(\log N)$  result. This new algorithm still retains the simplicity and robustness of the random peer-selection scheme, but it significantly reduces the number of neighbors required to achieve a certain performance guarantee.

Throughout this paper, we have assumed a uniformly random peer-selection scheme. It is highly likely that more sophisticated peer-selection schemes (albeit with a higher complexity) may lead to even better performance, e.g., an even smaller factor  $\alpha$ . For instance, one may assign a larger number of downstream neighbors to a peer with a larger upload capacity. However, we caution that the resulting performance improvement is not automatic. As we have seen in Section III for the hybrid algorithm, the effect of *local* improvement on the *global* performance can be difficult to quantify. Thus, the insights obtained from our analysis may be used to guide the design of more sophisticated algorithms. Furthermore, this paper has focused on P2P live-streaming systems. For future work, we will investigate whether similar insights can also be extended to P2P video-on-demand services, which have also become increasingly popular.

#### APPENDIX A PROOF OF LEMMA 6

*Proof:* By Chernoff bounds, we have for  $\theta > 0$

$$\begin{aligned} \mathbf{P}(D_m \leq (1-\epsilon)\mathbf{E}[D_m]|Y=y, t \text{ is ON}) \\ \leq \frac{\mathbf{E}[e^{-\theta D_m}|Y=y]}{e^{-(1-\epsilon)\theta\mathbf{E}[D_m]|Y=y}} = e^{\phi(\theta)+\phi_s(\theta)} \end{aligned} \quad (15)$$

where

$$\begin{aligned} \phi(\theta) &= \log \mathbf{E} \left[ e^{-\theta \sum_{j=1}^m \sum_{i=m+1}^{y+1} C_{ji}} \right] \\ &\quad + \theta(1-\epsilon)m(y-m+1)u/(N-1) \\ \phi_s(\theta) &= \log \mathbf{E} \left[ e^{-\theta \sum_{i=m+1}^{y+1} C_{si}} \right] \\ &\quad + \theta(1-\epsilon)(y-m+1)u_s/(y+1). \end{aligned}$$

Now we apply Proposition 5. Recall that we define a cut on  $H_t$  by dividing peers into sets  $W_m$  and  $W_m^c$ . We could also view  $W_m$  and  $W_m^c$  as subsets of some cut  $V_k$  and  $V_k^c$  of network  $G$ . We need to exclude the server from  $W_m$  since it has a different upload capacity. For each peer in  $W_m \setminus s$ , it will choose  $M$  downstream neighbors randomly from the entire network. Hence,  $|\hat{V}| = V$ . According to Proposition 5, we have  $q = |W_m \setminus s| = m$ ,  $r = |W_m^c| = y - m + 1$  and  $|\hat{V}| = N$ . Therefore, using (5), we have

$$\begin{aligned} \phi(\theta) &\leq \log \left\{ \exp \left[ Mm \frac{y+1-m}{N-1} (e^{-\theta \frac{u}{M}} - 1) \right] \right\} \\ &\quad + \theta(1-\epsilon)(y+1-m) \frac{u}{N-1} \\ &= \frac{1}{N-1} [Mm (e^{-\theta \frac{u}{M}} - 1) + \theta(1-\epsilon)u] (y+1-m). \end{aligned}$$

Note that the server only choose neighbors from the  $y+1$  ON peers,  $|\hat{V}| = y+1$ . Using similar techniques, for the server, we can bound  $\phi_s(\theta)$  by

$$\phi_s(\theta) \leq \frac{1}{y+1} \left[ M (e^{-\theta \frac{u_s}{M}} - 1) + \theta(1-\epsilon)u_s \right] (y+1-m).$$

Define  $\tilde{\phi}(\theta) \triangleq M (e^{-\theta \frac{u}{M}} - 1) + \theta(1-\epsilon)u$ , and  $\tilde{\phi}_s(\theta) \triangleq M (e^{-\theta \frac{u_s}{M}} - 1) + \theta(1-\epsilon)u_s$ . The  $\phi(\cdot)$  and  $\phi_s(\cdot)$  can be written as

$$\begin{aligned} \phi(\theta) &\leq \frac{1}{N-1} \tilde{\phi}(\theta)m(y+1-m) \\ \phi_s(\theta) &\leq \frac{1}{y+1} \tilde{\phi}_s(\theta)(y+1-m). \end{aligned}$$

Let  $\tilde{\phi}_{\min}$  and  $\tilde{\phi}_{s,\min}$  be the minimum of  $\tilde{\phi}(\theta)$  and  $\tilde{\phi}_s(\theta)$ , respectively, over  $\theta > 0$ . It is easy to see  $\tilde{\phi}_{\min} = \tilde{\phi}_{s,\min} < 0$ . Also since  $\tilde{\phi}$  and  $\tilde{\phi}_s$  is convex on  $\theta > 0$ , these minima are attainable. Let  $\theta_{\min}$  and  $\theta_{s,\min}$  be the minimizer, respectively. We must have

$$\tilde{\phi}_s(\theta_{s,\min}) = \tilde{\phi}_{s,\min} = \tilde{\phi}_{\min} \leq \tilde{\phi}(\theta_{s,\min}). \quad (16)$$

One can show that  $\theta_{s,\min} = -\frac{M}{u_s} \log(1-\epsilon)$ . Note that for  $0 < a < 1$  and  $0 \leq x \leq 1$ , we have  $(1-x)^a \leq 1-ax$  since  $(1-x)^a$  is concave and its derivative at 0 is  $-a$ . Moreover, for  $0 \leq x \leq 1$ , one can see that  $(1-x) \log(1-x) \geq x^2/2 - x$  by checking  $\frac{d}{dx}(1-x) \log(1-x) - (x^2/2 - x) = -\log(1-x) - x \geq 0$  and  $(1-x) \log(1-x) = x^2/2 - x$  when  $x = 0$ . Then, substituting  $\theta_{s,\min}$  into (16) and using the above relationship

$$\begin{aligned} \tilde{\phi}_s(\theta_{s,\min}) &\leq \tilde{\phi}(\theta_{s,\min}) \\ &= M \left[ (1-\epsilon) \frac{u}{u_s} - 1 \right] - M \frac{u}{u_s} (1-\epsilon) \log(1-\epsilon) \\ &\leq M \left[ 1 - \frac{u}{u_s} \epsilon - 1 - \frac{u}{u_s} \left( \frac{\epsilon^2}{2} - \epsilon \right) \right] = -M \frac{u}{u_s} \frac{\epsilon^2}{2}. \end{aligned}$$

Consequently

$$\begin{aligned} m\phi(\theta_{s,\min}) + \phi_s(\theta_{s,\min}) \\ \leq \tilde{\phi}(\theta_{s,\min})m(y+1-m)/(N-1) \\ \quad + \tilde{\phi}_s(\theta_{s,\min})(y+1-m)/(y+1) \\ \leq - \left( m \frac{y+1-m}{N-1} + \frac{y+1-m}{y+1} \right) M \frac{u}{u_s} \frac{\epsilon^2}{2}. \end{aligned}$$

Since (15) holds for any  $\theta > 0$ , letting  $\theta = \theta_{s,\min}$  yields

$$\begin{aligned} \mathbf{P}(D_m \leq (1-\epsilon)\mathbf{E}[D_m]|Y=y, t \text{ is ON}) \\ \leq \exp(m\phi(\theta_{s,\min}) + \phi_s(\theta_{s,\min})) \\ \leq \exp \left[ - \left( Mm \frac{y+1-m}{N-1} + M \frac{y+1-m}{y+1} \right) \frac{u}{u_s} \frac{\epsilon^2}{2} \right]. \end{aligned}$$

Similarly, one can show that if  $t$  is ON, we have

$$\begin{aligned} \mathbf{P}(D_m \leq (1-\epsilon)\mathbf{E}[D_m]|Y=y, t \text{ is OFF}) \\ \leq \exp \left[ - \left( Mm \frac{y+1-m}{N-1} + M \frac{y-m}{y} \right) \frac{u}{u_s} \frac{\epsilon^2}{2} \right]. \end{aligned}$$

Since  $\frac{y+1-m}{y+1} \geq \frac{y-m}{y}$ , we have

$$\begin{aligned} \mathbf{P}(D_m \leq (1-\epsilon)\mathbf{E}[D_m]|Y=y, t \text{ is ON}) \\ \leq \mathbf{P}(D_m \leq (1-\epsilon)\mathbf{E}[D_m]|Y=y, t \text{ is OFF}). \end{aligned}$$

Hence

$$\begin{aligned} & \mathbf{P}(D_m \leq (1 - \epsilon)\mathbf{E}[D_m]|Y = y) \\ & \leq \mathbf{P}(D_m \leq (1 - \epsilon)\mathbf{E}[D_m]|Y = y, t \text{ is OFF}) \\ & \leq \exp\left[-\left(Mm\frac{y+1-m}{N-1} + M\frac{y-m}{y}\right)\frac{u}{u_s}\frac{\epsilon^2}{2}\right]. \end{aligned}$$

□

## APPENDIX B PROOF OF LEMMA 11

*Proof:* Let  $Y$  be the number of ON users in the system, which is a random variable with binomial distribution  $\text{Bin}(N, p)$ . For any subset  $\mathcal{S}$  of  $V$ , define  $T_{\mathcal{S}}$  as the number of ON peers that are the upstream neighbors of at least one peer in  $\mathcal{S}$ , i.e.,  $T_{\mathcal{S}} = |\{i \in \mathcal{U}(\mathcal{S}) | i \text{ is ON}\}|$ . Let  $S = |\mathcal{S}|$  be the number of peers in  $\mathcal{S}$ . Then,  $\sum_{i \in \mathcal{U}(\mathcal{S})} U_i = uT_{\mathcal{S}}$ , and the following two events are equivalent (defined as  $\Gamma_{\mathcal{S}}$ ):

$$\begin{aligned} \Gamma_{\mathcal{S}} & \triangleq \left\{ \theta \sum_{i \in \mathcal{U}(\mathcal{S})} U_i \leq \sum_{j \in \mathcal{S}} \left[ (1 - \epsilon)C_f - \sum_{i \in V} C_{ij}^U \right] \right\} \\ & = \left\{ \theta u T_{\mathcal{S}} + \sum_{i \in V} \sum_{j \in \mathcal{S}} C_{ij}^U \leq S(1 - \epsilon)C_f \right\}. \end{aligned} \quad (17)$$

In the last event of (17), the first term of the left-hand side is the capacity from the more sophisticated allocation, and the second term is the capacity from uniform allocation. We divide the proof into two parts according to the value of  $S$ .

- 1) We first consider the case when  $S$  is small, i.e.,  $S \leq \delta N$ , where  $\delta \in (0, \theta/2)$  is a small constant that does not depend on  $N$ . We will show that when  $s$  is very small, the capacity of the more sophisticated allocation  $\theta u T_{\mathcal{S}}$  alone will be sufficient with high probability, i.e., it may be larger than  $S(1 - \epsilon)C_f$ . Recall that  $C_f = \min\{up + \frac{u_s}{N}, u_s\} \leq u(p + \frac{u_s}{uN})$ . Let  $p' = p + \frac{u_s}{uN}$ . Then, for any  $\epsilon > 0$ , there exists  $N_0$  such that whenever  $N > N_0$ ,  $p' < p + \epsilon$ . We thus have  $\theta u T_{\mathcal{S}} < S(1 - \epsilon)C_f$  implies  $T_{\mathcal{S}} < (1 - \epsilon)Sp'/\theta$ . Therefore

$$\begin{aligned} \mathbf{P}(\Gamma_{\mathcal{S}}) & \leq \mathbf{P}(\theta u T_{\mathcal{S}} \leq S(1 - \epsilon)C_f) \\ & \leq \mathbf{P}(T_{\mathcal{S}} < (1 - \epsilon)Sp'/\theta). \end{aligned} \quad (18)$$

Next, we are going to show that the probability that  $T_{\mathcal{S}} < (1 - \epsilon)Sp'/\theta$  for some  $\mathcal{S} \subset V$  is very small. To prove this, we first make the following claim: If there exists a set of peers  $\mathcal{S}$  such that  $T_{\mathcal{S}} < (1 - \epsilon)Sp'/\theta$ , then there exists another set of peers  $\mathcal{S}'$  such that

$$T_{\mathcal{S}'} \in \mathcal{I}_{\mathcal{S}'}(\epsilon, p') \triangleq \left[ \frac{(1 - \epsilon)(S' - 1)p'}{\theta}, \frac{(1 - \epsilon)S'p'}{\theta} \right] \quad (19)$$

where  $S' = |\mathcal{S}'|$ . To see this, first note that if  $S = 1$  and  $T_{\mathcal{S}} < (1 - \epsilon)Sp'/\theta$ , (19) automatically holds by letting  $\mathcal{S}' = \mathcal{S}$ . Suppose that  $T_{\mathcal{S}} < (1 - \epsilon)Sp'/\theta$  for some  $S > 1$ , but  $T_{\mathcal{S}} < (1 - \epsilon)(S - 1)p'/\theta$ . We then remove one peer from  $\mathcal{S}$  and obtain  $\mathcal{S}'$ . Clearly,  $S' = |\mathcal{S}'| = S - 1$ . We will have

$$T_{\mathcal{S}'} \leq T_{\mathcal{S}} < (1 - \epsilon)(S - 1)p'/\theta = (1 - \epsilon)S'p'/\theta.$$

Hence,  $\mathcal{S}'$  still satisfies  $T_{\mathcal{S}'} < \frac{(1 - \epsilon)S'p'}{\theta}$ . If (19) is still not true for  $\mathcal{S}'$ , we can remove another node from  $\mathcal{S}'$  and repeat these steps until we find a set that satisfies (19). Note that by removing nodes one by one from  $\mathcal{S}$ , in the worst case we will end up with a set  $\mathcal{S}'$  that contains one peer. However, as mentioned above, if  $S' = |\mathcal{S}'| = 1$ , (19) is automatically satisfied. As a result, we can always find a set  $\mathcal{S}'$  that satisfies (19) by removing the nodes from  $\mathcal{S}$  one by one. Therefore, the claim holds. Consequently

$$\begin{aligned} \mathbf{P}\left(T_{\mathcal{S}} < \frac{(1 - \epsilon)Sp'}{\theta} \text{ for some } \mathcal{S}\right) \\ \leq \mathbf{P}(T_{\mathcal{S}} \in \mathcal{I}_{\mathcal{S}}(\epsilon, p') \text{ for some } \mathcal{S}). \end{aligned} \quad (20)$$

Now we are going to characterize the probability on the right-hand side of (20). Define  $r_i(\mathcal{S})$  to be the probability that a given user  $i$  selects at least one of the peers in  $\mathcal{S}$  as its downstream neighbor. For any peer  $i \in \mathcal{S}$ ,  $r_i(\mathcal{S})$  is equal to 1 minus the probability that peer  $i$  chooses all its  $M$  downstream neighbors from the peers that are not in  $\mathcal{S}$ . More specifically, for  $i \in \mathcal{S}$ , we have

$$r_i(\mathcal{S}) = \mathbf{P}(i \in \mathcal{U}(\mathcal{S})) = 1 - \binom{N - S}{M} / \binom{N - 1}{M}. \quad (21)$$

Similarly, for  $i \in V \setminus \mathcal{S}$ , we have

$$r_i(\mathcal{S}) = \mathbf{P}(i \in \mathcal{U}(\mathcal{S})) = 1 - \binom{N - S - 1}{M} / \binom{N - 1}{M}.$$

Note that for any peer  $i$ , the value of  $r_i(\mathcal{S})$  is identical for all the sets  $\mathcal{S}$  that have the same size  $|\mathcal{S}|$ . In the rest of the proof, we will use  $r_i(\mathcal{S})$  to denote the probability that user  $i$  selects at least one of the peers in  $\mathcal{S}$  as its downstream neighbor for all the sets  $\mathcal{S}$  that satisfies  $|\mathcal{S}| = S$ , i.e.,  $r_i(\mathcal{S}) = r_i(\mathcal{S}), \forall \mathcal{S} \subset V$  such that  $|\mathcal{S}| = S$ . Note that

$$1 - \binom{N - S}{M} / \binom{N - 1}{M} \leq 1 - \binom{N - S - 1}{M} / \binom{N - 1}{M}.$$

Thus, for any  $i \in V$ , (21) becomes a lower bound of  $r_i(\mathcal{S})$

$$r_i(\mathcal{S}) \geq 1 - \binom{N - S}{M} / \binom{N - 1}{M}. \quad (22)$$

The second term on the right-hand side of (22) satisfies

$$\frac{\binom{N - S}{M}}{\binom{N - 1}{M}} = \frac{(N - S)!}{M!(N - S - M)!} \leq \left(1 - \frac{M}{N}\right)^S \leq e^{-\frac{SM}{N}}.$$

Combining (22) and the above inequality, we get a uniform lower bound of  $r_i(\mathcal{S})$  for all  $i$ , which is denoted by  $r(S)$ ,  $r_i(\mathcal{S}) \geq 1 - e^{-\frac{SM}{N}} \triangleq r(S)$ . Now we have, for  $y \geq (1 - \epsilon)Np$

$$\begin{aligned} & \mathbf{P}(T_{\mathcal{S}} \in \mathcal{I}_{\mathcal{S}}(\epsilon, p') | Y = y) \\ & \leq \sum_{t=\lceil (1 - \epsilon)(S - 1)p'/\theta \rceil}^{\lfloor (1 - \epsilon)Sp'/\theta \rfloor} \binom{y}{t} r(S)^t (1 - r(S))^{y - t} \\ & \leq \frac{1}{\theta} \left( \binom{y}{\lfloor (1 - \epsilon)Sp'/\theta \rfloor} \right) r(S)^{\lfloor (1 - \epsilon)\frac{Sp'}{\theta} \rfloor} (1 - r(S))^{y - \lfloor (1 - \epsilon)\frac{Sp'}{\theta} \rfloor} \\ & \leq \frac{1}{\theta} N^{\frac{Sp'}{\theta}} e^{-\frac{SM}{N}(1 - \epsilon)} (Np - \frac{Sp'}{\theta}). \end{aligned}$$

Then, for  $y \geq (1 - \epsilon)Np$ , we have

$$\begin{aligned}
 & \mathbf{P}\left(T_S \in \mathcal{I}_S(\epsilon, p') \text{ for some } S \leq \delta N \mid Y = y\right) \\
 & \leq \sum_{S=1}^{\delta N} \binom{N}{S} \mathbf{P}(T_S \in \mathcal{I}_S(\epsilon, p') \mid Y = y) \\
 & \leq \sum_{S=1}^{\delta N} \frac{1}{\theta} N^S N^{\frac{Sp'}{\theta}} e^{-\frac{S}{N}M(1-\epsilon)(Np - \frac{Sp'}{\theta})} \\
 & \leq \sum_{S=1}^{\delta N} \frac{1}{\theta} N^{S(1+\frac{p'}{\theta})} N^{-\alpha S(1-\epsilon)(p - \frac{p'\delta}{\theta})} \\
 & \quad \text{(since } M = \alpha \log N\text{)}.
 \end{aligned}$$

It follows that

$$\begin{aligned}
 & \mathbf{P}(T_S \in \mathcal{I}_S(\epsilon, p') \text{ for some } S \leq \delta N) \\
 & \leq \mathbf{P}(Y < (1 - \epsilon)Np) + \sum_{[y=(1-\epsilon)Np]}^N \mathbf{P}(Y = y) \\
 & \quad \times \mathbf{P}\left(T_S \in \mathcal{I}_S(\epsilon, p') \text{ for some } S \leq \delta N \mid Y = y\right) \\
 & \leq O(\exp(-\epsilon^2 p^2 N)) + \sum_{S=1}^{\delta N} \frac{1}{\theta} N^{S(1+p'/\theta)} N^{-\alpha S(1-\epsilon)(p - p'\delta/\theta)}.
 \end{aligned} \tag{23}$$

Note that when  $N$  is large,  $\alpha$  satisfies

$$\alpha > \frac{2 + (p + \epsilon)/\theta + d}{[p - (p + \epsilon)\delta/\theta](1 - \epsilon)} \geq \frac{2 + p'/\theta + d}{[p - p'\delta/\theta](1 - \epsilon)}.$$

We have

$$\begin{aligned}
 & \sum_{S=1}^{\delta N} \frac{1}{\theta} N^{S(1+p'/\theta)} N^{-\alpha S(1-\epsilon)(p - p'\delta/\theta)p} \\
 & \leq \frac{1}{\theta} N^{1+1+p'/\theta} N^{-(2+p'/\theta+d)} = \frac{1}{\theta N^d} = O\left(\frac{1}{N^d}\right).
 \end{aligned} \tag{24}$$

Finally, combining (18), (20), (23), and (24), we have

$$\begin{aligned}
 & \mathbf{P}\left(\bigcup_{S \subset V, |S| \leq \delta N} \Gamma_S\right) \\
 & \leq \mathbf{P}\left(T_S < \frac{(1 - \epsilon)sp'}{\theta} \text{ for some } S \leq \delta N\right) \\
 & \leq \mathbf{P}(T_S \in \mathcal{I}_S(\epsilon, p') \text{ for some } S \leq \delta N) \leq O\left(\frac{1}{N^d}\right).
 \end{aligned}$$

2) When  $s$  is large, i.e.,  $S > \delta N$ , the capacity from sophisticated allocation alone may not be adequate. We then need to count both parts of the capacity in (17). Consider the quantity  $\theta u T_S + \sum_{i \in V} \sum_{j \in S} C_{ij}^U$  in (17). It can be viewed as the maximum capacity that can be assigned to  $S$  from both the more sophisticated and uniform rate allocation. Now consider a purely uniform rate allocation. The total capacity allocated to  $S$  must be a lower bound of the above value. Next, we will show that the above lower bound will be larger than  $(1 - \epsilon)SC_f$  with high probability.

More precisely, let  $I_{ij}$  be the indicator function of the event that there is a link between node  $i$  and node  $j$ , and node  $i$  is an ON peer or the server. Then, we have

$$\sum_{i \in V} \sum_{j \in S} C_{ij}^U = \frac{u_s}{M} \sum_{j \in S} I_{sj} + \frac{(1 - \theta)u}{M} \sum_{i \in V \setminus s} \sum_{j \in S} I_{ij}.$$

Note that for fixed  $i \in V$ ,  $\sum_{j \in S} I_{ij} \leq M$ . Furthermore, if  $i$  is ON or  $i \notin \mathcal{U}(S)$ , then  $\sum_{j \in S} I_{ij} = 0$ . Recall that  $T_S$  is the number of ON users in  $\mathcal{U}(S)$ . We have

$$\sum_{i \in V \setminus s} \sum_{j \in S} I_{ij} = \sum_{i \in \mathcal{U}(S), i \text{ is ON}} \sum_{j \in S} I_{ij} \leq T_S M$$

and hence  $T_S \geq \frac{1}{M} \sum_{i \in V \setminus s} \sum_{j \in S} I_{ij}$ . Then, the total available capacity from  $\mathcal{U}(S)$  to  $S$  will be

$$\begin{aligned}
 & \theta u T_S + \sum_{i \in V} \sum_{j \in S} C_{ij}^U \\
 & \geq \frac{u_s}{M} \sum_{j \in S} I_{sj} + \theta u \frac{1}{M} \sum_{i \in V \setminus s} \sum_{j \in S} I_{ij} + \frac{(1 - \theta)u}{M} \sum_{i \in V} \sum_{j \in S} I_{ij} \\
 & = \frac{u_s}{M} \sum_{j \in S} I_{sj} + \frac{u}{M} \sum_{i \in V \setminus s} \sum_{j \in S} I_{ij}.
 \end{aligned}$$

The above value is equal to the capacity from  $\mathcal{U}(S)$  to  $S$  if we use purely uniform rate allocation scheme. Note that

$$\begin{aligned}
 & \mathbf{E}\left[\frac{u_s}{M} \sum_{j \in S} I_{sj} + \frac{u}{M} \sum_{i \in V \setminus s} \sum_{j \in S} I_{ij}\right] \\
 & = N \frac{M}{N} \cdot \frac{u_s}{M} + N S \frac{M}{N} \cdot \frac{u}{M} p \geq C_f.
 \end{aligned}$$

Applying Chernoff bound and Lemma 5, and using similar argument as we did when proving Lemma 6, we can show that

$$\begin{aligned}
 & \mathbf{P}(\Gamma_S) = \mathbf{P}\left(\theta u T_S + \sum_{i \in V} \sum_{j \in S} C_{ij}^U \leq (1 - \epsilon)SC_f\right) \\
 & \leq \mathbf{P}\left(\frac{u_s}{M} \sum_{j \in S} I_{sj} + \frac{u}{M} \sum_{i \in V \setminus s} \sum_{j \in S} I_{ij} \leq (1 - \epsilon)SC_f\right) \\
 & \leq e^{-\frac{\epsilon^2}{2} \frac{u}{u_s} M S p}.
 \end{aligned}$$

Consequently

$$\begin{aligned}
 & \mathbf{P}\left(\bigcup_{S \subset V, |S| > \delta N} \Gamma_S\right) \\
 & \leq \sum_{S=\delta N+1}^N \binom{N}{S} e^{-\frac{\epsilon^2}{2} \frac{u}{u_s} M S p} \leq \sum_{S=\delta N+1}^N \left(\frac{Ne}{S}\right)^S e^{-\frac{\epsilon^2}{2} \frac{u}{u_s} M S p} \\
 & \leq \sum_{S=\delta N+1}^N e^{S(1 - \log \delta)} e^{-\frac{\epsilon^2}{2} \frac{u}{u_s} M S p} \leq 2e^{(1 - \log \delta - \frac{\epsilon^2}{2} \frac{u}{u_s} M p) \delta N}.
 \end{aligned}$$

Hence, as long as  $1 - \log \delta - \frac{\epsilon^2}{2} \frac{u}{u_s} M p < 0$ , the above expression will converge to 0 exponentially fast. In fact, if  $M = \alpha \log N$  and  $\alpha$  satisfies (14), then for sufficiently

large  $N$ , the inequality  $1 - \log \delta - \frac{\epsilon^2}{2} Mp < 0$  always holds. Hence, if (14) holds, we have

$$\mathbf{P} \left( \bigcup_{S \subset V, |S| > \delta N} \Gamma_S \right) \leq O(1/N^d).$$

Finally, by combining the result of parts 1) and 2) together, we can thus prove the lemma.  $\square$

## REFERENCES

- [1] C. Zhao, X. Lin, and C. Wu, "The streaming capacity of sparsely-connected P2P systems with distributed control," in *Proc. IEEE INFOCOM*, Apr. 2011, pp. 1449–1457.
- [2] X. Zhang, J. Liu, B. Li, and Y.-S. Yum, "Coolstreaming/donet: A data-driven overlay network for peer-to-peer live media streaming," in *Proc. IEEE INFOCOM*, 2005, vol. 3, pp. 2102–2111.
- [3] X. Hei, C. Liang, J. Liang, Y. Liu, and K. Ross, "A measurement study of a large-scale p2p iptv system," *IEEE Trans. Multimedia*, vol. 9, no. 8, pp. 1672–1687, Dec. 2007.
- [4] T. Silerston and O. Fourmaux, "Measuring P2P IPTV systems," in *Proc. NOSSDAV*, Jun. 2007.
- [5] C. Wu, B. Li, and S. Zhao, "Exploring large-scale peer-to-peer live streaming topologies," *Trans. Multimedia Comput. Commun. Appl.*, vol. 4, no. 3, pp. 1–23, 2008.
- [6] W. Liang, J. Bi, R. Wu, Z. Li, and C. Li, "On characterizing PP-Stream: Measurement and analysis of P2P IPTV under large-scale broadcasting," in *Proc. IEEE GLOBECOM*, Nov. 2009, pp. 1–6.
- [7] R. Kumar, Y. Liu, and K. Ross, "Stochastic fluid theory for P2P streaming systems," in *Proc. IEEE INFOCOM*, May 2007, pp. 919–927.
- [8] L. Massoulié and A. Twigg, "Rate-optimal schemes for peer-to-peer live streaming," *Perform. Eval.*, vol. 65, no. 11–12, pp. 804–822, 2008.
- [9] C. Feng and B. Li, "On large-scale peer-to-peer streaming systems with network coding," in *Proc. 16th ACM Int. Conf. Multimedia*, Vancouver, BC, Canada, 2008, pp. 269–278.
- [10] T. Bonald, L. Massoulié, F. Mathieu, D. Perino, and A. Twigg, "Epidemic live streaming: Optimal performance trade-offs," in *Proc. ACM SIGMETRICS*, 2008, pp. 325–336.
- [11] S. Liu, R. Zhang-Shen, W. Jiang, J. Rexford, and M. Chiang, "Performance bounds for peer-assisted live streaming," in *Proc. ACM SIGMETRICS*, 2008, pp. 313–324.
- [12] S. Sengupta *et al.*, "Peer-to-peer streaming capacity," *IEEE Trans. Inf. Theory*, vol. 57, no. 8, pp. 5072–5087, Aug. 2011.
- [13] S. Liu *et al.*, "P2P streaming capacity under node degree bound," in *Proc. IEEE ICDCS*, Genoa, Italy, Jun. 2010, pp. 587–598.
- [14] M. Castro *et al.*, "Splitstream: High-bandwidth multicast in cooperative environments," in *Proc. SOSP*, Bolton Landing, NY, USA, 2003, pp. 298–313.
- [15] V. Pai *et al.*, "Chainsaw: Eliminating trees from overlay multicast," in *Proc. IPTPS*, 2005, pp. 127–140.
- [16] P. Gupta and P. Kumar, "The capacity of wireless networks," *IEEE Trans. Inf. Theory*, vol. 46, no. 2, pp. 388–404, Mar. 2000.
- [17] D. Wu, C. Liang, Y. Liu, and K. Ross, "View-upload decoupling: A redesign of multi-channel P2P video systems," in *Proc. IEEE INFOCOM*, 2009, pp. 2726–2730.
- [18] L. Massoulié, A. Twigg, C. Gkantsidis, and P. Rodriguez, "Randomized decentralized broadcasting algorithms," in *Proc. IEEE INFOCOM*, 2007, pp. 1073–1081.
- [19] A. Ramamoorthy, J. Shi, and R. Wesel, "On the capacity of network coding for random networks," *IEEE Trans. Inf. Theory*, vol. 51, no. 8, pp. 2878–2885, Aug. 2005.
- [20] C. Zhao, X. Lin, and C. Wu, "The streaming capacity of sparsely-connected P2P systems with distributed control," Purdue University, West Lafayette, IN, USA, Tech. Rep., 2012 [Online]. Available: <https://engineering.purdue.edu/~elinx/papers.html>

- [21] J. Edmonds, "Edge-disjoint branchings," in *Combinatorial Algorithms*, R. Rustin, Ed. New York, NY, USA: Algorithmics Press, 1973, pp. 91–96.
- [22] R. Ahlswede, N. Cai, S.-Y. Li, and R. Yeung, "Network information flow," *IEEE Trans. Inf. Theory*, vol. 46, no. 4, pp. 1204–1216, Jul. 2000.
- [23] S.-Y. Li, R. Yeung, and N. Cai, "Linear network coding," *IEEE Trans. Inf. Theory*, vol. 49, no. 2, pp. 371–381, Feb. 2003.
- [24] S. Ross, *A First Course in Probability*, 5th ed. Upper Saddle River, NJ, USA: Prentice-Hall, 1998.
- [25] A. D. Barbour, L. Holst, and S. Janson, *Poisson Approximation*. Oxford, U.K.: Oxford Univ. Press, 1992.
- [26] S. Janson, "Large deviation inequalities for sums of indicator variables," Uppsala Universitet, Uppsala, Sweden, Tech. Rep., 1994.
- [27] C. Wu and B. Li, "Optimal rate allocation in overlay content distribution," in *Proc. 6th Internat. IFIP-TC6 Conf. Ad Hoc Sensor Netw., Wireless Netw., Next Generation Internet*, 2007, pp. 678–690.
- [28] J. Gross and J. Yellen, *Graph Theory and its Applications*. Boca Raton, FL, USA: CRC Press, 1999.
- [29] J. Hao and J. B. Orlin, "A faster algorithm for finding the minimum cut in a graph," in *Proc. 3rd Annu. ACM-SIAM SODA*, Orlando, FL, USA, 1992, pp. 165–174.



**Can Zhao** (S'09–A'13) received the B.S. degree in electrical engineering from Tsinghua University, Beijing, China, in 2007, and the Ph.D. degree in electrical and computer engineering from Purdue University, West Lafayette, IN, USA, in 2012.

His research interests are mathematical modeling and evaluation of communication networks, including ad hoc networks and peer-to-peer networks.



**Xiaojun Lin** (S'02–M'05–SM'12) received the B.S. degree in electronics and information systems from Zhongshan University, Guangzhou, China, in 1994, and the M.S. and Ph.D. degrees in electrical and computer engineering from Purdue University, West Lafayette, IN, USA, in 2000 and 2005, respectively.

He is currently an Associate Professor of electrical and computer engineering with Purdue University. He is currently serving as an Area Editor for *Computer Networks*, and has served as a Guest Editor for *Ad Hoc Networks*. His research interests are in

the analysis, control, and optimization of wireless and wireline communication networks.

Dr. Lin received the IEEE INFOCOM 2008 Best Paper Award and 2005 Best Paper of the Year Award from the *Journal of Communications and Networks*. His paper was also one of two runner-up papers for the Best Paper Award at IEEE INFOCOM 2005. He received the NSF CAREER Award in 2007.



**Chuan Wu** (S'06–M'08) received the B.E. and M.E. degrees in computer science and technology from Tsinghua University, Beijing, China, in 2000 and 2002, respectively, and the Ph.D. degree in electrical and computer engineering from the University of Toronto, Toronto, ON, Canada, in 2008.

She is currently an Assistant Professor with the Department of Computer Science, University of Hong Kong, Hong Kong. Her research interests include cloud computing, peer-to-peer networks, and online/mobile social network.

Dr. Wu is a member of the Association for Computing Machinery (ACM).