Providing Incentives in P2P Adaptive Streaming

Guibin Tian*, Yang Xu*, Yong Liu*, and Keith Ross†
*Electrical & Computer Engineering, Polytechnic Institute of NYU, Brooklyn, NY 11201
† Computer Science and Engineering, Polytechnic Institute of NYU, Brooklyn, NY 11201

***Abstract*—Adaptive streaming, such as Dynamic Adaptive Streaming over HTTP (DASH), has been widely deployed to provide uninterrupted video streaming service to users with dynamic network conditions. In this report, we analytically study the potential of using P2P in conjunction with adaptive streaming. Our focus is on how to provide incentives in P2P adaptive streaming. We first study P2P adaptive streaming under a *taxation* framework. Then we develop more general pricing mechanisms for P2P adaptive streaming using cooperative game theory.**

I. Introduction

We have recently witnessed the wide deployment of *adaptive streaming* that provides uninterrupted video streaming service to users with dynamic network conditions. To our knowledge, all deployed adaptive streaming solutions to date are server-based [1]. Notably, Netflix's online video streaming service is implemented using Dynamic Adaptive Streaming over HTTP (DASH) [2], [3]. In adaptive streaming, the video server encodes the video into multiple versions at different rates. Each client then dynamically chooses a video version that matches the available bandwidth along the server-client connection. To ensure continuous playback, low quality video will be streamed if either the server is overloaded, or the server-client connection has low available bandwidth. P2P video streaming is a proven technology that can efficiently reduce the load on servers, and provide robust video streaming services in face of peer churn and bandwidth variations [4], [5], [6], [7], [8]. It is therefore natural to consider integrating P2P into adaptive streaming.

In P2P adaptive streaming, peers have heterogeneous and time-varying upstream and downstream bandwidth availability. A peer dynamically switches between video versions to match its current network condition. A peer downloads video either from the server, or from other peers watching the same version. To maximally exploit the *multiplexing gain*, it is desirable to facilitate P2P sharing among peers watching different versions of the same video. Towards such a paradigm, the key design questions for P2P adaptive streaming are:

1) *Which video version (rate) should each peer receive?*
2) *How should we generate and distribute multiple versions of the same video among heterogeneous peers?*
3) *How to we deliver stable video quality to peers in face of temporal bandwidth variations?*

Video rate allocation among peers reflects the fundamental trade-off between providing *social equality* and *contribution incentives.* On one hand we want to maximize the minimum viewing quality across all peers; on the other hand, we want to incentivize individual peers to maximally contribute bandwidth by providing them a better viewing experience. One extreme is to pool all upload bandwidth in the system and evenly distribute it to all peers so that they watch the same video version. This design is "fair" but does not provide incentives for peers to contribute upload bandwidth. Another extreme is to make a peer's video download rate equal to its upload contribution, so that peers are motivated to contribute more to improve their viewing experience. However, in this case low bandwidth peers will receive very poor quality video. Also a peer with temporary upload bandwidth dips will experience immediate video quality degradation. This works against P2P's multiplexing advantage, both spatially (among heterogeneous peers at the same time), and temporally (cross different time instants on a single peer).

II. Taxation-based P2P Adaptive Streaming

In this section, we employ *taxation* to strike a balance between fairness and incentives in P2P adaptive streaming.To enable video sharing between peers watching different versions of the same video, transcoding can be applied: a peer can transcode its received video into a different (normally lower) quality level, and upload it to other peers watching at that level. More recently, scalable video coding techniques, such as layered video and MDC, have been adopted in P2P streaming. Both of them incur computation and coding overhead on the servers and peers. Another alternative is the *helper-based* design, where a peer downloads a sub-stream of a video version different from the version it is watching, and then uploads it to other peers watching that version. Different schemes call for different video generation and P2P distribution designs. Finally, to achieve video stability, both rate allocation and P2P sharing have to be robust against temporal bandwidth variations.

A. Taxation-based Incentive

Taxation-based incentive policy offers a flexible framework that allows the tradeoff between the system-wide social welfare and the incentive to individuals [9], [10], [11]. Let u_d be the upload bandwidth contributed by peer d. Under a tax rate $0 \leq t \leq 1$, the target received video rate of peer d is

$$r_d = (1-t)u_d + r_d^{(P)},$$

where the first portion is called the *entitled rate*, which is a fraction of its own upload contribution $(1-t)u_d$, and the second portion is a share from the taxed bandwidth pool shared by all users. If $t = 0$, the allocation degenerates into the 'tit-for-tat' incentive: a peer's video download rate matches its video upload rate; if $t = 1$, all peers' upload bandwidth are taxed to the common pool to maximize the social welfare.

B. Model with Continuous Video Rates

Our design objective is to maximize the aggregate video quality on all peers under the taxation incentive policy. We consider a system with one server and N classes of peers. The server upload bandwidth is u_s. Let S_i be the set of peers in class i. There are n_i peers in S_i, each of them has upload bandwidth of u_i. Without loss of generality, we assume peer classes are ordered in a decreasing order of their upload bandwidth, $u_1 > u_2 > \cdots > u_N$. Let r_{ij} be received video rate of peer j in S_i. We introduce vector notations $\mathcal{U} \triangleq \{u_i, 1 \leq i \leq N\}$ and $\mathcal{N} \triangleq \{n_i, 1 \leq i \leq N\}$. Let $\mathcal{R} \triangleq \{r_{ij} 1 \leq i \leq N, 1 \leq j \leq n_i\}$ be the received video rates on all peers. PSNR (Peak Signal-to-Noise Ratio) is the standard objective metric to evaluate the quality of a compressed video. PSNR of a video coded at rate r_c can be approximated by a logarithmic function $\beta \log(r_c)$, where β is a constant related to the video feature. In this section, we study the case that the server can generate arbitrary number of video versions, each of which can be at arbitrary rate. We study the system capacity under three situations: video transcoding on peers, layered video coding on server, helper-based solution without transcoding and layered coding.

C. Optimal Rate Allocation among Peers

When peers' upload bandwidth are the only bottleneck, the optimal video rate allocation among all peers should maximize the aggregate video quality.

OPT I objective:

$$\max_{\mathcal{R}} \quad \sum_{i=1}^{N} \sum_{j=1}^{n_i} \log(r_{ij}), \tag{1}$$

subject to:

$$r_{ij} \geq (1-t)u_i, \forall i = 1, 2, \cdots, N; j = 1, 2, \cdots, n_i \tag{2}$$

$$\sum_{i=1}^{N} \sum_{j=1}^{n_i} r_{ij} \leq \sum_{i=1}^{N} n_i u_i + u_s. \tag{3}$$

In OPT I, (1) denotes the aggregate utility of all peers. (2) states that each peer should get at least its entitled rate. (3) states that the aggregate peer video download rate can not exceed the aggregate video upload rate in the system.

We develop a water-filling type of algorithm to get a feasible solution of **OPT I**. In Algorithm 1, each peer reports its upload bandwidth to a tracker. After collecting all peers' information, the tracker can calculate K^*, W^*, and further determines the video rates $\mathcal{R}$ of all peers. In the algorithm, B is the taxed bandwidth pool can be used to maximize the system-wide utility. According to the water-filling policy, one should always use the common tax pool to help "weaker" peers. If B is used to only help peers in class k and above, those helped peers can get the same video rate at

$$W_k = \frac{\sum_{i=1}^{N} n_i u_i + u_s - (1-t)\sum_{i=1}^{(K-1)} n_i u_i}{\sum_{i=K}^{N} n_i} \tag{4}$$

Algorithm 1 Water-Filling-Continuous $(\mathcal{U}, \mathcal{N}, u_s, N)$

All peers enter a FIFO queue $Queue_p$ in the increasing order of their upload bandwidth.
for each peer j in S_i **do**
  $r_{ij} = (1-t)u_i$
end for
Put residual bandwidth of peers and servers to a pool B, now $B = t\sum_{i=1}^{N}(u_i n_i) + u_s$. Initialize $S_w = \phi$.
while 1 **do**
  Select peers with the same smallest upload bandwidth u_j out of $Queue_p$ and assume set of those peers is S_j.
  if $\frac{B+(1-t)|S_j|u_j}{|S_w|+|S_j|} < (1-t)u_j$ **then**
    $W^* = \frac{B}{|S_w|}, K^* = j+1, S_K^* = S_w$
    $r_{ij} = W^*, \forall (i,j) \in S_w$
    break
  else
    $B = B + (1-t)|S_j|u_j, S_w = S_w \cup S_j,$
  end if
end while

To find a feasible solution satisfying the entitled rate constraint, we have to make $W_k \geq (1-t)u_k$. In the water-filling algorithm, we try to find K^*, the smallest k such that $W_k \geq (1-t)u_k$. Let $W^* = W_{K^*}$, then the received video rates of all peers are given as

$$r_{ij}^* = \begin{cases} (1-t)u_i, & \text{if} \quad (1-t)u_i > W^* \\ W^*, & \text{if} \quad (1-t)u_i \leq W^* \end{cases} \tag{5}$$

In other words, all peers at least get the base rate of W^*, and peers in class 1 through K^*-1 will get their entitled rates, which are higher than W^*.

Theorem 1: The video rate $\mathcal{R}^*$ obtained by Algorithm 1 is the global maximum solution of **OPT I**.

Proof: We can formulate **OPT I** into a standard convex programming problem with the following form:

$$\begin{aligned} &\textbf{max} \quad f(\mathcal{R}) \\ &\textbf{subject to} \quad A\mathcal{R} \geq b \end{aligned}$$

Using Karush-Kuhn-Tucker (KKT) conditions, one can easily verify that, for the obtained $\mathcal{R}^*$, there exists λ^* such that

$$\begin{aligned} &\bigtriangledown f(\mathcal{R}^*) = A^{\mathrm{T}}\lambda_*, \quad \lambda_* \geq 0, \\ &\lambda_*^{\mathrm{T}}(A\mathcal{R}^* - b) = 0, \\ &Z^{\mathrm{T}}\bigtriangledown^2 f(\mathcal{R}^*)Z \quad \text{is positive semi-definite,} \end{aligned}$$

where Z is a null-space matrix for the matrix of active constraints at $\mathcal{R}^*$. Generally, the KKT condition is a necessary condition for the solution to be optimal. Since the objective $\log()$ here is strictly concave, KKT condition is also a sufficient condition. Thus, $\mathcal{R}^*$ is the global optimal solution. ■

To achieve the optimal rate $\mathcal{R}^*$, a feasible P2P video

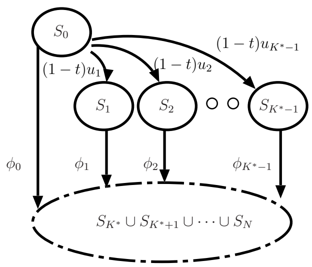


Fig. 1: Trans-coding Once

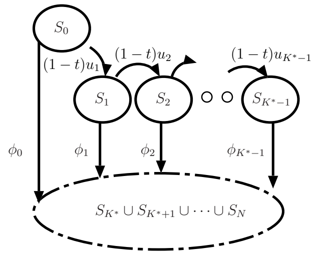


Fig. 2: Trans-coding Twice

sharing scheme has to be developed. For single-rate P2P video streaming, it has been shown [12], [13] that in a P2P swarm with n peers, the optimal achievable video rate on all peers is

$$r = \min\{u_s, \frac{u_s + B_n}{n}\}, \tag{6}$$

where u_s is the server upload bandwidth and B_n is the aggregate upload bandwidth among all n peers. It has also been shown that distributed P2P distribution designs can approach the optimal rate [14], [15].

In (5), there are K^* different video rates to be achieved between different classes of peers. Based on (6), for each video rate r with n peers watching at that rate, it is sufficient to design P2P video distribution such that there is a server/peer generating the video version at rate r, and the total upload bandwidth reserved for those n peers is nr. In the following, we investigate how to achieve the optimal rate $\mathcal{R}^*$ with peer transcoding, layered coding, and helper-based P2P distribution.

D. P2P Distribution with Peer Transcoding

If peers have video transcoding capabilities, a peer receiving a video can transcode the video to multiple lower rates and upload them to peers watching at the lower rates.

Theorem 2: If the server's upload bandwidth satisfies $u_s \geq (1-t)\sum_{i=1}^{K^*-1} u_i$, the optimal rate $\mathcal{R}^*$ can be achieved as long as peers can do video transcoding once.

Proof: We prove this by construction. Fig. 1 illustrates the video distribution between the server and peers. The server, denoted by S_0, produces K^* video versions. For the i-th version, where $1 \leq i \leq K^*-1$, the rate is $(1-t)u_i$. We assume server's upload bandwidth $u_s \geq (1-t)\sum_{i=1}^{K^*-1} u_i$. Thus, it is large enough to deliver i-th video to the corresponding peers in S_i. For the K^*-th video, that video version rate is W^*. Server contributes its residual bandwidth $\phi_0 = u_s - (1-t)\sum_{i=1}^{K^*-1} u_i$ to distribute that video version.

For peers in $S_i (1 \leq i \leq K^*-1)$, after getting video version with rate $(1-t)u_i$ from the server S_0, they only need $(1-t)(n_i-1)u_i$ more bandwidth to send to all peers in S_i their entitled rate. It is achievable since the total upload bandwidth of these peers $n_i u_i$ is larger than the bandwidth needed. Then, peers in S_i can do transcoding to generate video version with rate W^* and contribute their residual bandwidth $\phi_i = n_i u_i - (1-t)(n_i-1)u_i$ to peers watching the base rate.

Then, we only need to show that all peers in set $S_{K^*} \cup S_{K^*+1} \cup \cdots \cup S_N$ can receive video version with rate W^*. The total aggregate upload bandwidth Φ to distribute that video version is the sum of upload bandwidth of those peers watching that video version and residual bandwidth from set $S_0 \cup S_1 \cup S_2 \cup \cdots \cup S_{K^*-1}$. Thus, the value of Φ can be expressed as below:

$$\begin{aligned}\Phi &= \sum_{i=K^*}^{N} n_i u_i + \phi_0 + \phi_1 + \cdots + \phi_{K^*-1} \\ &= \sum_{i=K^*}^{N} n_i u_i + u_s + t \sum_{i=1}^{K^*-1} n_i u_i = (\sum_{i=K^*}^{N} n_i) W^*\end{aligned}$$

Hence, all peers in set $S_{K^*} \cup S_{K^*+1} \cup \cdots \cup S_N$ can successfully receive video rate W^*. ∎

Even if the server bandwidth is only enough to send out one stream at the highest rate, the optimal rates $\mathcal{R}^*$ can still be achieved if peers can do video transcoding twice. Fig 2 illustrates the P2P distribution. Due to the space limit, we skip the detailed proof here.

E. P2P Distribution with Layered Video Coding

P2P sharing between peers downloading video at different rates can be also enabled if layered video coding is employed by the server. Specifically, the server encodes the video into multiple layers with nested decoding dependency. A base layer has to be received by all peers. An enhancement layer k can be decoded iff all layers up to k are received. Ideally, if peer A's video download rate is higher than peer B, then A has all layers B needs. A and B share with each other their common layers without the need of transcoding.

When layered video coding is employed, the server needs to determine the total number of layers and the rate of each layer to generate. We must determine which layers each peer should download. We can show that when using layered video coding, if coding overhead is negligible, the optimal video allocation in (5) can also be achieved.

Theorem 3: The optimal rate in (5) can be achieved if the server generates K^* video layers, and all peers subscribing to the same layer share video with each other.

Proof: The server generates K^* video layers: the rate ξ_1 for the 1st video layer is $\xi_1 = W^*$, the rate ξ_2 for the 2nd video layer is $\xi_2 = (1-t)u_{K^*-1} - W^*$ and the rate for the i-th($i > 2$) video layer is $\xi_i = (1-t)(u_{K^*+1-i} - u_{K^*+2-i})$. The set of peers receiving the j-th video layer is $S_j^{'} = S_1 \cup S_2 \cup \cdots \cup S_{K^*+1-j}$.

For the i-th $(1 < i \leq K^*)$ video layer, the server transmits one copy of that layer to peers in set $S_i^{'}$. All peers in $S_i^{'}$ form a P2P swarm. They need $\xi_i(|S_i^{'}|-1)$ more bandwidth to distribute layer i to all peers in the swarm. After allocating all the i-th $(1 < i \leq K^*)$ video layers, we denote the aggregate peer residual bandwidth be u_p^{Rest}. Then,

$$\begin{aligned} u_p^{Rest} =& \sum_{i=1}^{N} n_i u_i - \sum_{i=2}^{K^*} \xi_i(|S_i^{'}|-1) \\ =& (1-t)u_1 + \sum_{i=1}^{K^*-1} t n_i u_i + \sum_{i=K^*}^{N} n_i u_i + \\ & W^*(\sum_{i=1}^{K^*-1} n_i - 1) \end{aligned}$$

And the residual bandwidth on the server is $u_s^{Rest} = u_s - \sum_{i=2}^{K^*} \xi_i = u_s - (1-t)u_1 + W^*$. It is straightforward to check that $u_p^{Rest} + u_s^{Rest} = (\sum_{i=1}^{N} n_i)W^* = W^*|S_1^{'}|$. Hence, the optimal solution (5) can be achieved with K^* video layers. ■

F. Helper-Based P2P Distribution

In practice, video transcoding on peers may impose too great of a computational burden on peers or altogether impractical. Moreover, layered encoding may suffer from low coding efficiency. In this Section, we study P2P distribution when neither transcoding nor layered encoding is feasible. In this case, in order to optimize the average video quality while satisfying the entitled video-rate constraints, it may be necessary for certain peers to act as "helpers," that is, to download video versions that they are not watching and redistribute those versions to other peers.

Let G be the set of peers viewing a particular version. As shown in [16], [17], with helpers, the maximal achievable video rate r^* for the peers in G is:

$$r^* = \frac{B^{(W)} + B^{(H)}}{|G|} - \frac{B^{(H)}}{|G|^2}, \tag{7}$$

where $B^{(W)}$ is the aggregate upload bandwidth of the peers in G plus the amount of server bandwidth allocated to the version and $B^{(H)}$ is bandwidth used by helpers to help the peers in G. The last term in (7) reflects the *helper-overhead*, which is the upload bandwidth (e.g., from the server's allocation to helpers) used to send video content (from the version) to the helpers, so that they in turn can redistribute (and amplify) the video to the viewers in G. Notably, helper-overhead decrease as the number of viewers increases.

Unfortunately, with the helper-overhead, it is no longer possible to exactly achieve the optimal rate for **OPT 1**. Instead, we develop a heuristic algorithm for helper-based P2P distribution scheme, then study how far away it is from the optimal. In the water-filling algorithm in Algorithm 1, bandwidth-rich peers only get their entitled rates, and all the bandwidth-poor peers get the same rate W^*, which is higher than their entitled rates. Thus, we propose a heuristic Algorithm 2 for the helper-based case. In that algorithm, we first use the water-filling approach in Algorithm 1 to get the base rate W^* without considering the helper overhead. Fig 1 illustrates the distribution design with transcoding or layered coding. When transcoding and layered coding are not available, we can use peers from S_1 to S_{K^*-1} as helpers for peers in the base class. Due to the helper overhead, W^* is not achievable in the base class. To circumvent this, we first let the server reserve bandwidth of W^* to feed the base video to all helpers.[1] Now we run the water-filling algorithm again with the server bandwidth of $u_s - W^*$. We get a lower base rate $W^{'}$ and the corresponding P2P distribution design as illustrated in Fig 1. We treat those peers with video rates higher than $W^{'}$ as the helpers for peers at rate $W^{'}$, and use the reserved server bandwidth of $W^* > W^{'}$ to feed the base rate video $W^{'}$ to all helpers, then all peers in the base level will get rate of $W^{'}$.

Algorithm 2 Video Version Allocation For Continuous Version under Helper-based condition

1: Water-Filling-Continuous($\mathcal{U}, \mathcal{N}, u_s, N$) to generate W^*, K^*
2: Water-Filling-Continuous($\mathcal{U}, \mathcal{N}, u_s - W^*, N$) to generate $\mathcal{R}, W^{'}, K^{'}$

Theorem 4: From Algorithm 2, we get $W^*, K^*, W^{'}, K^{'}$ respectively. Then, the utility gap between the result obtained from heuristic Algorithm 2 and the optimal solution is smaller than $\frac{W^*}{W^{'}}$.

Proof: When considering helper overhead, the optimal capacity for this problem could not exceed the optimal solution for OPT I. Thus, we can use the utility gap between (5) and the result of Algorithm 2 as the upper-bound. The result of Algorithm 2 can be expressed as

$$r_{ij}^{'} = \begin{cases} (1-t)u_i, & \text{if} \quad (1-t)u_i > W^{'} \\ W^{'}, & \text{if} \quad (1-t)u_i \leq W^{'} \end{cases} \tag{8}$$

If we assume that $r_i = (1-t)u_i, \forall i \leq (K^{'}-1)$, then $W^{'}$ can be expressed as

$$W^{'} = \frac{\sum_{i=1}^{N} n_i u_i + u_s - W^* - (1-t)\sum_{i=1}^{(K^{'}-1)} n_i u_i}{\sum_{i=K^{'}}^{N} n_i} \tag{9}$$

Compared with (5), we have $W^{'} < W^*, K^{'} \geq K^*$. Then, the utility gap upper-bound can be expressed as:

$$\begin{aligned} & \sum_{i=1}^{N}\sum_{j=1}^{n_i}(\log(r_{ij}^*) - \log(r_{ij}^{'})) \\ =& \sum_{i=1}^{K^*} n_i(\log((1-t)u_i) - \log((1-t)u_i)) + \sum_{i=K^*}^{K^{'}} n_i(\log(W^*) \\ & - \log((1-t)u_i)) + \sum_{i=K^{'}}^{N} n_i(\log(W^*) - \log(W^{'})) \\ <& \sum_{i=K^*}^{N} n_i \frac{W^* - r_{ij}^{'}}{W^{'}} < \frac{W^*}{W^{'}} \end{aligned} \tag{10}$$

[1]The server does not have to send the whole base video to each helper. In fact, each helper only needs to download a very small sub-stream of the base video so that it can upload it to all peers at the base level by using up its residual bandwidth. As shown in [16], the total bandwidth a server needs to feed all the helpers is at most the base video rate.

TABLE I: Peer Uplink Capacity Setting

Types	Uplink Capacity	Number
Server	4000 kbps	1
Peer1	1500 kbps	100
Peer2	1000 kbps	200
Peer3	500 kbps	300

In (10), the first inequality is due to the concavity of $\log()$ function and the fact that $r'_{ij} \geq W'$, the second inequality uses the fact that $\sum_{i=K^*}^{N}(n_i W^* - n_i r'_{ij}) = W^*$, which is just the bandwidth we reserve to deal with overhead. ■

Note that the upper bound of $\frac{W^*}{W'}$ is for the aggregate utility among all peers. When the peer number is large, the per-peer utility in the helper-based distribution is almost the same as the optimal case.

III. Numerical Study for Taxation

We conduct numerical case studies to verify our analysis and further illustrate design trade-offs in P2P adaptive streaming. We use AMPL/CPLEX package to solve OPT I, OPT II. For the various heuristic algorithms, we use MATLAB to realize them. Layered coding incurs coding rate overhead. When employing SVC, an r-d optimized multi-layer encoder [18] encodes 10% more compared to the single-layer H.264/AVC coding. In our simulation, we use 0.1 as the overhead of employing layered video, although this number is much higher in reality [19].

A. Impact of Taxation Ratio

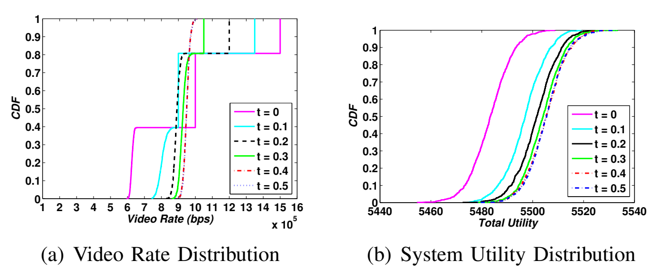
We study the impact of taxation ratio. To isolate the effects of discrete video versions and helper overhead, we use continuous transcoding here. In the simulation, there are three types of peers as listed in Table I. Under different taxation ratios, Fig. 3 shows the distribution of video rates of peers from different classes. The system-wide utility is also plotted. At low taxation ratio, video rates are quite diverse. It gives strong incentive for the bandwidth-rich peers, but the bandwidth-poor peers suffer bad quality, as a result the system-wide utility is low. As the taxation ratio increases, the video rate gap between classes decreases. Bandwidth-poor peers are helped a lot by the taxed bandwidth pool. The system-wide utility quickly approaches the optimal at tax rate around 0.38, where heterogenous peers turn to watch video at more similar video rates. It demonstrates that taxation can be used to tradeoff between system-wide utility and incentive for individual peers.

B. Capacity Comparison of Three Distribution Designs

We compare the capacity of transcoding, layered coding, and helper-based distribution when varying the number of peers in the system. The taxation ratio is set to be 0.02 and there are three classes of peers with uplink capacity listed in Table I. Initially each class only has 4 peers. Then, we gradually add 10 more peers to each class at each round. Fig. 4(a) plots video rate distributions of three classes of peers under different distribution designs. As shown in Theorem 2, transcoding can achieve the optimal rate. Due to the layered coding overhead, the rate achieved by layered coding is now a fraction lower than that of transcoding regardless of peer numbers. Helper-based distribution incurs helper-overhead. The achieved rates are slightly lower than the optimal transcoding case. Even when peer number is very small, helper-based distribution achieves higher rate than layered-coding. When peer number is large, helper-overhead is almost negligible, and the achieved rates are almost the same as the transcoding case. Fig. 4(b) compares the per-peer utility gap from the optimal for helper-based distribution and layered coding. As expected, layered coding leads to a constant utility loss, while the utility loss for helper-based distribution quickly converges to zero as the number of peers increases. Although video transcoding always gives us the optimal solution, it incurs computation overhead on peers. Since P2P streaming systems usually have a large number of peers, helper-based distribution is a promising simple approach. We will just use the helper-based approach in the following simulations.

C. Results under Bandwidth Variations

To investigate the impact of peer bandwidth variations, we first simulate a system with only one peer class, and without peer churn. We assume that there are 4 upload bandwidth levels: $1,500$ kbps, $1,000$ kbps, 600 kbps, 400 kbps, and the server's upload bandwidth is $15,000$kbps. Bandwidth variations on all peers follow the same switching probability matrix $\mathbf{P} = \begin{pmatrix} 0.4 & 0.45 & 0.1 & 0.05 \\ 0.2 & 0.5 & 0.2 & 0.1 \\ 0.1 & 0.4 & 0.4 & 0.1 \\ 0.05 & 0.1 & 0.45 & 0.4 \end{pmatrix}$, the mean sojourn time at level i is $1/\mu_i = 1 (1 \leq i \leq 4)$. Then, we change the taxation ratio. At each taxation ratio, we run Monte Carlo simulations with $10,000$ sample points. Fig. 5(a) and Fig. 5(b) show video rate and system utility distributions under different taxation ratios. Since we use continuous model here, the base video rate can take any value, and has continuous probability distribution, the entitled rates only take discrete values. As the taxation ratio increases, video rates of all peers turn to be more similar and the whole system utility is enhanced. When the taxation ratio is large enough, more taxation ratio does not help any more, like the curves of $t = 0.4$ and $t = 0.5$ are overlapped in the figure. Since we are dealing with a single class of peers, their upload bandwidth assume a homogeneous distribution. A high taxation ratio is justifiable. In this case, taxation achieves the temporal multiplexing gain. It allows a peer with temporary bandwidth dips continue to get stable video download from other peers.



(a) Video Rate Distribution (b) System Utility Distribution

Fig. 5: Single Class without Peer Churn

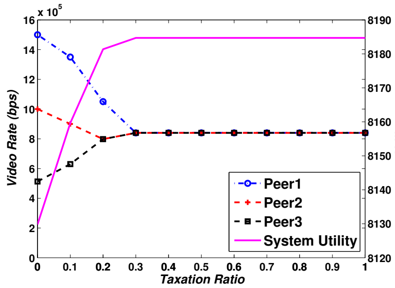
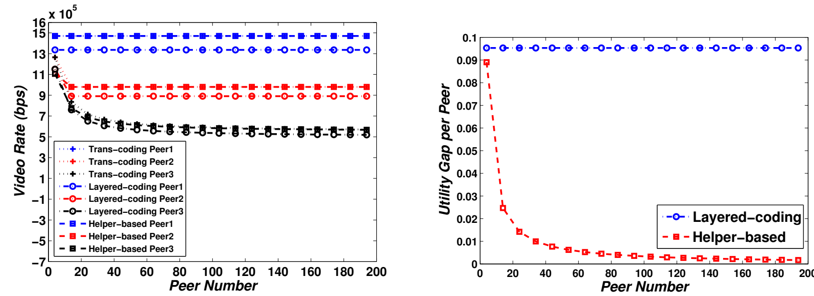


Fig. 3: Video Rates and System Utilities under Different Taxations

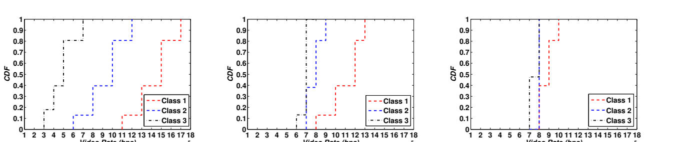


(a) Video Rate Distribution

(b) Utility Gap from the Optimum

Fig. 4: Comparison of Transcoding, Layered Coding, and Helper-based Distribution

Now, we study a system with multiple classes. In the simulation, there are three classes. In each class, peer's uplink capacity has four levels like in Table II, and the server's upload bandwidth is $15,000$ kbps. We assume that system provides 18 discrete video versions with rates ranging from 100 kbps to $1,800$ kbps, and the rate difference between the adjacent video version is exactly 100 kbps. For each class, we reuse the switching probability matrix $\mathbf{P}$ in the previous section to control the switching between its four levels [2], and the mean sojourn time at level i in class c is still $1/\mu_i^{(c)} = 1 (1 \leq i \leq 4, 1 \leq c \leq 3)$. Then, we also change the taxation ratio and run Monte Carlo simulation with $10,000$ samples for each ratio. Fig. 6(a), Fig. 6(b) and Fig. 6(c) plot the video rate distributions when the taxation ratio $t = 0, 0.2, 0.5$ respectively. Table III shows the mean and standard deviation (SD) of the video rate for each class and the system utility under different taxation ratios (TR). We can see that with higher taxation ratio, the system utility becomes larger and its variance becomes smaller. For all classes, a higher taxation ratio makes video rate variation smaller, which is beneficial for all peers. Meanwhile, higher taxation ratio also has the effect that there would be much smaller difference between strong peers' video rates and week peers' video rates. Thus, an appropriate taxation ratio should be determined by jointly considering video quality variation, system wide efficiency, and incentive to individual peers.



(a) t=0

(b) t=0.2

(c) t=0.5

Fig. 6: Video Rates for Multiple Classes without Peer Churn

TABLE II: Peer Class Setting

Peer Class(c)	Upload Capacity (kbps)				Peer Number
	Level 1	Level 2	Level 3	Level 4	
1	1700	1500	1300	1100	100
2	1200	1000	800	600	200
3	700	500	300	100	300

[2]Since different classes have different four bandwidth levels, the actual bandwidth variation processes between different classes are different

TABLE III: Rate (Mbps) and Utility Variation for Multiple Classes without Peer Churn

TR	Class 1		Class 2		Class 3		Utility	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD
0	1.43	0.19	0.93	0.19	0.44	0.17	8037.9	10.05
0.1	1.23	0.19	0.83	0.14	0.58	0.04	8104.5	7.11
0.2	1.11	0.16	0.78	0.07	0.65	0.05	8119.2	6.56
0.3	0.97	0.09	0.77	0.04	0.70	0.03	8128.1	5.96
0.4	0.88	0.07	0.79	0.03	0.72	0.04	8130.5	6.04
0.5	0.80	0.00	0.80	0.01	0.74	0.05	8132.3	6.06
0.6	0.80	0	0.80	0.01	0.74	0.05	8132.5	5.89

IV. Cooperative Games for P2P Streaming

A. *Cost-sharing Mechanism for Single Version P2P Streaming*

In P2P video streaming, cooperation between peers can give gains in both content and bandwidth. In this section, we apply cooperative game theory to study how to share the cost of a video streaming service among participating peers. One unique feature of this P2P streaming game is that a peer's participation in the service set can potentially reduce the system-wide bandwidth cost.

We start with the *basic P2P streaming cooperative game* where only one video version is available, and where each peer bids to see the video. We will extend this basic game to P2P streaming with multiple versions in the next subsection. We normalize the video rate to be 1 (and normalize the link bandwidths accordingly). We denote by $\mathcal{A}$ the set of peers interested in this video. Without loss of generality, we assume each peer in $\mathcal{A}$ has download bandwidth larger than the video rate. For peer i, its upload bandwidth is denoted by u_i and its utility for watching the video is v_i.

1) Service Cost: To serve a set S of peers, the server's cost can be calculated by

$$c(S) = c_c(S) + c_d(S), \tag{11}$$

where $c_c(S)$ is the total content cost and $c_d(S)$ is the total distribution cost for streaming the video to all peers in S. We assume the server content cost $c_c(S)$ is an increasing concave function of $|S|$. In other words, the server incurs decreasing marginal content cost for serving additional peers. Without P2P sharing, the distribution cost $c_d(S)$ is $w|S|$, where w is the cost per unit server bandwidth. When peer-assisted distribution is employed, video sharing between peers can significantly offload the server, thereby reducing the server distribution cost. In P2P streaming, the upload contribution of a peer is determined by its upload bandwidth as well as the efficiency of the P2P streaming algorithm in utilizing peers' upload bandwidth. As demonstrated in [13], if all peers are fully connected, all peers' upload capacity can be fully utilized to achieve high video streaming rate. In real P2P systems, however, a peer in a larger swarm has a better chance to find neighbors to exchange content, and it is also more likely to find nearby peers to achieve high P2P data-transfer throughput. As a result, the peer upload capacity utilization is less than 1 but normally increases as the number of peers increases [20], [7]. For our P2P streaming cooperative game, we model the average upload bandwidth utilization as an increasing concave function $\rho(|S|)$ of the streaming swarm size, with $\rho(|S|)$ approaching 1 when the number of peers is large. Taking into account the peers' upload contribution, the server's upload bandwidth contribution is given by:

$$U_d(S) = 1 + \left[|S| - 1 - \rho(|S|) \min(|S| - 1, \sum_{i \in S} u_i) \right], \quad (12)$$

where the first term is the unavoidable server bandwidth cost to stream at least one full copy of the video to the swarm, the second term is the additional bandwidth the server has to provide on top of peer upload contribution to ensure all peers receive the video stream.

2) Cost-sharing Mechanism for $u_i \leq 1$: When $u_i \leq 1$ for all $i \in \mathcal{A}$, so that each peer's upload bandwidth is no more than the video rate, the server distribution cost can be approximated by

$$\hat{U}_d(S) = |S| - \rho(|S|) \sum_{i \in S} u_i = \sum_{i \in S} (1 - \rho(|S|) u_i), \quad (13)$$

with approximation error bound $-1 \leq \hat{U}_d(S) - U_d(S) \leq 0$. Using this minor approximation, the server cost becomes

$$c(S) = c_c(S) + w \sum_{i \in S} (1 - \rho(|S|) u_i).$$

Therefoe, a natural cost-sharing scheme for peer i with upload capacity u_i is:

$$\textbf{Scheme-I:} \quad p(i, u_i, S) \triangleq \frac{c_c(|S|)}{|S|} + w(1 - \rho(|S|) u_i),$$

where the first term is the equal share of the content cost, the second term is the distribution bandwidth cost discounted by each peer's upload contribution.

Theorem 5: **Scheme-I** is *cross-monotone.*

Proof: For all $S, T \subseteq N$ and $i \in S$, due to the concavity of $c_c(\cdot)$, $\frac{c_c(S)}{|S|} \geq \frac{c_c(S \cup T)}{|S \cup T|}$. Also since $\rho(\cdot)$ is an increasing function, we have $\rho(|S|) \leq \rho(|S \cup T|)$. Therefore, we have $p(i, u_i, S) \geq p(i, u_i, S \cup T)$, which satisfies the definition of *cross-monotone.* ■

As part of the bidding, if each peer reveals its true upload bandwidth (but not necessarily its true utility), then we can directly apply the theory in [21]. In particular, it follows from Theorem 5 and the theory in [21] that under the cost-sharing Scheme-I and the mechanism of Algorithm 1, every peer will bid its true utility value, v_i, and Algorithm 1 returns the maximal service set $S^*(\mathbf{v})$.

But it is more natural to suppose that each peer may not only lie about its utility v_i but also lie about its upload bandwidth u_i. In particular a peer (or a group of colluding peers) may bid a higher upload bandwidth than it actually has in order to increase its chances of being included in the swarm. In this case, the bidding game is different from the traditional game [21]. Recall that in the traditional game, the total cost $c(S)$ and the cost for each agent $\xi(i, S)$ is determined only by the cooperation set S and not the bids. But now, since the cost shares $p(i, u_i, S)$ depend on the bids (the upload rates), the classical cooperative game theory no longer directly applies. Algorithm 3 develops a cost-sharing mechanism $M(p)$ out of a cost-sharing scheme p for a P2P cooperative game.

Algorithm 3 P2P Cost Sharing Mechanism $M(p)$

Each peer bids its utility and upload bandwidth $\langle v_i, u_i \rangle$;
Initialize $Q \leftarrow \mathcal{A}$.
Repeat
    Let $Q \leftarrow \{i \in Q : v_i \geq p(i, u_i, Q)\}$.
Until for all $i \in Q, v_i \geq p(i, u_i, Q)$.
Return $S = Q$ and $p_i = p(i, u_i, Q)$, for all $i \in Q$, $p_i = 0$, for all $i \notin Q$.

To summarize the P2P cooperative game, the following steps are taken:

1) We design a cost structure $p(i, \mathbf{u}, S)$ which in general may depend on the vector of upload bandwidths $\mathbf{u} = \{u_i, i \in \mathcal{A}\}$.
2) Each peer i bids $\langle v_i', u_i' \rangle$, which may or may not be truthful.
3) After the bidding is completed, Algorithm 3 is used to determine the set of participants S' based on the $\langle v_i', u_i' \rangle$ values.
4) The P2P streaming is carried out over the set S'. Peers are charged according to their actual upload bandwidths rather than their bidded bandwidths. The profit for peer i is given by $v_i - p(i, \mathbf{u}, S')$ when $i \in S'$ and is zero otherwise.

Theorem 6: In a P2P cooperative game, if the cost-sharing scheme p is cross-monotone and the cost sharing of each peer $p(i, \mathbf{u}, S)$ only depends on $\mathbf{u}$ through its own upload contribution u_i, then the cost-sharing mechanism $M(p)$ defined in Algorithm 3 is *group-strategyproof.*

Proof: Let S be the service set returned by $M(p)$ under truthful bids. Since p is a cross-monotone cost-sharing scheme, S is the unique maximal set satisfying $v_i - p(i, u_i, S) \geq 0$, $\forall i \in S$.

We prove by contradiction by supposing there is strategic coalition with associated bids. Specificlly, suppose there is a group of colluding peers C and associated bids $\langle v_i', u_i' \rangle$, $i \in C$, resulting in service set S' returned by $M(p)$, with the following properties: (a) $v_i 1(i \in S') - p(i, u_i, S') \geq v_i 1(i \in S) - p(i, u_i, S)$ holds for every $i \in C$, and (b) there is at least one $i_0 \in C$ such that $v_{i_0} 1(i_0 \in S') - p(i_0, u_{i_0}, S') > v_{i_0} 1(i_0 \in S) - p(i_0, u_{i_0}, S)$.

Consider a peer $i \in S' - C$. Such a peer bids truthfully, and according to the definition of $M(p)$, we immediately have $v_i - p(i, u_i, S') \geq 0$. Now consider a colluding peer $i \in S' \cap C$. By (a) we have $v_i - p(i, u_i, S') \geq v_i 1(i \in S) - p(i, u_i, S)$. If $i \notin S$, then it directly follows $v_i - p(i, u_i, S') \geq 0$. If $i \in S$ then it follows $v_i - p(i, u_i, S') \geq v_i - p(i, u_i, S) \geq 0$, where this last inequality follows from the property of S stated in the first paragraph of this proof. Combining these cases, we have $v_i - p(i, u_i, S') \geq 0$, $\forall i \in S'$. Since S is the maximal service set under truthful bid, then $S' \subseteq S$.

For (b) to hold, it is easy to see that $i \in S'$ and thus $i \in S$. Thus, $v_{i_0} - p(i_0, u_{i_0}, S') > v_{i_0} - p(i_0, u_{i_0}, S)$. Then $p(i_0, u_{i_0}, S') < p(i_0, u_{i_0}, S)$, which contradicts $S' \subseteq S$ and p is cross-monotone. Thus, $M(p)$ is group-strategyproof. ∎

Corollary 6.1: M(Scheme-I) is group-strategyproof.

3) Cost-sharing Scheme for the General Case: For the case where there is at least one $i \in \mathcal{A}$, with $u_i > 1$, the server cost approximation in (13) can introduce considerable error for some peer combinations. For this case, we need to design a new cost-sharing scheme. Motivated by Scheme-I, it is natural to consider **Scheme-II** to reward a peer by its relative upload contribution:

$$p(i, u_i, S) \triangleq \frac{c_c(S)}{|S|} + w(1 - \min(1, \frac{|S|-1}{\sum_{i \in S} u_i}) u_i \rho(|S|)),$$

where the first term is equal content cost share, and the second term is the adjusted peer upload contribution. Note that in Scheme-II, the costs $p(i, \mathbf{u}, S)$ now depend on $\mathbf{u}$ and not just on u_i. It is straight-forward to establish the following result:

Theorem 7: Scheme-II is *budget-balanced*.

Although Scheme-II is very natural and budget-balanced, it is neither cross-monotone nor group-strategyproof.

Theorem 8: Scheme-II is not cross-monotone.

Proof: We provide a counter example. Suppose we have four peers, with upload bandwidth of $0, 0, 2, 2$ respectively. When only the first three peers form a coalition, $S_3 = \{1, 2, 3\}$, the price for peer 3 is:

$$\begin{aligned} p(3, 2, S_3) &= \frac{c_c(S_3)}{|S_3|} + w(1 - \min(1, \frac{|S_3|-1}{\sum_{i \in S_3} u_i}) u_i \rho(|S_3|)) \\ &= \frac{c_c(S_3)}{3} + w(1 - \min(1, \frac{2}{2}) 2\rho(3)) \\ &= \frac{c_c(S_3)}{3} + w(1 - 2\rho(3)) \end{aligned} \tag{14}$$

When all four peers form a coalition, $S_4 = \{1, 2, 3, 4\}$, the price for peer 3 is:

$$\begin{aligned} p(3, 2, S_4) &= \frac{c_c(S_4)}{|S_4|} + w(1 - \min(1, \frac{|S_4|-1}{\sum_{i \in S_4} u_i}) u_i \rho(|S_4|)) \\ &= \frac{c_c(S_4)}{4} + w(1 - \min(1, \frac{3}{4}) 2\rho(4)) \\ &= \frac{c_c(S_4)}{4} + w(1 - \frac{3}{2}\rho(4)) \end{aligned} \tag{15}$$

Comparing (14) with (15), if $\frac{c_c(S_3)}{3} - \frac{c_c(S_4)}{4} < \frac{w\rho(3)}{2}$, establishing that Scheme-II is not cross-monotone. ∎

Theorem 9: Let M(Scheme-II) be the mechanism obtained from Algorithm 3. M(Scheme-II) is not *group-strategyproof.*

Proof: We prove this using the example in the proof of Theorem 6. Set the utility values large enough for all four peers so that if they all bid truthfully, they will all be included in the service set, that is, set $v_i \geq p(i, u_i, S_4), i = 1, 2, 3, 4$. The cost shares for peer 3 and 4 are the same, as calculated in (15):

$$p(3, 2, S_4) = p(4, 2, S_4) = \frac{c_c(S_4)}{4} + w(1 - \frac{3}{2}\rho(4)).$$

Now suppose peer 3 announces that its upload bandwidth is 10 instead of 2. The cost shares for peers 3 and 4 become

$$\begin{aligned} p'(3, 10, S_4) &= \frac{c_c(S_4)}{4} + w(1 - \min(1, \frac{2}{12}) 10\rho(4)) \\ &= \frac{c_c(S_4)}{4} + w(1 - \frac{5}{3}\rho(4)) \end{aligned} \tag{16}$$

and

$$\begin{aligned} p'(4, 2, S_4) &= \frac{c_c(S_4)}{4} + w(1 - \min(1, \frac{2}{12}) 2\rho(4)) \\ &= \frac{c_c(S_4)}{4} + w(1 - \frac{1}{3}\rho(4)) \end{aligned} \tag{17}$$

respectively. Thus, if we set peer 4's utility value to $v_4 = \frac{c_c(S_4)}{4} + w(1 - \frac{1}{2}\rho(4)) < p'(4, 2, S_4)$, and set the other three peers to have large values, then the resulting coalition set for $M(p')$ becomes $\{1, 2, 3\}$. Since $u_1 = u_2 = 0$, it is easy to check that the cost share allocated to peer 3 is $p'(3, 10, S_3) = p(3, 2, S_3) < p(3, 2, S_4)$ as shown in Theorem 8. Thus, because peer 3 gains benefit by lying about its upload bandwidth, M(Scheme-II) is not *group-strategyproof.* ∎

Based on the above cost-sharing scheme, now consider **Scheme-III**:

$$p(i, u_i, S) = \frac{c_c(S)}{|S|} + w(1 - \min(1, \frac{|S|-1}{|S|\bar{u}}) u_i \rho(|S|)),$$

where $\bar{u}$ denotes the average upload bandwidth of peers who are watching the video. The value can be determined by the server using historical data. It is straightforward to establish the following three results.

Theorem 10: Scheme-III is *budget-balanced if $\bar{u}$ is indeed the average upload bandwidth of peers in S.*

Theorem 11: Scheme-III is *cross-monotone.*

Corollary 11.1: Let M(Scheme-III) be the mechanism obtained from Algorithm 3. M(Scheme-III) is group strategy-proof.

B. Multi-Version P2P Streaming

Now suppose there are J video versions with decreasing video rates $r_1 > r_2 > \cdots > r_J$. Similar to the single-version case, there is a set $\mathcal{A}$ of peers. Peer i has upload capacity of u_i, and video utility v_i. We assume each peer i first and foremost wants to watch the highest video rate allowed by its utility budget v_i. After being assigned to the highest possible video rate, it desires to pay the lowest possible price without changing the assigned rate.

Let S_j be the set of peers watching version j. We set the pricing scheme for peer i watching video version j based on **Scheme-III:**

$$p_j(i, u_i, S_j) = c_j(|S_j|) + w(r_j - \min(1, \frac{(|S_j| - 1)r_j}{|S_i|\bar{u}_j})u_i\rho(|S_j|)), \tag{18}$$

where $c_j(|S_j|) \triangleq c_j^{(c)}(|S_j|)/|S_j|$ is the content price for version j, $\bar{u}_j$ is the expected upload bandwidth of peers watching version j.

Algorithm 4 Top-Down Multi-Round Bidding for Multi-version P2P Streaming

Each peer bids its utility and upload bandwidth $\langle v_i, u_i \rangle$;
Initialize $\mathcal{S} \leftarrow \mathcal{A}$.
for $j = 1, ..., J$ **do**
    $T = S$
    REPEAT
        Let $S \leftarrow \{i \in S : v_i \geq p_j(i, u_i, S)\}$.
    Until for all $i \in S, v_i \geq p_j(i, u_i, S)$.
    $S_j = S$; $R_i = r_j$, for all $i \in S_j$;
    $S = T - S_j$;
    If $S = \emptyset$ then Stop.
end for
If $S \neq \emptyset$ then $R_i = 0$ for all $i \in S$;

Algorithm 4 selects video rate R_i for each peer i. It starts from the highest video rate. Once a peer is selected to watch a particular video rate, it is not considered for the lower video rates. When each peer has been allocated to some video version, or all video versions have been considered, the algorithm ends.

V. Conclusion

In this paper, we studied the incentive issues in P2P adaptive streaming. We demonstrated that incentive-compatible sharing between peers watching different video versions can be enabled through taxation. Simple helper-based P2P distribution can achieve close-to-optimal efficiency without video transcoding and layered video coding. We also showed that pricing mechanisms derived from cooperative game theory can be used to foster and sustain P2P sharing in adaptive streaming.

References

[1] S. Akhshabi, A. C. Begen, and C. Dovrolis, "An experimental evaluation of rate-adaptation algorithms in adaptive streaming over http," in *Proceedings of ACM conference on Multimedia systems*, 2011.

[2] V. K. Adhikari, Y. Guo, F. Hao, M. Varvello, V. Hilt, M. Steiner, and Z.-L. Zhang, "Unreeling netflix: Understanding and improving multi-cdn movie delivery," in *Proceedings of IEEE INFOCOM*, 2012.

[3] J. F. Kurose and K. Ross, *Computer Networking: A Top-Down Approach Featuring the Internet*, 6th ed. Addison-Wesley, 2012.

[4] Y. Chu, S. Rao, and H. Zhang, "A case for end system multicast," in *Proceedings of ACM SIGMETRICS*, 2000.

[5] X. Zhang, J. Liu, B. Li, and P. Yum, "Coolstreaming/DONet: A data-driven overlay network for efficient live media streaming," in *IEEE INFOCOM*, 2005.

[6] Mea Wang and Baochun Li, "R2: Random push with random network coding in live peer-to-peer streaming," *IEEE J. Select. Areas Commun.*, vol. 25, pp. 1655–1666, Dec. 2007.

[7] X. Hei, C. Liang, J. Liang, Y. Liu, and K. W. Ross, "A Measurement Study of a Large-Scale P2P IPTV System," *IEEE Transactions on Multimedia*, December 2007.

[8] 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 ACM SIGCOMM*, 2008.

[9] Y. hua Chu, J. Chuang, and H. Zhang, "A case for taxation in peer-to-peer streaming broadcast," in *ACM SIGCOMM workshop on Practice and theory of incentives in networked systems*, 2004.

[10] Y.-W. Sung, M. Bishop, and S. Rao, "Enabling Contribution Awareness in an Overlay Broadcasting System," in *SIGCOMM*, 2006.

[11] H. Hu, Y. Guo, and Y. Liu, "Peer-to-Peer Streaming of Layered Video: efficiency, fairness and incentive," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 21, no. 8, August 2011.

[12] J. Li, P. A. Chou, and C. Zhang, "Mutualcast: An Efficient Mechanism for Content Distribution in a P2P Network," in *Sigcomm Asia Workshop*, 2005.

[13] R. Kumar, Y. Liu, and K. Ross, "Stochastic Fluid Theory for P2P Streaming Systems," in *Proc. IEEE INFOCOM*, May 2007.

[14] Y. Guo, C. Liang, and Y. Liu, "Adaptive Queue-based Chunk Scheduling for P2P Live Streaming," in *Proceedings of IFIP Networking*, May 2008.

[15] L. Massoulie, A. Twigg, C. Gkantsidis, and P. Rodriguez, "Randomized decentralized broadcasting algorithms," in *Proceedings of IEEE INFOCOM*, 2007.

[16] D. Wu, C. Liang, Y. Liu, and K. W. Ross, "Redesigning Multi-Channel P2P Video Systems with View-Upload Decoupling," in *Computer Networks Jounral, Elsevier*, 2010, accepted for publication.

[17] M. Chen, M. Ponec, S. Sengupta, J. Li, and P. A. Chou, "Utility maximization in peer-to-peer systems," in *Proceedings of ACM SIGMETRICS*, vol. 36, Jun. 2008.

[18] H. Schwarz and T. Wiegand, "R-D Optimized Multi-Layer Encoder Control for SVC," in *Proceedings of the International Conference on Image Processing)*, 2007.

[19] M. Wien, H. Schwarz, and T. Oelbaum, "Performance Analysis of SVC," *IEEE Trans. Circuit and System for Video Technology*, vol. 17, pp. 1194–1203, Sep. 2007.

[20] D. Qiu and R. Srikant, "Modeling and performance analysis of bittorrent-like peer-to-peer networks," in *Proceedings of ACM SIGCOMM*, 2004.

[21] N. Nisan, T. Roughgarden, E. Tardos, and V. V. Vazirani, *Algorithmic Game Theory*. New York, NY, USA: Cambridge University Press, 2007.