

# Capacity Analysis of Peer-to-Peer Adaptive Streaming

Yang Xu<sup>1</sup>, Yong Liu<sup>1</sup> and Keith Ross<sup>2</sup>

<sup>1</sup>Electrical & Computer Engineering, Polytechnic Institute of New York University

<sup>2</sup>Computer Science & Engineering, Polytechnic Institute of New York University

**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 paper, we analytically study the potential of using P2P in conjunction with adaptive streaming. We first study the capacity of P2P adaptive streaming by developing utility maximization models that take into account peer heterogeneity, taxation-based incentives, multi-version videos at discrete rates. We further develop stochastic models to study the performance of P2P adaptive streaming in face of bandwidth variations and peer churn. Through analysis and simulations, we demonstrate that incentive-compatible video sharing between peers can be easily achieved with simple video coding and distribution designs. P2P adaptive streaming not only significantly reduces the load on the servers, but also improves the stability of user-perceived video quality in the face of dynamic bandwidth changes.

## 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 do 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). In this paper, we employ *taxation* to strike a balance between fairness and incentives.

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.

In this paper, for live video streaming, we analytically study the capacity of P2P adaptive streaming by developing utility maximization models that take into account peer heterogeneity, taxation-based incentives, and different video coding choices. We further develop stochastic models to study the performance

of P2P adaptive streaming in face of peer bandwidth variations. Through analysis and simulations, we demonstrate that

- Taxation serves as a simple, yet powerful, tool to strike the desired balance between social welfare and individual welfare in P2P adaptive streaming.
- Optimal P2P sharing between peers watching different video versions can be achieved if either peers do simple video transcoding, or the server generates layered video.
- A helper-based P2P sharing design, which does not require peer transcoding and server layered video coding, is almost as efficient as the optimal P2P adaptive streaming with transcoding.
- P2P adaptive streaming not only significantly reduces the load on servers, but also improves the stability of video quality perceived by users in face of time-varying bandwidth changes.

#### A. Related Work

Several research efforts on P2P adaptive streaming have been attempted using scalable video coding. Layered coding is adopted in [9] to fully utilize the available peer upload bandwidth in a tree-based P2P overlay multicast. Authors of [10] proposed a 3-stage chunk scheduling algorithm for mesh-based layered video streaming to achieve high throughput and low video quality jitter. In [11], layered coding is utilized to implement ‘tit-for-tat’ type of incentive in P2P streaming. Taxation-based incentive has been proposed for multiple descriptions coding (MDC) based P2P streaming in [12], [13]. Taxation-based layered P2P streaming is investigated in [14]. Different from previous work, our analysis explore the capacity of generic P2P adaptive streaming systems. We compare the capacity of different design choices, including peer transcoding, server layered coding and helper-based distribution. We also theoretically study the impact of bandwidth variations, which is crucial in adaptive streaming.

## II. CAPACITY WITH CONTINUOUS VIDEO RATES

In this section, we study the capacity of taxation-based P2P adaptive streaming systems with continuous video rates.

#### 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 [12], [13], [14]. 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  $(1-t)u_d$  is called the *entitled rate*, which is a fraction of its own upload contribution, and  $r_d^{(P)}$  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,  $u_1 > u_2 > \dots > u_N$ . 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  $\beta$  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:**

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

**subject to:**

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

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

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  $K^*$ ,  $W^*$  (the meaning of these two terms will be discussed later), and further determines the video rates  $\mathcal{R}$  of all peers. In the algorithm,  $B$  is the taxed bandwidth pool that 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

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

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

- 1: All peers enter a FIFO queue  $Queue_p$  in the increasing order of their upload bandwidth.
- 2: **for** each peer  $j$  in  $S_i$  **do**
- 3:    $r_{ij} = (1-t)u_i$
- 4: **end for**
- 5: 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  $S_w = \phi$ .
- 6: **while** 1 **do**
- 7:   Select peers with the same smallest upload bandwidth  $u_j$  out of  $Queue_p$  and assume set of those peers is  $S_j$ .
- 8:   **if**  $\frac{B+(1-t)|S_j|u_j}{|S_w|+|S_j|} < (1-t)u_j$  **then**
- 9:      $W^* = \frac{B}{|S_w|}$ ,  $K^* = j + 1$ ,  $S_{K^*} = S_w$
- 10:     $r_{ij} = W^*$ ,  $\forall (i, j) \in S_w$
- 11:    **break**
- 12:   **else**
- 13:      $B = B + (1-t)|S_j|u_j$ ,  $S_w = S_w \cup S_j$ ,
- 14:    **end if**
- 15: **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

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

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:

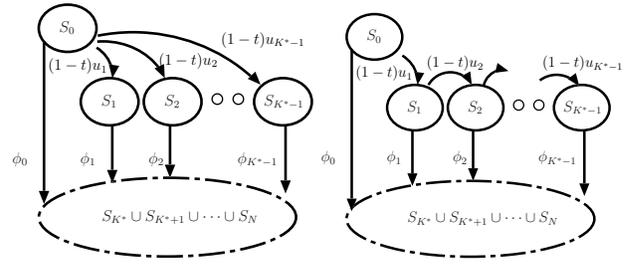
$$\begin{aligned} & \max f(\mathcal{R}) \\ & \text{subject to } A\mathcal{R} \geq b \end{aligned}$$

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

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

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 sharing scheme has to be developed. For single-rate P2P video streaming, it has been shown that in a P2P swarm with  $n$  peers,



**Fig. 1:** Trans-coding Once **Fig. 2:** Trans-coding Twice

the optimal achievable video rate on all peers is

$$r = \min\left\{u_s, \frac{u_s + B_n}{n}\right\}, \quad (6)$$

where  $u_s$  is the server upload bandwidth and  $B_n$  is the aggregate upload bandwidth of all peers [15], [16], [17].

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.

*Theorem 3:* 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.

The above two theorems could be proved by construct the P2P distributions like Fig. 1 and Fig 2 illustrate. Due to the space limit, the detailed proofs are put in our technical report [18].

#### 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 4:** 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  $\xi_1$  for the 1st video layer is  $\xi_1 = W^*$ , the rate  $\xi_2$  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  $S'_j = S_1 \cup S_2 \cup \dots \cup S_{K^*+1-j}$ .

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,

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

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 [19], [20], with helpers, the maximal achievable video rate  $r^*$  for the peers in  $G$  is:

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

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  $|G|$  increases.

Unfortunately, with the helper-overhead, it is no longer possible to exactly achieve the optimal rate for **OPT I**. 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.<sup>1</sup> 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 get the number of video versions  $K^*$ , the base video rate  $W^*$ , and the optimal video rate for each peer  $\mathcal{R}^*$ ;
  - 2: Water-Filling-Continuous( $\mathcal{U}, \mathcal{N}, u_s - W^*, N$ ) to get the number of video versions  $K'$ , the base video rate  $W'$ , and the video rate for each peer  $\mathcal{R}'$ .
- 

**Theorem 5:** The utility gap between the video rate vector  $\mathcal{R}'$  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

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

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

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

<sup>1</sup>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 [19], the total bandwidth a server needs to feed all the helpers is at most the base video rate.

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

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

In (10), the first inequality is due to the concavity of  $\log(\cdot)$  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. CAPACITY WITH DISCRETE VIDEO RATES

The optimal solutions in the previous section assume the server can generate an arbitrary number of video versions or layers at infinitesimal granularity. Real systems only allow finite video versions/layers, and each version/layer will be encoded at one of a finite set of discrete rates. The system capacity bounds obtained in the previous section are therefore upper bounds for real systems. In this section, we extend our baseline continuous analysis to discrete cases.

#### A. Transcoding between Discrete Versions

We start with transcoding between discrete video versions, and assume there are in total  $Q$  video versions. The rate for video version  $q$  is  $v_q$ , and the set of video rates is given by  $V \triangleq \{v_q, 1 \leq q \leq Q\}$ , and we order  $V$  in a decreasing order so that  $v_1 > v_2 > \dots > v_Q$ . Based on **OPT I**, we formulate a utility maximization problem **OPT II** for the discrete case.

**OPT II: objective:**

$$\max \sum_{i=1}^N \sum_{j=1}^{n_i} \log(r_{ij}), \tag{11}$$

**subject to:**

$$r_{ij} = \sum_{q=1}^Q a_{ijq} v_q, \quad \sum_{q=1}^Q a_{ijq} = 1, \quad a_{ijq} = 0 \text{ or } 1 \tag{12}$$

$$\sum_{i=1}^N \sum_{j=1}^{n_i} r_{ij} \leq \sum_{i=1}^N n_i u_i + u_s, \tag{13}$$

$$r_{ij} \geq e_i, \tag{14}$$

where  $a_{ijq}$  is a binary variable for the video version chosen by peer  $j$  in class  $i$ , (12) states that each peer only gets one video

version. (13) is the total video bandwidth constraint. Since peer entitled rate calculated from the taxation is continuous, we can find the corresponding *discrete entitled rate*  $e_i$  for class  $i$ , which is the highest discrete video rate  $v_{q(i)}$  such that  $v_{q(i)} \leq (1-t)u_i$  and  $v_{q(i)-1} > (1-t)u_i$ . Note that discrete entitled rate is always no greater than the corresponding continuous entitled rate. (14) guarantees that peers in class  $i$  at least get their discrete entitled rates.

Similar to the water-filling algorithms for the continuous case, we propose Algorithm 3 for the discrete case that invests upload bandwidth to enhance peers with lower video rates first. In Algorithm 3, we first assign peers with their discrete entitled rates. The residual upload bandwidth is put into a bandwidth pool  $B$ . Each time we select a peer with the smallest video rate to enhance its video rate to the adjacent higher one. If the bandwidth pool  $B$  is large enough for such an operation, we execute it, update bandwidth pool  $B$ , peer video rate, and move on to the further iteration; otherwise, the algorithm terminates, we get the lowest video rate  $v_{K^*}$ , and the second lowest video rate  $v_{K^*-1}$ .

---

#### Algorithm 3 Water-Filling-Discrete ( $\mathcal{U}, \mathcal{N}, V, u_s, N$ )

---

- 1: **for** each user  $i$  **do**
  - 2:    $r_{ij} = e_i$
  - 3: **end for**
  - 4:  $B = \sum_{i=1}^N (u_i - e_i) + u_s$
  - 5: All peers form a queue  $Queue_p$ . In the queue, peer  $i$  with smallest video rate  $r_{ij}$  appears first. If peers' video rates are the same, the peer contributing higher bandwidth  $u_i$  appears first.
  - 6: **while** 1 **do**
  - 7:   Select peer appearing in front of  $Queue_p$ , assume the video rate of that peer is  $v_j$
  - 8:   **if**  $v_{j-1} - v_j > B$  **then**
  - 9:      $v_{K^*} = v_j, v_{K^*-1} = v_{j-1}$
  - 10:    **break**
  - 11:   **end if**
  - 12:   Enhance video rate of peer from  $v_j$  to  $v_{j-1}$ ,  $B = B - (v_{j-1} - v_j)$ , insert that peer to the proper position of  $Queue_p$  according to its video rate
  - 13: **end while**
- 

*Theorem 6:* After Algorithm 3 completes, let  $\mathcal{X}$  be the obtained peer video rate allocation. If the bandwidth pool  $B$  is exhausted, then  $\mathcal{X}$  is the optimal for the discrete case. If  $B$  is not exhausted, the utility gap between  $\mathcal{X}$  and the optimal solution is upper-bounded by  $\frac{v_{K^*-1} - v_{K^*}}{v_{K^*}}$ .

*Proof:* After Algorithm 3 completes, some peers will get the lowest rate  $v_{K^*}$ , some peers will get the second lowest rate  $v_{K^*-1}$ , all other peers will get their discrete entitled rates. Let  $k_0$  be the class index such that peers from class  $k$ ,  $1 \leq k \leq k_0$  will watch their entitled rates. And  $B$  is the total upload bandwidth allocated. It is straightforward to see that  $B > \sum_{i=1}^N u_i n_i + u_s - (v_{K^*-1} - v_{K^*})$ , because otherwise the algorithm can always allocate the available bandwidth of

$v_{K^*-1} - v_{K^*}$  to bring one more peer from video rate level  $K^*$  to  $K^* - 1$ .

Now suppose  $\mathcal{Y}$  is another feasible discrete allocation with the total allocated bandwidth of  $B$ . Due to the entitled rate constraint, in  $\mathcal{Y}$ , all peers from class  $k \leq k_0$  will at least get the same video rates as in  $\mathcal{X}$ . Let  $\Delta$  be the total surplus rate obtained by peers from the first  $k_0$  class. Then their total utility satisfies:

$$\sum_{k=1}^{k_0} \sum_{j=1}^{n_k} \log r_{kj} \leq \sum_{k=1}^{k_0} n_k \log e_k + \frac{\Delta}{e_{k_0}}, \quad (15)$$

where the inequality is due to the concavity of  $\log(\cdot)$  function.

For peers from classes  $k > k_0$ , we sort their allocated rates in a non-decreasing order, the resulting vector is  $\{y_j, 1 \leq j \leq L \triangleq \sum_{k=k_0+1}^N n_k\}$ . Let  $\{x_j, 1 \leq j \leq L\}$  be the associated vector in  $\mathcal{X}$ . We have

$$\sum_{j=1}^L y_j = \sum_{j=1}^L x_j - \Delta. \quad (16)$$

Since  $x_j$  takes value of either  $v_{K^*}$  or  $v_{K^*-1}$ , and there is no other video rate in between, we must have  $y_1 \leq x_1 = v_{K^*}$ . Let  $j_0$  be the index such that  $y_{j_0} > v_{K^*}$  and  $y_{j_0-1} \leq v_{K^*}$ , then we have  $y_j \leq x_j, \forall j < j_0$ , and  $y_j \geq x_j, \forall j \geq j_0$ . Then the total utility difference is  $\sum_{j=1}^L (\log y_j - \log x_j) \leq \sum_{j=1}^L \frac{y_j - x_j}{x_j}$ . If  $x_{j_0} = v_{K^*}$ , then  $x_j = v_{K^*}, 1 \leq j \leq j_0$ , then

$$\sum_{j=1}^L (\log y_j - \log x_j) \leq \sum_{j=1}^{j_0-1} \frac{y_j - x_j}{v_{K^*}} + \sum_{j=j_0}^L \frac{y_j - x_j}{v_{K^*}} = -\frac{\Delta}{v_{K^*}} \quad (17)$$

If  $x_{j_0} = v_{K^*-1}$ , then  $x_j \leq v_{K^*-1}, 1 \leq j \leq j_0$ , then

$$\sum_{j=1}^L (\log y_j - \log x_j) \leq \sum_{j=1}^{j_0-1} \frac{y_j - x_j}{v_{K^*-1}} + \sum_{j=j_0}^L \frac{y_j - x_j}{v_{K^*-1}} = -\frac{\Delta}{v_{K^*-1}}. \quad (18)$$

Based on (15), (17), (18) and  $v_{K^*} \leq v_{K^*-1} \leq e_{k_0}$ , we have the total utility of  $\mathcal{Y}$  is no more than  $\mathcal{X}$ . In other words,  $\mathcal{X}$  is optimal among all allocations used up bandwidth of  $B$ . Since  $B > \sum_{i=1}^N u_i n_i + u_s - (v_{K^*-1} - v_{K^*})$ , the utility gap to the optimal allocation used up all bandwidth is at most  $\frac{v_{K^*-1} - v_{K^*}}{v_{K^*}}$ . ■

*Corollary 6.1:* If the gaps between two adjacent video rates are the same, Algorithm 3 always find the optimal discrete video rate allocation.

*Proof:* Given the video rate gap  $\delta$  among two adjacent video rates, for any two feasible discrete video rate allocations, the gap between their total allocated bandwidth must be a multiple of  $\delta$ . As proved in Theorem 6, after Algorithm 3 completes, the unused bandwidth is less than  $v_{K^*-1} - v_{K^*} = \delta$ . Therefore the total used bandwidth  $B$  of Algorithm 3 is no less than any other feasible allocation. Since we proved in Theorem 6 that  $\mathcal{X}$  is optimal among all allocations used bandwidth up to  $B$ . Then  $\mathcal{X}$  is optimal among all feasible discrete allocations. ■

## B. Discrete Layered Video Coding

When employing layered video coding, we assume that there are totally  $L$  video layers, and the rate for layer  $l$  is denoted by  $s_l$ . Similar to the analysis in the continuous case, without considering coding overhead, discrete layered video coding can be converted into an equivalent multiple video version case. Specifically, if we let  $Q = L$  and  $v_j = \sum_{i=1}^{Q+1-j} s_i, \forall 1 \leq j \leq Q$ , then this problem can be casted into **OPT II**. We can still use the similar approach like Algorithm 3 to solve this problem. Each time, we check whether we can add one more video layer to the peers with the smallest video rate.

## C. Helper-based P2P Distribution

**Algorithm 4** Video Version Allocation For Discrete Version under Helper-based condition

- 1: Run Algorithm Water-Filling-Discrete  $(\mathcal{U}, \mathcal{N}, V, u_s, N)$  to obtain  $v_{K^*}, v_{K^*-1}$ ;
- 2: Run Algorithm Water-Filling-Discrete  $(\mathcal{U}, \mathcal{N}, V, u_s - v_{K^*} - v_{K^*-1}, N)$  to obtain  $v_{K'}, v_{K'-1}, \mathcal{R}'$

To cope with helper-overhead, we apply the same two-round allocation trick in Section II-F to the discrete case. We can first use the discrete water-filling approach to find the lowest video rate  $v_{K^*}$  and the second lowest video rate  $v_{K^*-1}$  without considering the helper-overhead. Different from continuous case, peers in the two bottom groups might need help from the helpers. Thus, the server should reserve these two rates before executing the second round water-filling approach, as shown in Algorithm 4. Finally we get the new lowest video rate  $v_{K'}$  and second lowest video rate  $v_{K'-1}$ .

*Theorem 7:* The utility gap between the video rate vector  $\mathcal{R}'$  from Algorithm 4 and the optimal solution is upper-bounded by  $(\lfloor \frac{v_{K^*} + v_{K^*-1}}{v_{K'-1} - v_{K'}} \rfloor + 2) \frac{v_{K'-1} - v_{K'}}{v_{K'}}$ .

*Proof:* Compared with the result of OPT II, Algorithm 4 deducts bandwidth of  $v_{K^*} + v_{K^*-1}$  to feed the helpers, and might leave some unused bandwidth of  $B_{rest}$  ( $B_{rest} < (v_{K'-1} - v_{K'})$ ) in the second round water-filling. The maximal utility gain can be achieved if one can invest the total bandwidth of  $B_{rest} + v_{K^*} + v_{K^*-1}$  to increase the video rate of the poorest peers. Thus, the utility gap upper bound is

$$\begin{aligned} & \left( \left\lfloor \frac{B_{rest} + v_{K^*} + v_{K^*-1}}{v_{K'-1} - v_{K'}} \right\rfloor + 1 \right) [\log(v_{K'-1}) - \log(v_{K'})] \\ & < \left( \left\lfloor \frac{v_{K^*} + v_{K^*-1}}{v_{K'-1} - v_{K'}} \right\rfloor + 2 \right) \frac{v_{K'-1} - v_{K'}}{v_{K'}} \end{aligned} \quad \blacksquare$$

## IV. STOCHASTIC MODELS FOR BANDWIDTH VARIATIONS AND PEER CHURN

In the previous sections, we assume that there are multiple classes of peers with heterogeneous upload bandwidth, and study the optimal sharing between them under the assumption that peers are stable and the upload bandwidth of peers within

each class is time-invariant. In practice, peers join and leave the system dynamically, and their upload bandwidth varies over time. In this section, we develop stochastic models to study the impact of bandwidth variations and peer churn on P2P adaptive streaming.

### A. System without Peer Churn

Similar to the static case, we assume there are totally  $N$  possible upload bandwidth levels,  $\{u_i, 1 \leq i \leq N\}$ , and there are  $C$  classes of peers, with  $n^{(c)}$  peers in class  $c$ ,  $1 \leq c \leq C$ . Similar to [21], we model the upload bandwidth variation of peers in class  $c$  by a continuous time random process:

- 1) The peer upload bandwidth remains constant at a bandwidth level for a random amount of time before switching to another level. The holding time at level  $u_i$  has an arbitrary distribution with mean  $1/\mu_i^{(c)}$ .
- 2) The switch probability from level  $i$  to level  $j$  is  $p_{ij}^{(c)}$ . Let  $\mathbf{P}^{(c)}$  denote the  $N \times N$  switching matrix for class  $c$ .
- 3) Bandwidth varying processes between all peers in the same class are i.i.d. Bandwidth varying processes between peers in different classes are independent.

Let  $M_i^{(c)}$  be a random variable denoting the number of peers in class  $c$  at bandwidth level  $i$ . Because the total number of peers in class  $c$  is fixed at  $n^{(c)}$ , we have  $M_1^{(c)} + \dots + M_N^{(c)} = n^{(c)}$ . Thus, the random variables  $M_1^{(c)}, \dots, M_N^{(c)}$  are dependent. We can characterize the joint distribution of  $\{M_i^{(c)}, 1 \leq i \leq N\}$  by modeling the joint bandwidth switching process of peers in class  $c$  as an infinite-server Jackson queuing network [22], with each bandwidth level being a node in the network, and each peer as a customer which sojourns at node  $i$  for a random amount of time with mean  $1/\mu_i^{(c)}$ . Let  $\lambda^{(c)} = (\lambda_1^{(c)}, \dots, \lambda_N^{(c)})$  be the unique probability distribution that satisfies  $\lambda^{(c)} = \lambda^{(c)} \mathbf{P}^{(c)}$ . Let  $\rho_i^{(c)} = \lambda_i^{(c)} / \mu_i^{(c)}$ . We normalize the  $\rho_i^{(c)}$ 's so that  $\rho_1^{(c)} + \dots + \rho_N^{(c)} = 1$ . We immediately arrive at the following result:

*Theorem 8:* For any  $m_1^{(c)}, \dots, m_N^{(c)}$  with  $m_1^{(c)} + \dots + m_N^{(c)} = n^{(c)}$ , we have

$$P\left(M_1^{(c)} = m_1^{(c)}, \dots, M_N^{(c)} = m_N^{(c)}\right) = n^{(c)}! \frac{\rho_1^{(c)}}{m_1^{(c)}!} \dots \frac{\rho_N^{(c)}}{m_N^{(c)}!} \quad (19)$$

Essentially,  $(M_1^{(c)}, \dots, M_N^{(c)})$  has a multinomial distribution.

Let  $\mathcal{M} \triangleq \left\{ \{M_i^{(c)}\}_{i=1}^N \right\}_{c=1}^C$  be the collection of bandwidth levels of peers from all classes, and  $\omega \triangleq \left\{ \{m_i^{(c)}\}_{i=1}^N \right\}_{c=1}^C$  be a specific combination with  $\sum_{i=1}^N m_i^{(c)} = n^{(c)}$ . Due to the independence assumption on bandwidth variations between peers in different classes, we have

$$P(\mathcal{M} = \omega) = \prod_{c=1}^C P\left(\{M_i^{(c)} = m_i^{(c)}\}_{i=1}^N\right) \quad (20)$$

### B. Video Quality

For a specific peer upload bandwidth combination  $\omega$ , we can calculate the optimal video rate for all peers according to Theorem 1 by setting

$$n_i = \sum_{c=1}^C m_i^{(c)}, 1 \leq i \leq N.$$

Specifically, the base video rate is calculated as  $W^*(\omega)$ , and the bandwidth level threshold is  $K^*(\omega)$ , such that all peers at upload bandwidth level  $i < K^*(\omega)$  will get video rate  $(1-t)u_i$ , the rest of peers will get video rate  $W^*(\omega)$ .

To assess the video quality distribution of peers in class  $c$ , we define  $q^{(c)}(r)$  as the number of peers in class  $c$  with video rate no less than  $r$ . Then we have

$$q^{(c)}(r|\omega) = \begin{cases} n^{(c)}, & \text{if } W^*(\omega) \geq r \\ \sum_{i=1}^{k(r)} m_i^{(c)}, & \text{if } W^*(\omega) < r \end{cases} \quad (21)$$

where  $k(r)$  denotes the upload bandwidth level  $k$  such that  $(1-t)u_k \geq r$  and  $(1-t)u_{k+1} < r$ . Then the overall video rate distribution of class  $c$  can be calculated as

$$P(R^{(c)} \geq r) = P(\mathcal{M} \in \Omega_1(r)) + \sum_{\omega \in \Omega_2(r)} \frac{q^{(c)}(r|\omega)}{n^{(c)}} P(\mathcal{M} = \omega), \quad (22)$$

where  $\Omega_1(r) = \{\omega : W^*(\omega) \geq r\}$  and  $\Omega_2(r) = \{\omega : W^*(\omega) < r\}$ .  $P(R^{(c)} \geq r)$  can be easily calculated using the Monte Carlo method.

### C. Model Peer Churn

To include peer churn in the model, similar to [21], we use an open network of infinite-server queues. In such an open system, peers join and leave the system freely. Let  $\gamma_i^{(c)}$  be the exogenous arrival rate of peer class  $c$  at bandwidth level  $i$ . After staying at bandwidth level  $i$ , a peer leaves the system (peer churn) with probability  $p_{i0}^{(c)}$ , or switches to level  $j$  with probability  $p_{ij}^{(c)}$ . One can treat  $N$  bandwidth levels as an open Jackson network of  $N$  infinite-server queues (again with arbitrary sojourn time distributions). Let  $\lambda^{(c)} = (\lambda_1^{(c)}, \dots, \lambda_N^{(c)})$  be the effective arrival rates for all bandwidth levels, then

$$\lambda_i^{(c)} = \gamma_i^{(c)} + \sum_{j=1}^N \lambda_j^{(c)} p_{ji}^{(c)},$$

or in vector-matrix form  $\lambda^{(c)} = \gamma^{(c)} + \lambda \mathbf{P}^{(c)}$ , where  $\mathbf{P}^{(c)}$  is the  $N \times N$  level switching matrix for class  $c$ , and  $\sum_{j=1}^N p_{ij}^{(c)} = 1 - p_{i0}^{(c)}$ . Let  $\rho_i^{(c)} = \lambda_i^{(c)} / \mu_i^{(c)}$ , and  $\rho_i^{(c)}$  is the expected number of peers at level  $i$ . From the theory of Jackson networks [22], we can calculate bandwidth distribution for peers in class  $c$ :

*Theorem 9:* For the multi-class system with peer churn, the peer upload bandwidth distribution in class  $c$  is:

$$P(M_1^{(c)} = m_1^{(c)}, \dots, M_N^{(c)} = m_N^{(c)}) = \prod_{i=1}^N \frac{(\rho_i^{(c)})^{m_i^{(c)}} e^{-\rho_i^{(c)}}}{m_i^{(c)}!}$$

**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

Using (20), we can calculate the joint upload bandwidth distribution of peers from all classes  $P(\mathcal{M} = \omega)$ . Similar to the churnless case, we can further calculate video rate distribution based on (21) and (22).

## V. NUMERICAL STUDY

We conduct numerical case studies to verify our analysis and further illustrate the 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 [23] 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 [24].

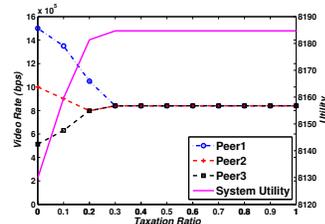
### A. Impact of Taxation Ratio

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

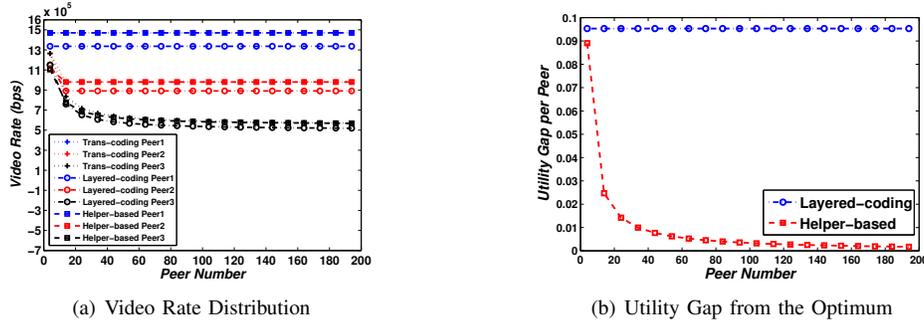
distribution achieves higher rate than layered-coding. When peer number is large, helper-overhead is almost negligible, and the achieved rates are almost the same as the transcoding case. Fig. 4(b) compares the per-peer utility gap from the optimal for helper-based distribution and layered coding. As 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.



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

### C. Impact of Discrete Video Rates

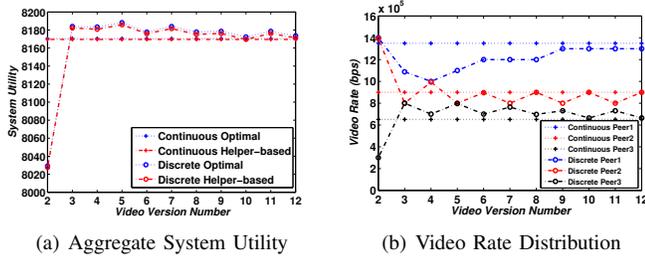
We investigate the impact of discrete video rates. We assume server's uplink capacity is 11,000 kbps, other peers' bandwidth setting still follows Table I and the taxation ratio is set to be 0.1. We set the discrete video versions according to Table II, where each column represents a configuration with a certain number of versions, the checked entries within each column are the offered video rates. Fig. 5(a) shows the utility of using helper-based approach and the corresponding optimal solution. The utilities for the continuous curves are invariant under version numbers. Conforming to the bound in Theorem 5, our helper-based distribution according to Algorithm 2 can achieve a close-to-optimal solution. For the discrete curves, conforming to the bound in Theorem 7, our discrete helper-based distribution according to Algorithm 4 can also achieve a close-to-optimal solution. It is interesting to notice that the optimal utility of the discrete case is initially lower than the continuous case, then quickly jumps up at three video versions, and converges back to the continuous optimum as the number of versions increases. This is because the discrete entitled rate of a peer is always lower than its continuous entitled rate. The gap can be large if the offered video versions are too coarse. As a result, the system can collect more taxes from strong peers in the discrete case than in the continuous case. The additional taxed bandwidth can be used to improve the system-wide utility, at the price of reduced rates on strong peers. This is illustrated in Fig. 5(b), when the video version numbers are three or four, the discrete entitled rate of peers in class 1 is much lower than their entitled rate in the continuous case. More of their upload bandwidth is used to increase the video rates of peers from class 2 and 3, leading to an improved system-wide utility. As the number of video version increases,


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

both the video rate distribution and the aggregate system utility converges to the continuous case. *This demonstrates that the number of provisioned video versions also plays an important role in the utility and incentive trade-off.*

**TABLE II:** Video Version Selection

Video Version	Video Version Number											
	2	3	4	5	6	7	8	9	10	11	12	
300 kbps	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
400 kbps									✓	✓	✓	
500 kbps					✓	✓	✓	✓	✓	✓	✓	
600 kbps			✓	✓	✓	✓	✓	✓	✓	✓	✓	
700 kbps				✓	✓	✓	✓	✓	✓	✓	✓	
800 kbps		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
900 kbps				✓	✓	✓	✓	✓	✓	✓	✓	
1000 kbps			✓	✓	✓	✓	✓	✓	✓	✓	✓	
1100 kbps				✓	✓	✓	✓	✓	✓	✓	✓	
1200 kbps					✓	✓	✓	✓	✓	✓	✓	
1300 kbps						✓	✓	✓	✓	✓	✓	
1400 kbps	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	


**Fig. 5:** Impact of Video Version Numbers in Discrete Case

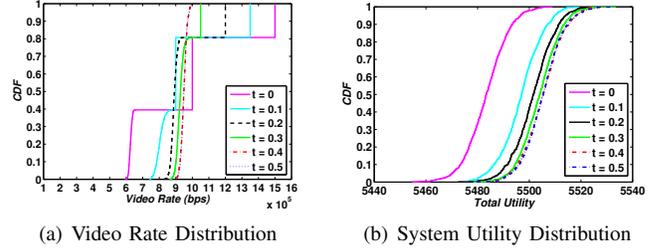
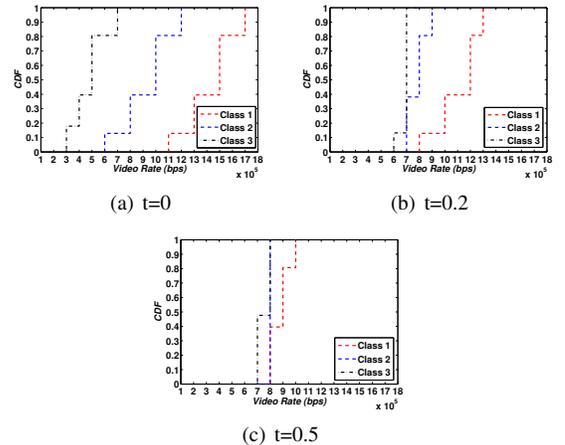
#### D. Results under Bandwidth Variations

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 four peer 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. 6(a) and Fig. 6(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 get closer 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, a high taxation ratio is justifiable. *Within the same class, taxation achieves the temporal multiplexing gain: it allows a peer with temporary bandwidth dips continue to get stable video download from other peers.*


**Fig. 6:** Single Class without Peer Churn

**Fig. 7:** Video Rates for Multiple Classes without Peer Churn

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 III, 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 two adjacent video versions is 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,<sup>2</sup> 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. 7(a), Fig. 7(b) and Fig. 7(c) plot the video rate distributions when the taxation ratio  $t = 0, 0.2, 0.5$  respectively. Table IV 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 weak 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.

**TABLE III: 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

**TABLE IV: 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

## VI. CONCLUSION

In this paper, we studied the capacity of P2P adaptive streaming by developing utility maximization models that take into account peer heterogeneity, taxation-based incentives, and multi-version videos at discrete rates. We demonstrated that incentive-compatible sharing between peers watching different video versions can be enabled through taxation. We characterized the capacity regions of P2P adaptive streaming with peer transcoding, layered video encoding, or helper-based distribution. Through analysis and simulations, we demonstrated that simple helper-based P2P distribution can achieve close-to-optimal efficiency. We further developed stochastic models to study the performance of P2P adaptive streaming in face of bandwidth variations and peer churn. We showed that P2P adaptive streaming not only significantly reduces the load on

the servers, but also improves the stability of user-perceived video quality in face of dynamic bandwidth changes. As future work, we plan to develop a helper-based P2P adaptive streaming system and test its adaptiveness and robustness through experiments on the Internet.

## 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] J. Zhao, F. Yang, Q. Zhang, Z. Zhang, and F. Zhang, "LION: Layered Overlay Multicast with Network Coding," *IEEE Trans. Multimedia*, vol. 8, pp. 1021–1032, Oct. 2006.
- [10] X. Xiao, Y. Shi, Y. Gao, and Q. Zhang, "Layerp2p: A new data scheduling approach for layered streaming in heterogeneous networks," in *IEEE INFOCOM*, 2009.
- [11] Z. Liu, Y. Shen, S. S. Panwar, K. W. Ross, and Y. Wang, "Substream Trading: Towards an Open P2P Live Streaming System," in *ICNP*, 2008.
- [12] 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.
- [13] Y.-W. Sung, M. Bishop, and S. Rao, "Enabling Contribution Awareness in an Overlay Broadcasting System," in *SIGCOMM*, 2006.
- [14] 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.
- [15] J. Li, P. A. Chou, and C. Zhang, "Mutualcast: An Efficient Mechanism for Content Distribution in a P2P Network," in *Sigcomm Asia Workshop*, 2005.
- [16] R. Kumar, Y. Liu, and K. Ross, "Stochastic Fluid Theory for P2P Streaming Systems," in *Proc. IEEE INFOCOM*, May 2007.
- [17] L. Massoulié, A. Twigg, C. Gkantsidis, and P. Rodriguez, "Randomized decentralized broadcasting algorithms," in *Proceedings of IEEE INFOCOM*, 2007.
- [18] Y. Xu, Y. Liu, and K. Ross, "Capacity Analysis of Peer-to-Peer Adaptive Streaming," *Technical Report, Polytechnic Institute of NYU*, July 2013, [http://eeweb.poly.edu/faculty/yongliu/docs/lyang\\_TR1.pdf](http://eeweb.poly.edu/faculty/yongliu/docs/lyang_TR1.pdf).
- [19] D. Wu, C. Liang, Y. Liu, and K. W. Ross, "Redesigning Multi-Channel P2P Video Systems with View-Upload Decoupling," in *Computer Networks Journal, Elsevier*, 2010, accepted for publication.
- [20] M. Chen, M. Ponc, S. Sengupta, J. Li, and P. A. Chou, "Utility maximization in peer-to-peer systems," in *Proceedings of ACM SIGMETRICS*, vol. 36, Jun. 2008.
- [21] D. Wu, C. Liang, Y. Liu, and K. W. Ross, "Modeling and analysis of multi-channel p2p live video systems," *IEEE/ACM Transactions on Networking*, August 2010.
- [22] F. P. Kelly, *Reversibility and Stochastic Networks*. Wiley, 1979.
- [23] H. Schwarz and T. Wiegand, "R-D Optimized Multi-Layer Encoder Control for SVC," in *Proceedings of the International Conference on Image Processing*, 2007.
- [24] 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.

<sup>2</sup>Since different classes have different four bandwidth levels, the actual bandwidth variation processes between different classes are different.