

Hindawi Publishing Corporation
International Journal of Digital Multimedia Broadcasting
Volume 2012, Article ID 807520, 9 pages
doi:10.1155/2012/807520

Research Article

Improving Streaming Capacity in Multi-Channel P2P VoD Systems via Intra-Channel and Cross-Channel Resource Allocation

Yifeng He and Ling Guan

Department of Electrical and Computer Engineering, Ryerson University, Toronto, ON, Canada M5B2K3

Correspondence should be addressed to Yifeng He, yhe@ee.ryerson.ca

Received 13 August 2011; Accepted 6 December 2011

Academic Editor: Ivan Lee

Copyright © 2012 Y. He and L. Guan. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Multi-channel Peer-to-Peer (P2P) Video-on-Demand (VoD) systems can be categorized into *independent-channel* P2P VoD systems and *correlated-channel* P2P VoD systems. Streaming capacity for a channel is defined as the maximal streaming rate that can be received by every user of the channel. In this paper, we study the streaming capacity problem in multi-channel P2P VoD systems. In an independent-channel P2P VoD system, there is no resource correlation among channels. Therefore, we can find the average streaming capacity for the independent-channel P2P VoD system by finding the streaming capacity for each individual channel, respectively. We propose a distributed algorithm to solve the streaming capacity problem for a single channel in an independent-channel P2P VoD system. The average streaming capacity for a correlated-channel P2P VoD system depends on both the intra-channel and cross-channel resource allocation. To better utilize the cross-channel resources, we first optimize the server upload allocation among channels to maximize the average streaming capacity and then propose cross-channel helpers to enable cross-channel sharing of peer upload bandwidths. We demonstrate in the simulations that the correlated-channel P2P VoD systems with both intra-channel and cross-channel resource allocation can obtain a higher average streaming capacity compared to the independent-channel P2P VoD systems with only intra-channel resource allocation.

1. Introduction

Video-on-demand (VoD) services have been attracting a lot of users because it allows users to watch any video at any time. Traditional client/server architectures for VoD services cannot provide video streams to a large number of concurrent users. To offload the server upload burden, Peer-to-Peer (P2P) technology has been integrated into VoD applications by utilizing the uplink bandwidths of the peers [1–5].

Most of the P2P VoD systems offer many video channels. Users can choose any of the channels that they are interested in at any time. The P2P VoD systems with multiple channels are called *multi-channel* P2P VoD systems. Depending on the resource correlation, multi-channel P2P VoD systems can be categorized into *independent-channel* P2P VoD systems and *correlated-channel* P2P VoD systems. In an independent-channel P2P VoD system, the peers watch the same channel form an independent overlay and share the resources with

each other exclusively within the overlay. In a correlated-channel P2P VoD system, overlay m formed by the peers watching channel m can be correlated with overlay k formed by the peers watching channel k , such that the peers in overlay m can share the resources not only with other peers in overlay m but also with the peers in overlay k . The resources in a correlated-channel P2P VoD system can be utilized in a better way compared to an independent-channel P2P VoD system.

In P2P VoD systems, users would like to watch the video at a high quality. The streaming rate can be used to indicate the video quality. Let \mathbf{M} denote the set of the channels in a P2P VoD system. If channel $m (m \in \mathbf{M})$ is associated with overlay m , *streaming capacity* for channel m , denoted by c_m , is defined as the maximum streaming rate that can be received by every user in overlay m [6, 7]. The *average streaming capacity* c_{avg} for a multi-channel P2P VoD system is defined as $c_{\text{avg}} = \sum_{m \in \mathbf{M}} p_m c_m$ where p_m is the priority of channel m ,

and $\sum_{m \in M} p_m = 1$. The priority of a channel can be set by the service provider. For example, the more expensive channel can be assigned a higher priority.

Streaming capacity problem in multi-channel P2P VoD systems is a challenging problem. The average streaming capacity in a multi-channel P2P VoD system is dependent on the number of the peers, the playback time, and the bandwidth of each peer, the server capacity, the overlay construction, and the resource allocation. Optimal resource allocation in a multi-channel P2P VoD system is expected to improve the streaming capacity. However, resource allocation in multi-channel P2P VoD systems is quite challenging due to the following reasons. (1) Peers have heterogeneous upload and download bandwidths and different playback progress. (2) Each channel is heterogeneous in terms of available resources, since the number of the peers in each channel is different. (3) Peers may leave or join a channel dynamically.

In this paper, we improve the average streaming capacity for multi-channel P2P VoD systems by better utilizing both the intra-channel resources and the cross-channel resources. We first investigate the streaming capacity for an independent-channel P2P VoD system, in which we find the streaming capacity for a single channel by optimizing the intra-channel resource allocation in a distributed manner. We then investigate the streaming capacity for a correlated-channel P2P VoD system, in which we find the average streaming capacity for multiple channels by optimizing both intra-channel and cross-channel resource allocation. The proposed cross-channel resource allocation consists of two steps as follows. (1) We optimize the server upload allocation among channels to maximize the average streaming capacity. (2) We introduce *cross-channel helpers* to establish cross-channel links and then utilize cross-channel peer upload bandwidths to improve the average streaming capacity.

The remainder of this paper is organized as follows. Section 2 discusses the related work. Section 3 formulates and solves the streaming capacity problem for an independent-channel P2P VoD System. Section 4 finds the average streaming capacity for a correlated-channel P2P VoD System. The simulation results are provided in Section 5, and the conclusions are drawn in Section 6.

2. Related Work

Streaming capacity in P2P live systems has been examined in the recent literature [6–9]. In [6], the streaming capacity problem is formulated into an optimization problem, which maximizes the streaming rate that can be supported by multitree-based overlay. Sengupta et al. provide a taxonomy of sixteen problem formulations on streaming capacity, depending on whether there is a single P2P session or there are multiple concurrent sessions, whether the given topology is a full mesh graph or an arbitrary graph, whether the number of peers a node can have is bounded or not, and whether there are nonreceiver relay nodes or not [7]. Liu et al. analyze the performance bounds for minimum server load, maximum streaming rate, and minimum tree depth in tree-based P2P live systems, respectively [8]. The streaming capacity under node degree bound is investigated in [9].

Streaming capacity for a single channel in P2P VoD systems has been studied in [10–12]. In [10, 12], the streaming capacity for a single channel is formulated into an optimization problem which maximizes the streaming rate under the peer bandwidth constraints. The throughput maximization problem in a scalable P2P VoD system is studied in [13]. Helpers have been proposed in P2P systems to improve the system performance [11, 14, 15]. In P2P VoD systems, each additional helper increases the system upload capacity, thus offloading the server burden [15]. In [11], the algorithms on helper assignment and rate allocation are proposed to improve the streaming capacity for P2P VoD systems.

Cross-channel resource sharing has been recently studied in multi-channel P2P streaming systems [16–20]. In [16], Wu et al. propose an online server capacity provisioning algorithm to adjust the server capacities available to each of the concurrent channels, taking into account the number of peers, the streaming quality, and the priorities of channels. In [17], a View-Upload Decoupling (VUD) scheme is proposed to decouple what a peer uploads from what it views, bringing stability to multi-channel P2P streaming systems and enabling cross-channel resource sharing. In [18], Wu et al. develop infinite-server queueing network models to analytically study the performance of multi-channel P2P live streaming systems. In [19], the bandwidth satisfaction ratio is used to compare three bandwidth allocation schemes, namely, Naive Bandwidth allocation Approach (NBA), Passive Channel-aware bandwidth allocation Approach (PCA), and Active Channel-aware bandwidth allocation Approach (ACA), in multi-channel P2P streaming systems. In [20], Zhao et al. investigate the streaming capacity in multi-channel P2P live streaming systems when each peer can only connect to a small number of neighbors.

3. Streaming Capacity for an Independent-Channel P2P VoD System

In an independent-channel P2P VoD system, the peers watching the same channel form an overlay. We assume that the server upload bandwidth allocated to channel m , denoted by s_m , is predetermined. An independent-channel P2P VoD system is illustrated in Figure 1. The peers within the same overlay redistribute the video content to each other. A peer belonging to overlay m only caches the video content of channel m , and it does not serve any video content to any peer outside overlay m . In other words, each overlay in an independent-channel P2P VoD System is an isolated subsystem. Therefore, the streaming capacity of a channel is independent of that of another channel. The streaming capacity problem in an independent-channel P2P VoD system is to find the streaming capacity of each individual channel, respectively. We will next focus on the streaming capacity for a single channel.

3.1. Complete Overlay. The streaming capacity for a single channel depends on the constructed overlay. In a P2P VoD system, each peer maintains a buffer to cache the recently received packets in order to smoothen the playback and serve

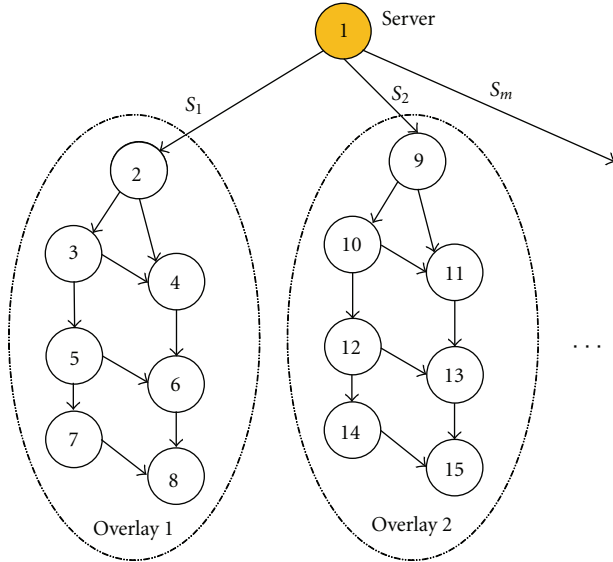


FIGURE 1: Illustration of an independent-channel P2P VoD system.

other peers. There is a *parent-child relationship* between two peers (e.g., peer k and peer j) if the two peers satisfy the following two conditions: (1) peer k (the parent) has an earlier playback progress than peer j (the child), and (2) peer k is buffering the segment(s) which are being requested by peer j . The parent-child relationship is illustrated in Figure 2(a). At the current moment, peer k is playing segment 11, and peer j is playing segment 7. Each peer can hold 5 segments in its buffer. Segment 8, held by peer k , is being requested by peer j . Therefore, peers k and j can establish a parent-child relationship, in which peer k is the parent, and peer j is the child. The server is regarded as a special peer because it contains all the segments of the video. The server has a parent-child relationship with any other peer.

A parent-child relationship is *implemented*, if an overlay link has been established from the parent to the child. A *complete overlay* for a channel is defined as the overlay in which all of the available parent-child relationships have been implemented, and an *incomplete overlay* for a channel is defined as the overlay in which only a part of the available parent-child relationships have been implemented [12]. An example of complete overlay is illustrated in Figure 2(b). Compared to the incomplete overlay, the complete overlay contains more overlay links, thus supporting a higher streaming capacity. Therefore, we will formulate the streaming capacity problem based on the complete overlay. The method for overlay construction is not a study focus of this paper. A complete overlay can be constructed using the existing approaches [2].

3.2. Streaming Capacity Problem for a Single Channel. The set of the overlays (channels) in the independent-channel P2P VoD system is denoted by \mathbf{M} . Since each channel is independent in terms of resources, we can just study the streaming capacity of channel m ($\forall m \in \mathbf{M}$).

Overlay m in a P2P VoD system can be modeled as a directed graph $\mathbf{G}^{(m)} = (\mathbf{N}^{(m)}, \mathbf{L}^{(m)})$, where $\mathbf{N}^{(m)}$ is the set of

nodes and $\mathbf{L}^{(m)}$ is the set of directed overlay links. Peer 1 is defined as the server. The relationship between a node and its outgoing links is represented with a matrix $\mathbf{A}^{(m)+}$, whose elements are given by

$$a_{il}^{(m)+} = \begin{cases} 1, & \text{if link } l \text{ is an outgoing link from node } i, \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

The relationship between a node and its incoming links is represented with a matrix $\mathbf{A}^{(m)-}$, whose elements are given by

$$a_{il}^{(m)-} = \begin{cases} 1, & \text{if link } l \text{ is an incoming link into node } i, \\ 0, & \text{otherwise.} \end{cases} \quad (2)$$

The upload capacity of peer i is denoted by $O_i^{(m)}$. The server is denoted as Peer 1. The server upload bandwidth s_m for channel m is actually the upload capacity of Peer 1, which is given by $O_1^{(m)} = s_m$. Upload capacity is typically the bottleneck in P2P systems. We consider *upload constraint* at peer i ($\forall i \in \mathbf{N}^{(m)}$), which is given by $\sum_{l \in \mathbf{L}^{(m)}} a_{il}^{(m)+} x_l^{(m)} \leq O_i^{(m)}$ where $x_l^{(m)}$ is the link rate at link l of overlay m . The upload constraint represents that the total outgoing rate from peer i is no larger than its upload capacity $O_i^{(m)}$. The streaming rate for channel m is denoted by $r^{(m)}$. Each peer except the server (e.g., Peer 1) receives the streaming rate $r^{(m)}$, which can be expressed by $\sum_{l \in \mathbf{L}^{(m)}} a_{il}^{(m)-} x_l^{(m)} = f_i^{(m)} r^{(m)}$, for all $i \in \mathbf{N}^{(m)}$, where $f_i^{(m)} = 0$ if $i = 1$, or $f_i^{(m)} = 1$ otherwise. The *download constraint* is given by $r^{(m)} \leq \min_{i \in \mathbf{N}^{(m)}} I_i^{(m)}$ where $I_i^{(m)}$ is the download capacity of peer i . From the download constraint, we can see that the maximal streaming rate is limited by the minimal download capacity among the peers. Therefore, the users with a very low download bandwidth should not be admitted into the P2P VoD system in order for maintaining a high streaming capacity.

The streaming capacity for channel m is stated as to maximize the streaming rate $r^{(m)}$ that can be received by every user in channel m by optimizing the streaming rate $r^{(m)}$ and the link rate $x_l^{(m)}$ ($\forall l \in \mathbf{L}^{(m)}$) under the upload constraint at each peer. Mathematically, the problem can be formulated as follows:

$$\begin{aligned} & \text{maximize} && r^{(m)} \\ & \text{subject to} && \sum_{l \in \mathbf{L}^{(m)}} a_{il}^{(m)-} x_l^{(m)} = f_i^{(m)} r^{(m)}, \quad \forall i \in \mathbf{N}^{(m)}, \\ & && \sum_{l \in \mathbf{L}^{(m)}} a_{il}^{(m)+} x_l^{(m)} \leq O_i^{(m)}, \quad \forall i \in \mathbf{N}^{(m)}, \\ & && x_l^{(m)} \geq 0, \quad \forall l \in \mathbf{L}^{(m)}, \\ & && 0 \leq r^{(m)} \leq \min_{i \in \mathbf{N}^{(m)}} I_i^{(m)}. \end{aligned} \quad (3)$$

The optimization problem in (3) is a Linear Programming (LP). It can be solved in a centralized way using

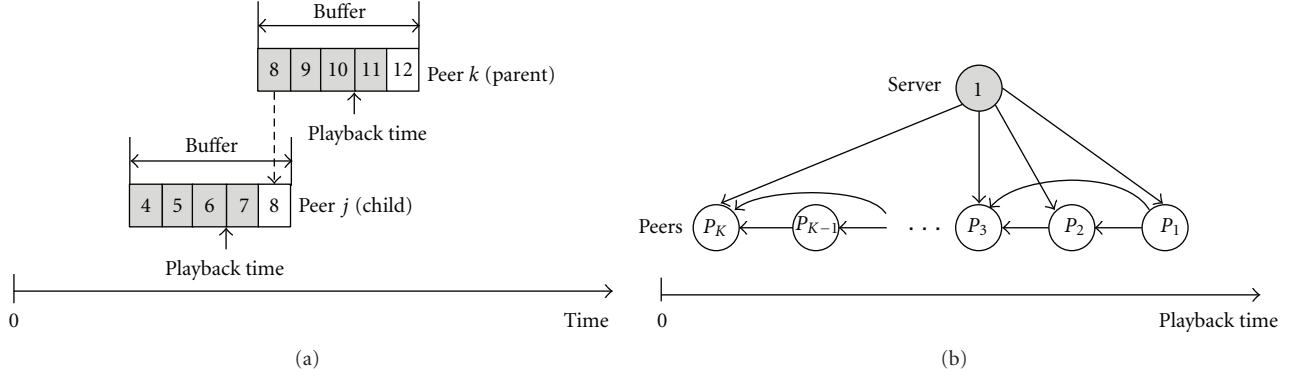


FIGURE 2: Overlay construction for a single channel in an independent-channel P2P VoD system: (a) the parent-child relationship between peer k and peer j , and (b) the complete overlay.

the interior point method [21]. However, the centralized solution is not scalable. Therefore, we will develop a distributed algorithm using the primal-dual method to solve the streaming capacity problem.

3.3. Distributed Algorithm. We will use primal-dual method [22] to develop a distributed algorithm for the optimization problem (3). However, the objective function in problem (3) is not strictly convex with respect to the optimization variables $r^{(m)}$ and $x_l^{(m)}$ ($\forall l \in \mathbf{L}^{(m)}$). Therefore, the corresponding dual function is nondifferentiable, and the optimal values of $r^{(m)}$ and $x_l^{(m)}$ ($\forall l \in \mathbf{L}^{(m)}$) are not immediately available. We first convert the original optimization problem (3) to an equivalent minimization problem by changing the objective from maximizing $r^{(m)}$ to minimizing $-r^{(m)}$. We then add a quadratic regularization term for the streaming rate $r^{(m)}$ and each link rate $x_l^{(m)}$ ($\forall l \in \mathbf{L}^{(m)}$) to make the objective function strictly convex. Finally, the optimization problem (3) is approximated to the following:

$$\begin{aligned}
 & \text{minimize} && -r^{(m)} + \varepsilon(r^{(m)})^2 + \varepsilon \sum_{l \in \mathbf{L}^{(m)}} (x_l^{(m)})^2 \\
 & \text{subject to} && \sum_{l \in \mathbf{L}^{(m)}} a_{il}^{(m)-} x_l^{(m)} = f_i^{(m)} r^{(m)}, \quad \forall i \in \mathbf{N}^{(m)}, \\
 & && \sum_{l \in \mathbf{L}^{(m)}} a_{il}^{(m)+} x_l^{(m)} \leq O_i^{(m)}, \quad \forall i \in \mathbf{N}^{(m)}, \\
 & && x_l^{(m)} \geq 0, \quad \forall l \in \mathbf{L}^{(m)}, \\
 & && 0 \leq r^{(m)} \leq \min_{i \in \mathbf{N}^{(m)}} I_i^{(m)},
 \end{aligned} \tag{4}$$

where $(\varepsilon(r^{(m)})^2 + \varepsilon \sum_{l \in \mathbf{L}^{(m)}} (x_l^{(m)})^2)$ is the sum of the regularization terms, and ε ($\varepsilon > 0$) is called the *regularization factor*. When ε is small enough, the solution for the problem in (4) is arbitrarily close to the solution for the original streaming capacity problem (3).

Let us denote by $(r^{(m)*}, x_l^{(m)*}$ ($\forall l \in \mathbf{L}^{(m)}$)) the optimal solution to the original optimization problem (3), and $(r^{(m)\$}, x_l^{(m)\$}$ ($\forall l \in \mathbf{L}^{(m)}$)) the optimal solution to the approximated optimization problem (4). Based on the original optimization problem (3), we have $r^{(m)*} \geq r^{(m)\$}$. Based

on the approximated optimization problem (4), we have $-r^{(m)\$} + \varepsilon(r^{(m)\$})^2 + \varepsilon \sum_{l \in \mathbf{L}^{(m)}} (x_l^{(m)\$})^2 \leq -r^{(m)*} + \varepsilon(r^{(m)*})^2 + \varepsilon \sum_{l \in \mathbf{L}^{(m)}} (x_l^{(m)*})^2$. Therefore, we can get $0 \leq r^{(m)*} - r^{(m)\$} \leq \varepsilon((r^{(m)*})^2 - (r^{(m)\$})^2 + \sum_{l \in \mathbf{L}^{(m)}} (x_l^{(m)*})^2 - \sum_{l \in \mathbf{L}^{(m)}} (x_l^{(m)\$})^2)$, from which we show that the streaming capacity obtained from the approximated optimization problem (4) with a small regularization factor can approach closely to the truly maximal streaming rate obtained from the original optimization problem (3).

The optimization problem (4) is a convex optimization problem with a strictly convex objective function and the linear constraints [23]. We introduce dual variables $(u_i, v_i, \forall i \in \mathbf{N}^{(m)})$ to formulate the Lagrangian corresponding to the primal problem (4) as below:

$$\begin{aligned}
 L(\mathbf{x}^{(m)}, r^{(m)}, \mathbf{u}, \mathbf{v}) &= -r^{(m)} + \varepsilon(r^{(m)})^2 + \varepsilon \sum_{l \in \mathbf{L}^{(m)}} (x_l^{(m)})^2 \\
 &+ \sum_{i \in \mathbf{N}^{(m)}} u_i \left(\sum_{l \in \mathbf{L}^{(m)}} a_{il}^{(m)-} x_l^{(m)} - f_i^{(m)} r^{(m)} \right) \\
 &+ \sum_{i \in \mathbf{N}^{(m)}} v_i \left(\sum_{l \in \mathbf{L}^{(m)}} a_{il}^{(m)+} x_l^{(m)} - O_i^{(m)} \right) \\
 &= \varepsilon(r^{(m)})^2 - r^{(m)} - r^{(m)} \sum_{i \in \mathbf{N}^{(m)}} u_i f_i^{(m)} \\
 &+ \sum_{l \in \mathbf{L}^{(m)}} \left(\varepsilon(x_l^{(m)})^2 \right. \\
 &\quad \left. + x_l^{(m)} \sum_{i \in \mathbf{N}^{(m)}} (u_i a_{il}^{(m)-} + v_i a_{il}^{(m)+}) \right) \\
 &- \sum_{i \in \mathbf{N}^{(m)}} v_i O_i^{(m)}.
 \end{aligned} \tag{5}$$

Let $I_{\min}^{(m)} = \min_{i \in \mathbf{N}^{(m)}} I_i^{(m)}$. The Lagrange dual function $G(\mathbf{u}, \mathbf{v})$ is the minimum value of the Lagrangian over the vector of link rates $\mathbf{x}^{(m)}$ and the streaming rate $r^{(m)}$:

$$G(\mathbf{u}, \mathbf{v}) = \min_{(x_l^{(m)} \geq 0, 0 \leq r^{(m)} \leq I_{\min}^{(m)})} \left\{ L(\mathbf{x}^{(m)}, r^{(m)}, \mathbf{u}, \mathbf{v}) \right\}. \tag{6}$$

The Lagrange dual problem is to maximize the Lagrange dual function. That is,

$$\begin{aligned} & \text{maximize} && G(\mathbf{u}, \mathbf{v}) \\ & \text{subject to} && v_i \geq 0, \quad \forall i \in \mathbf{N}^{(m)}. \end{aligned} \quad (7)$$

We use subgradient method [24] to solve the Lagrange dual problem (7). The dual variables $u_i^{(k+1)}$ and $v_i^{(k+1)}$ at the $(k+1)$ th iteration are updated, respectively, by

$$\begin{aligned} u_i^{(k+1)} &= u_i^{(k)} - \theta^{(k)} \left(f_i^{(m)}(r^{(m)})^{(k)} - \sum_{l \in \mathbf{L}^{(m)}} a_{il}^{(m)-} (x_l^{(m)})^{(k)} \right), \\ v_i^{(k+1)} &= \max \left\{ 0, v_i^{(k)} - \theta^{(k)} \left(O^{(m)} - \sum_{l \in \mathbf{L}^{(m)}} a_{il}^{(m)+} (x_l^{(m)})^{(k)} \right) \right\}, \end{aligned} \quad (8)$$

where $\theta^{(k)} > 0$ is the step size at the k th iteration. The algorithm is guaranteed to converge to the optimal value for a step-size sequence satisfying the nonsummable diminishing rule [24]:

$$\lim_{k \rightarrow \infty} \theta^{(k)} = 0, \quad \sum_{k=1}^{\infty} \theta^{(k)} = \infty. \quad (9)$$

The update of the streaming rate $(r^{(m)})^{(k)}$ at the k th iteration is given by

$$(r^{(m)})^{(k)} = \min \left\{ \max \left\{ 0, \frac{1 + \sum_{i \in \mathbf{N}^{(m)}} u_i^{(k)} f_i^{(m)}}{2\varepsilon} \right\}, I_{\min}^{(m)} \right\}. \quad (10)$$

At the k th iteration, the link rate $(x_l^{(m)})^{(k)}$ at link l can be calculated from the dual variables:

$$(x_l^{(m)})^{(k)} = \max \left\{ 0, \frac{-\sum_{i \in \mathbf{N}^{(m)}} (u_i^{(k)} a_{il}^{(m)-} + v_i^{(k)} a_{il}^{(m)+})}{2\varepsilon} \right\}. \quad (11)$$

As shown in (10) and (11), the optimization variables are decomposed. The optimization of the streaming rate is executed at the server as follows. At the k th iteration, the server collects the dual variable $u_i^{(k)}$ from peer i ($\forall i \in \mathbf{N}^{(m)}$), respectively, and then computes the streaming rate $(r^{(m)})^{(k)}$. The optimization of each link rate is executed at each peer as follows. At the k th iteration, each peer only collects the dual variable $u_j^{(k)}$ from each of its children, and then computes the link rate for each of the outgoing links.

A P2P system is inherently dynamic. Peers may join or leave the system at any time. Therefore, the streaming capacity for a single channel varies with time. To handle the time-varying streaming capacity, the server can encode the video using a scalable coding scheme and then adjust the output video rate adaptive to the current streaming capacity.

4. Streaming Capacity for a Correlated-Channel P2P VoD System

In a correlated-channel P2P VoD system, the resources among different channels can be shared with each other. The average streaming capacity in a correlated-channel P2P VoD system can be obtained by optimizing both the intra-channel resource allocation and the cross-channel resource allocation. The optimization of intra-channel resource allocation for a single channel has been presented in Section 3. In this section, we will focus on the cross-channel resource allocation. We will first optimize the server upload allocation among channels to maximize the average streaming capacity in Section 4.1, and then further utilize cross-channel peer upload bandwidth to improve the average streaming capacity in Section 4.2.

4.1. Optimization of Server Upload Allocation among Channels. The server upload allocated for channel m is denoted by s_m . In an independent-channel P2P VoD system, the server upload allocation for each channel is predetermined. Therefore, the server upload bandwidth is not utilized in an optimal way. In this subsection, we treat $\{s_m, \forall m \in \mathbf{M}\}$ as variables and optimize them to maximize the average streaming capacity for a correlated-channel P2P VoD system.

The optimization of server upload allocation for a correlated-channel P2P VoD system is stated as to maximize the average streaming rate by optimizing the server upload allocation, the link rates, and the streaming rate, for each channel, subject to the upload constraint at each peer. Since maximizing the average streaming rate is equivalent to minimizing the negative average streaming rate, we formulate the problem into a minimization problem as follows:

$$\begin{aligned} & \text{minimize} && - \sum_{m \in \mathbf{M}} p_m r^{(m)} \\ & \text{subject to} && \sum_{l \in \mathbf{L}^{(m)}} a_{il}^{(m)-} x_l^{(m)} = f_i^{(m)} r^{(m)}, \\ & && \forall i \in \mathbf{N}^{(m)}, \forall m \in \mathbf{M}, \\ & && \sum_{l \in \mathbf{L}^{(m)}} a_{il}^{(m)+} x_l^{(m)} \leq O_i^{(m)}, \\ & && \forall i \in \mathbf{N}^{(m)}, \forall m \in \mathbf{M}, \\ & && x_l^{(m)} \geq 0, \quad \forall l \in \mathbf{L}^{(m)}, \\ & && \sum_{m \in \mathbf{M}} s_m \leq s_T, \\ & && s_m \geq 0, \quad \forall m \in \mathbf{M}, \\ & && 0 \leq r^{(m)} \leq \min_{i \in \mathbf{N}^{(m)}} I_i^{(m)}, \quad \forall m \in \mathbf{M}, \end{aligned} \quad (12)$$

where p_m is the priority for channel m , and s_T is the server upload capacity. The optimization problem in (12) is an LP. The optimization variables in the optimization problem (12) are the server upload allocation s_m , the link rate vector $\mathbf{x}^{(m)}$, and the streaming rate $r^{(m)}$, for channel m , for all $m \in \mathbf{M}$. The number of the optimization variables is increased with the number of the channels and the number of the links in

each overlay. In order to solve the optimization efficiently, we use dual decomposition [22] to decompose the optimization problem (12) into multiple subproblems, each of which is associated with a channel.

We introduce a dual variable λ for the inequality constraint $\sum_{m \in \mathbf{M}} s_m \leq s_T$. The Lagrangian corresponding to the primal problem (12) is given by $L(\mathbf{s}, \mathbf{x}, \mathbf{r}, \lambda) = -\sum_{m \in \mathbf{M}} p_m r^{(m)} + \lambda(\sum_{m \in \mathbf{M}} s_m - s_T)$. Then, the Lagrange dual function [23] is given by $G(\lambda) = \min L(\mathbf{s}, \mathbf{x}, \mathbf{r}, \lambda) = \sum_{m \in \mathbf{M}} \min(-p_m r^{(m)} + \lambda s_m) - \lambda s_T$.

The Lagrange dual problem is to maximize the Lagrange dual function [23]. That is,

$$\begin{aligned} & \text{maximize} && G(\lambda) \\ & \text{subject to} && \lambda \geq 0. \end{aligned} \quad (13)$$

Subgradient method [24] is used to solve the Lagrange dual problem (13). The dual variable λ is updated at the $(k+1)$ th iteration by $\lambda^{(k+1)} = \max\{0, \lambda^{(k)} - \theta^{(k)}(s_T - \sum_{m \in \mathbf{M}} s_m^{(k)})\}$ where $\theta^{(k)}$ is the step size at the k th iteration. In order to guarantee the convergence, the sequence of the step sizes is required to satisfy the nonsummable diminishing rule in (9).

At the k th iteration, the primal variables $(s_m, \mathbf{x}^{(m)}, r^{(m)})$ for channel m are obtained by solving the following optimization problem:

$$\begin{aligned} & \text{minimize} && -p_m r^{(m)} + \lambda s_m \\ & \text{subject to} && \sum_{l \in \mathbf{L}^{(m)}} a_{il}^{(m)-} x_l^{(m)} = f_i^{(m)} r^{(m)}, \quad \forall i \in \mathbf{N}^{(m)}, \\ & && \sum_{l \in \mathbf{L}^{(m)}} a_{il}^{(m)+} x_l^{(m)} \leq O_i^{(m)}, \quad \forall i \in \mathbf{N}^{(m)}, \\ & && x_l^{(m)} \geq 0, \quad \forall l \in \mathbf{L}^{(m)}, \\ & && 0 \leq r^{(m)} \leq \min_{i \in \mathbf{N}^{(m)}} I_i^{(m)}, \quad s_m \geq 0. \end{aligned} \quad (14)$$

The optimization problem (12) is decomposed into $|\mathbf{M}|$ subproblems where $|\mathbf{M}|$ is the number of the channels. Subproblem m , represented in (14), is associated with channel m . Subproblem m , for all $m \in \mathbf{M}$, can be solved with a distributed algorithm.

4.2. Cross-Channel Sharing of Peer Upload Bandwidth.

Though the server upload allocation among channels is optimized, the cross-channel resources have not yet been fully utilized. In each channel, there are a number of under-utilized peers, which can be utilized by the other channels. For example, the leaf nodes of overlay m contribute zero upload bandwidth to the channel because they have no outgoing links. We can utilize the upload bandwidth of the leaf nodes in overlay m to serve the peers in another channel (e.g., channel k), thus improving the streaming capacity of channel k . However, it is challenging to establish the cross-channel links to enable the cross-channel sharing of peer upload bandwidth.

In this paper, we propose a scheme for cross-channel peer upload sharing. We introduce the concept of *cross-channel helpers*. A *cross-channel helper* is the peer who uses its remaining upload bandwidth to help other peers in another channel.

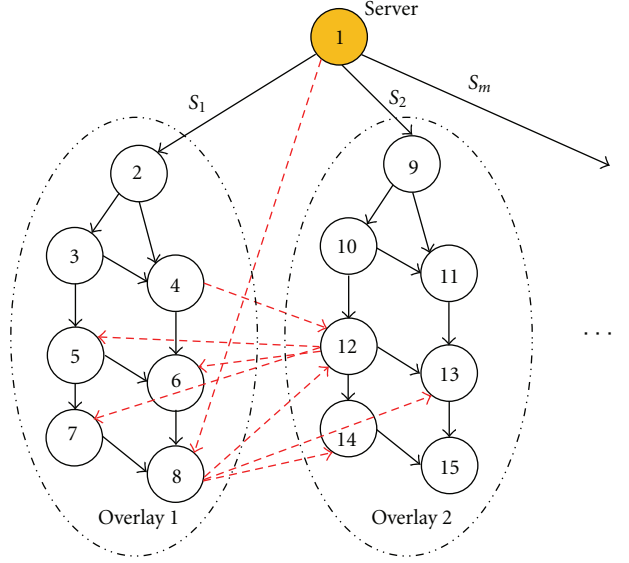


FIGURE 3: Illustration of a correlated-channel P2P VoD system.

Cross-channel helpers are chosen from the peers who have a remaining upload bandwidth greater than a threshold.

Suppose that peer i in channel m is a cross-channel helper serving segment j of channel k , and it has a remaining upload bandwidth b_i . We denote the set of peers watching segment j in channel k by $\mathbf{H}_j^{(k)}$. Peer i can download segment j of channel k from the server or the peers who are buffering segment j , at a rate R_i^{IN} , and then output it to peer h , for all $h \in \mathbf{H}_j^{(k)}$, at a rate R_i^h , respectively. The *bandwidth gain* for peer i is defined as $g_i = (\sum_{h \in \mathbf{H}_j^{(k)}} R_i^h) / R_i^{\text{IN}}$. In order to maximize the bandwidth gain g_i under the flow constraint $R_i^h \leq R_i^{\text{IN}}$, for all $h \in \mathbf{H}_j^{(k)}$, and the bandwidth constraint $\sum_{h \in \mathbf{H}_j^{(k)}} R_i^h \leq b_i$, the optimal download rate at peer i is $R_i^{\text{IN}*} = b_i / |\mathbf{H}_j^{(k)}|$ where $|\mathbf{H}_j^{(k)}|$ is the number of the peers watching segment j of channel k , and the optimal outgoing rate to peer h , for all $h \in \mathbf{H}_j^{(k)}$, is $R_i^{h*} = R_i^{\text{IN}*}$. The bandwidth gain should be larger than 1, otherwise, the cross-channel helper consumes a larger bandwidth than it contributes. Figure 3 illustrates a correlated-channel P2P VoD system. As shown in Figure 3, peer 8 in overlay 1 is a cross-channel helper, who downloads a segment from the server and then forwards it to peers 12–14 in overlay 2; peer 12 in overlay 2 is also a cross-channel helper, who downloads a segment from peer 4 in overlay 1 and then forwards it to peers 5–7 in overlay 1.

The proposed scheme for cross-channel peer upload sharing is described as follows.

- (1) Channel m ($\forall m \in \mathbf{M}$) finds a partner, channel k ($k \neq m, k \in \mathbf{M}$), to help each other, by using a *resource-balancing scheme*, which is described as follows. (i) Calculate B_m^{re} , the amount of the total remaining bandwidth, for channel m ($\forall m \in \mathbf{M}$), the average amount of the total remaining bandwidth is given by $B_{\text{avg}}^{\text{re}} = (\sum_{m \in \mathbf{M}} B_m^{\text{re}}) / |\mathbf{M}|$ where $|\mathbf{M}|$

- represents the number of channels. (ii) Given the set of unchosen channels \mathbf{M}^{un} , the partner of channel m is determined by $k = \arg_{j \in \mathbf{M}^{\text{un}}} \min((1/2)(B_j^{\text{rc}} + B_m^{\text{rc}}) - B_{\text{avg}}^{\text{rc}})^2$.
- (2) Determine the set of cross-channel helpers in channel m , denoted by $\mathbf{Y}^{(m)}$, by choosing the peers who have a remaining upload bandwidth greater than a threshold b_{th} . Sort $\mathbf{Y}^{(m)}$ in a descending order based on the remaining upload bandwidth.
 - (3) Determine the *demanding segment set* in channel k , denoted by $\mathbf{W}^{(k)}$, by choosing $|\mathbf{Y}^{(m)}|$ segments which are watched by the largest number of peers. Sort $\mathbf{W}^{(k)}$ in a descending order based on the number of watching peers.
 - (4) Assign the i th cross-channel helper in $\mathbf{Y}^{(m)}$ to serve the peers watching the i th segment in $\mathbf{W}^{(k)}$. Determine the incoming rate and outgoing rates at the i th cross-channel helper to maximize the bandwidth gain.
 - (5) After allocating the rate for each of the cross-channel links as in Step (4), revise the optimization problem (12) by integrating the cross-channel link rates and then solve it to obtain the optimal server upload allocation for each channel and the optimal link rates within each overlay.

Peer dynamics have an impact on the streaming capacity in correlated-channel P2P VoD systems. First, the peers may leave or join a channel dynamically. Second, the cross-channel helpers may leave the channel, which causes the disconnection of the cross-channel links. To handle the dynamic conditions, the optimizations of intra-channel resource allocation and cross-channel resource allocation need to be performed in a discrete-time manner, in which the peers and the overlay are assumed to remain unchanged during a time slot, and the algorithms for resource allocation are performed at the beginning of each time slot.

5. Simulations

In the simulations, we use two classes of peers: cable/DSL peers and Ethernet peers. Cable/DSL peers take 85% of the total peer population with download capacity uniformly distributed between 0.9 Mbps and 1.5 Mbps and upload capacity uniformly distributed between 0.3 Mbps and 0.6 Mbps. Ethernet peers take the remaining 15% of the total peer population with both upload and download capacities uniformly distributed between 1.5 Mbps and 3.0 Mbps. The length of the video is 60 minutes, which is evenly divided into 60 segments. Each peer maintains a buffer with a capacity of 5 segments. The playback time of each peer is randomly distributed between 0 and 60 minutes. The priorities for all channels are equal.

Figure 4 compares the average streaming capacity between the *equal server upload* scheme and the *optimized server upload* scheme. In the *equal server upload* scheme, each channel is allocated an equal server upload, the link

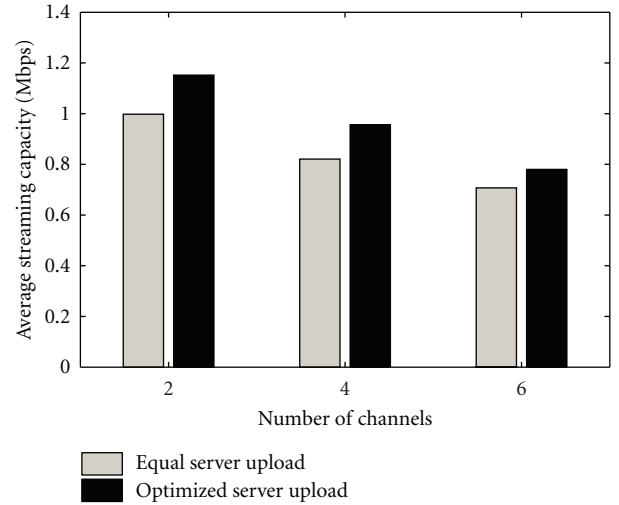


FIGURE 4: Comparison of average streaming capacity with different number of channels.

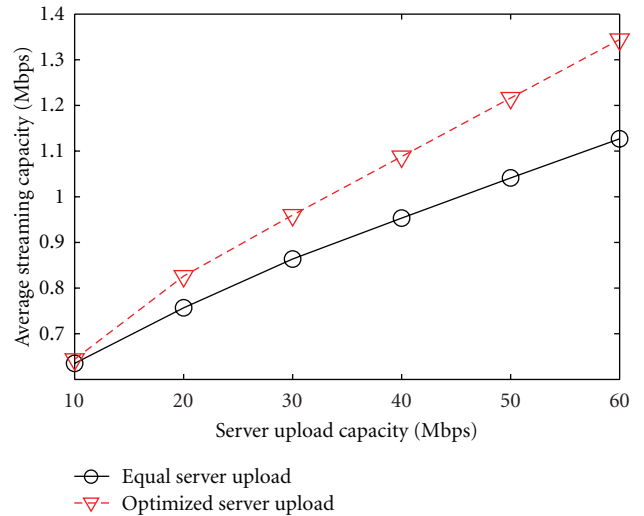


FIGURE 5: Impact of server upload capacity to the average streaming capacity.

rates within each channel are optimized by solving the optimization problem (3). In the *optimized server upload* scheme, the server upload for each channel and the link rates within each channel are jointly optimized by solving the optimization problem (12). The *equal server upload* scheme considers only intra-channel resource allocation [10], while the *optimized server upload* scheme considers both intra-channel and cross-channel resource allocation. The number of the peers in a channel is uniformly distributed between 10 and 150. The server upload capacity is 45.0 Mbps. The optimized server upload scheme improves the average streaming capacity by 14.1% in average compared to the equal server upload scheme.

We show in Figure 5 the impact of server upload capacity to the average streaming capacity for a P2P VoD system with 2 channels. The first channel has 40 peers, and the

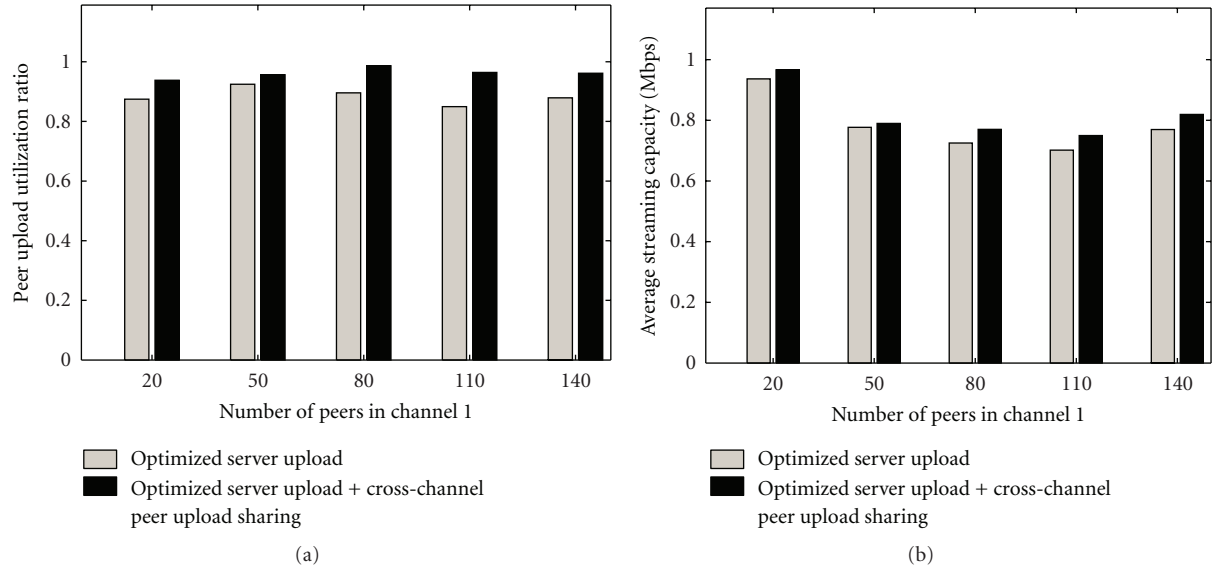


FIGURE 6: Performance improvement brought by cross-channel peer upload sharing: (a) peer upload utilization ratio, and (b) average streaming capacity.

second one has 120 peers. As shown in Figure 5, the average streaming capacity is increased with the server upload capacity. The improvement brought by the optimized server upload scheme is larger when the server upload capacity is larger.

Joint consideration of server upload optimization and cross-channel peer upload sharing can improve the performance in a correlated P2P VoD system, as shown in Figure 6. There are two channels with total 200 peers in the P2P VoD system. The server upload capacity is 20.0 Mbps. We vary the number of the peers in channel 1 from 20 to 140 and evaluate two metrics: peer upload utilization ratio and average streaming capacity. *Peer upload utilization ratio* is defined by $\beta = \frac{\sum_{i \in N, i \neq 1} v_i}{\sum_{i \in N, i \neq 1} O_i}$ where $\sum_{i \in N, i \neq 1} v_i$ represents the sum of the outgoing rates from all peers except the server, and $\sum_{i \in N, i \neq 1} O_i$ represents the sum of the upload capacities of all peers except the server. Due to a better utilization of cross-channel resources, joint consideration of server upload optimization and cross-channel peer upload sharing improves the peer upload utilization ratio by 7.7% in average, as shown in Figure 6(a), and improves the average streaming capacity by 0.04 Mbps in average, as shown in Figure 6(b), compared to *optimized server upload scheme*.

6. Conclusion

In this paper, we studied the streaming capacity problem for multi-channel P2P VoD systems. Depending on the resource correlation, multi-channel P2P VoD systems can be categorized into independent-channel P2P VoD systems and correlated-channel P2P VoD systems. Since there is no resource correlation among channels in an independent-channel P2P VoD system, we just need to find the streaming capacity for each individual channel, respectively. We formulated the streaming capacity problem for a single channel into

an LP problem and solved it with a distributed algorithm. In a correlated-channel P2P VoD system, we optimized both the intra-channel resource allocation and cross-channel resource allocation to improve the streaming capacity. In order to better utilize the cross-channel resource, we first optimized the server upload allocation among channels to maximize the average streaming capacity and then introduced cross-channel helpers to enable cross-channel sharing of peer upload bandwidth. We demonstrated in the simulations that the correlated-channel P2P VoD systems with both intra-channel and cross-channel resource allocation can achieve a higher average streaming capacity compared to the independent-channel P2P VoD systems with only intra-channel resource allocation.

References

- [1] J. Li, "PeerStreaming: an on-demand peer-to-peer media streaming solution based on a receiver-driven streaming protocol," in *Proceedings of the IEEE 7th Workshop on Multimedia Signal Processing (MMSP '05)*, pp. 1–4, November 2005.
- [2] H. Chi, Q. Zhang, J. Jia, and X. Shen, "Efficient search and scheduling in P2P-based media-on-demand streaming service," *IEEE Journal on Selected Areas in Communications*, vol. 25, no. 1, pp. 119–130, 2007.
- [3] S. Annapureddy, C. Gkantsidis, and P. Rodriguez, "Providing video-on-demand using peer-to-peer networks," in *Proceedings of the International Conference on World Wide Web*, 2006.
- [4] C. Huang, J. Li, and K. W. Ross, "Peer-assisted VoD: making Internet video distribution cheap," in *Proceedings of the IPTPS*, 2007.
- [5] Y. Huang, T. Z. J. Fu, D. M. Chiu, J. C. S. Lui, and C. Huang, "Challenges, design and analysis of a large-scale p2p-vod system," in *Proceedings of the ACM SIGCOMM Conference on Data Communication (SIGCOMM '08)*, pp. 375–388, August 2008.

- [6] S. Sengupta, S. Liu, M. Chen, M. Chiang, J. Li, and P. A. Chou, "Streaming capacity in peer-to-peer networks with topology constraints," Microsoft Research Technical Report, 2008.
- [7] S. Sengupta, S. Liu, M. Chen, M. Chiang, J. Li, and P. A. Chou, "Peer-to-peer streaming capacity," *IEEE Transactions on Information Theory*, vol. 57, no. 8, pp. 5072–5087, 2011.
- [8] S. Liu, R. Zhang-Shen, W. Jiang, J. Rexford, and M. Chiang, "Performance bounds for Peer-assisted live streaming," in *Proceedings of the ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS '08)*, pp. 313–324, June 2008.
- [9] S. Liu, M. Chen, S. Sengupta, M. Chiang, J. Li, and P. A. Chou, "P2P streaming capacity under node degree bound," in *Proceedings of the 30th IEEE International Conference on Distributed Computing Systems (ICDCS '10)*, pp. 587–598, June 2010.
- [10] Y. He and L. Guan, "Streaming capacity in P2P VoD systems," in *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS '09)*, pp. 742–745, May 2009.
- [11] Y. He and L. Guan, "Improving the streaming capacity in P2P VoD systems with helpers," in *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME '09)*, pp. 790–793, July 2009.
- [12] Y. He and L. Guan, "Solving streaming capacity problems in P2P VoD systems," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 20, no. 11, pp. 1638–1642, 2010.
- [13] Y. He, I. Lee, and L. Guan, "Distributed throughput maximization in P2P VoD applications," *IEEE Transactions on Multimedia*, vol. 11, no. 3, pp. 509–522, 2009.
- [14] J. Wang, C. Yeo, V. Prabhakaran, and K. Ramchandran, "On the role of helpers in peer-to-peer file download systems: design, analysis and simulation," in *Proceedings of the IPTPS*, 2007.
- [15] P. Garbacki, D. Epema, J. Pouwelse, and M. van Steen, "Offloading servers with collaborative video on demand," in *Proceedings of the IPTPS*, 2008.
- [16] C. Wu, B. Li, and S. Zhao, "Multi-channel live P2P streaming: refocusing on servers," in *Proceedings of the 27th IEEE Communications Society Conference on Computer Communications (INFOCOM '08)*, pp. 2029–2037, April 2008.
- [17] D. Wu, C. Liang, Y. Liu, and K. Ross, "View-upload decoupling: a redesign of multi-channel P2P video systems," in *Proceedings of the IEEE 28th Conference on Computer Communications (INFOCOM '09)*, pp. 2726–2730, April 2009.
- [18] D. Wu, Y. Liu, and K. W. Ross, "Queuing network models for multi-channel P2P live streaming systems," in *Proceedings of the IEEE 28th Conference on Computer Communications (INFOCOM '09)*, pp. 73–81, April 2009.
- [19] M. Wang, L. Xu, and B. Ramamurthy, "Linear programming models for multi-channel P2P streaming systems," in *Proceedings of the IEEE Conference on Computer Communications (INFOCOM '10)*, 2010.
- [20] C. Zhao, X. Lin, and C. Wu, "The streaming capacity of sparsely-connected P2P systems with distributed control," in *Proceedings of the IEEE Conference on Computer Communications (INFOCOM '11)*, pp. 1449–1457, 2011.
- [21] R. J. Vanderbei, *Linear Programming: Foundations and Extensions*, Springer, 2nd edition, 2001.
- [22] D. Palomar and M. Chiang, "A tutorial on decomposition methods and distributed network resource allocation," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 8, pp. 1439–1451, 2006.
- [23] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, 2004.
- [24] D. P. Bertsekas, A. Nedic, and A. E. Ozdaglar, *Convex Analysis and Optimization*, Athena Scientific, 2003.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

