Distributed Caching via Rewarding: An Incentive Scheme Design in P2P-VoD Systems

Weijie Wu, Richard T.B. Ma, and John C.S. Lui, *Fellow*, *IEEE*

Abstract—Peer-to-peer (P2P) systems rely on peers' cooperation to provide a more robust and scalable service as compared to the traditional client-server architecture. However, the peers might be selfish in nature—they would like to receive services from others, but would not like to contribute their own resources by default. To conquer this problem, proper incentive schemes are needed so as to stimulate the peers' contributions. In particular, in P2P video-on-demand (VoD) systems, peers need to distributively cache the proper videos so as to mutually upload and help each other to acquire the required data. Content providers of P2P-VoD services want to incentivize peers to do so and alleviate the workload of the content server. In this paper, we design a practical mechanism to incentivize distributed caching in such systems, under which the peers are rewarded based on the popularity of the video they cache. We characterize the impact of this incentive scheme on peers' caching behaviors. In particular, we formulate an optimization framework to decide the optimal reward price for each video so as to keep enough replicas and minimize the content provider's operational cost. We first derive close form solutions in an asymptotic system, and then extend our results to be adaptive to various practical issues. Via extensive simulations, we validate the effectiveness and efficiency of our incentive scheme.

Index Terms—P2P-VoD, incentive, pricing, mean field, optimization

1 Introduction

In the recent years, we have witnessed the success of adopting peer-to-peer (P2P) technologies into video-on-demand (VoD) systems. There are number of large-scale and commercial P2P-VoD systems like PPLive and PPStream. In such systems, peers cache video data in their local storage and deliver them to one another when they have available upload bandwidth. Compared with traditional VoD deployments, P2P-VoD systems can be more scalable and fault tolerant. In the meanwhile, the Internet service providers (ISPs) usually use the *volume-based* charging scheme. Therefore, by utilizing the distributed resources of the peers, the content provider can greatly reduce its operational cost due to the reduction of upload requirement at the content servers.

However, peers in a P2P-VoD system might be selfish in nature and would not be willing to contribute their resources (e.g., upload bandwidth, local storage space) by default. Hence, designing an effective incentive scheme is critical. Unlike traditional P2P file sharing applications in which many incentive schemes have been proposed, very limited work has been focusing on the P2P-VoD applications. Designing a proper incentive scheme is challenging due to the following reasons. First, in P2P-VoDs, sequential fetching or downloading is needed, in particular, when a peer starts watching a new video, or when the download rate is not sufficiently enough. This differs from the rarest-first strategy so that traditional BitTorrent protocol cannot directly apply to VoD. Second, the P2P-VoD system is large scale and stochastic in nature. It is difficult to predict, or even describe the peers' requests and resources at a particular time. Third, unlike the P2P file sharing system where *availability* of replica is enough (provided that peers are patient enough to wait a long time for downloading), P2P-VoDs need to keep *enough* replicas so as to guarantee the download rate to satisfy peers' viewing requirements. Last but not the least, peers in P2P-VoDs are heterogeneous, for example, they may have very different available bandwidth or storage to contribute, so they have various responses to a particular incentive scheme. Partially due to these difficulties, current commercial systems do not involve any incentives, but rather, they simply force the peers to contribute, for example, in PPStream, peers need to reserve a 2-GB storage space when installing the software. In this paper, we explore the design of a "*reward-based incentive scheme*". We believe such mechanism can greatly improve the fairness, robustness, and user friendliness.

Each peer in a P2P-VoD system needs contribute their 1) local storage space to distributively cache video data, and 2) upload bandwidth for uploading data to other peers. Both aspects are equally important because a peer cannot contribute if it fails to do either of them. Our previous work [22] focused on incentivizing the upload bandwidth. In this paper, we focus on stimulating peers to cache the needed video data. We propose a reward-based incentive scheme for distributed caching, where the content provider decides the *reward price*[1] of each video, and peers decide what

1. In the remainder of this paper, we use the term "price" for short.

- W. Wu is with the School of Information Security Engineering, Shanghai Jiao Tong University, 800 Dongchuan Road, Minhang District, Shanghai 200240, P.R. China. E-mail: wuwjpku@gmail.com.
- R.T.B. Ma is with the School of Computing, National University of Singapore, COM2-03-38, 15 Computing Drive, Singapore 117418. E-mail: tbma@comp.nus.edu.sg.
- J.C.S. Lui is with the Computer Science and Engineering Department, The Chinese University of Hong Kong, Shatin, New Territories, Hong Kong. E-mail: cslui@cse.cuhk.edu.hk.

Manuscript received 26 Apr. 2012; revised 12 Mar. 2013; accepted 20 Mar. 2013; published online 28 Mar. 2013.
Recommended for acceptance by D. Epema.
For information on obtaining reprints of this article, please send e-mail to: tpds@computer.org, and reference IEEECS Log Number TPDS-2012-04-0419.
Digital Object Identifier no. 10.1109/TPDS.2013.94.

videos to cache in a distributed manner. Peers are rewarded according to the preset price if they cache a particular video. Although most systems are free for ordinary service, our reward scheme is practical as the reward can be in credits or service fee rebate for premium services (e.g., HD channels) for which commercial systems do charge users.

We apply a *mean field model* [2], [3] to characterize the steady state of the caches in a large-scale P2P-VoD system. By transforming a large-scale stochastic system into a limiting deterministic one, the mean field technique enables us to represent the system state by the fraction of peers in each cache state. Based on this, we formulate a reward pricing problem using an optimization framework and solve the optimal reward prices that minimize the content provider's operational cost. Extended from our previous work [23], we consider more general system settings and practical issues. Our contributions are:

- We develop a stochastic model to capture the peers' caching behaviors, and use the mean field technique to characterize the system in a limited steady state.
- We formulate an optimal pricing problem of the content provider and derive the optimal prices. Our scheme keeps enough replicas for each video and minimizes the content provider's operational cost.
- We extend our protocol design to be adaptive to general and practical system environments.
- We validate the effectiveness of our incentive scheme by extensive simulations.

We organized this paper as follows: In Section 2, we develop a mathematical model to characterize the system's cache state, and present an optimization framework for the pricing scheme. In Section 3, we analyze the pricing schemes in a practical asymptotic case, where we give closed-form solutions for the optimal pricing schemes. We also analyze the nonasymptotic case in Section 4. We further generalize our model and the corresponding incentive schemes to adapt various practical issues in Section 5, and perform extensive simulations to evaluate our pricing schemes in Section 6. Section 7 and 8 state related work and conclusions.

2 Model

2.1 Preliminaries

We consider fixed number of peers and videos in the system. In a realistic system, the number of videos can be large; however, the incentive decision is often made only upon the popularity of the videos. We categorize the videos into M classes, each containing videos of similar popularity. We denote V_i as a typical video of class i.

We first characterize the system's state in caching each video. A straightforward way is using a stochastic model where the system's state is determined by *all* peers' cache states. However, this can be computationally expensive, in particular, when the number of peers is large, the system's state space becomes intractable. To overcome this difficulty, we model the peers using a *mean field* approach, where we are more interested in the *fractions* of peers in certain states in steady state, rather than the exact number of peers in the system. We state the rationale of using the mean field limit in the supplementary file, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TPDS.2013.94.

Users do not always stay in the system, and our model only focuses on those currently online peers. Given any time, some peers have just rejoined the system after disconnecting for some period. We argue these peers need to refresh their local storage with a certain probability because the contents they have cached are stale. To validate this argument, we note that in real systems, some peers mainly watch TV series and they frequently access the VoD service (e.g., on per day basis). A new episode is the most popular on its publishing date, and the popularity dramatically decreases thereafter. Hence, when they rejoin the system with the episode they cached yesterday, they have a lower chance to contribute. Some other peers mainly watch movies and on average, they have a long period being disconnected before rejoining the system (e.g., one or several weeks). Movie popularity changes smoothly compared to TV series, but it often becomes unpopular one week after publishing, thereby decreasing the chance for the rejoining peers to contribute. To summarize, a rejoining peer has a high probability to refresh its storage. We have not found papers with relevant measurement results, but we consulted engineers from a VoD company and they validated this fact.

Denote p_0 as the probability that a peer refreshes its local storage with an empty cache space after finishing watching a video. This happens when this peer rejoins the system with previously stale contents. This probability is not small because 1) the rejoining probability is not negligible, and 2) the contents in the rejoining peers' cache are quite likely to be stale. Otherwise (i.e., the peer has been in the system, or it rejoins the system with useful cache content), we denote p_j as the probability that the chosen video is of class j (or V_j), which can be considered as the *popularity* of V_j. All these transiting probabilities satisfy $\sum_{j=0}^{M} p_j = 1$.

In our scheme, the content provider provides an external reward of v_j for each video of V_j cached. We call v_j the *price* of V_j and define the price vector as $\boldsymbol{v} = (v_1, v_2, \dots, v_M)$. Our core design problem is to decide the optimal prices for various videos in the view of content provider. For any fixed price vector, without loss of generality, we index the videos in a nonincreasing order of prices, i.e., $v_1 \geq \cdots \geq v_M$. Later, we will explore its relationship with the order of video popularity.

Each peer decides whether to cache the video according to the price of the video. To characterize the peers' sensitivity on prices, we classify all peers into $M+1$ categories. Peers of type 0 are not willing to cache any video, whereas peers of type m are only willing to cache the m highest-priced videos. This implies that peers of type 0 care much on their storage costs and they do not want to cache any video, while peers of type M are the most insensitive ones and are willing to cache every video. In what follows, we characterize the caching behaviors of the peers of various types, and then explore the impact of the pricing scheme on the distribution of video replicas in the whole system.

2.2 Peers' Caching Behaviors

We consider a typical peer of type m that has C units of storage, i.e., it can cache up to C videos in its local storage.

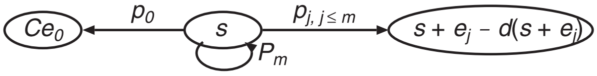


Fig. 1. Transition diagram.

We define the peer's state at any time as the videos that it has cached in its local storage. Each storage unit can be empty or holding any video of any class. Notice that a peer might cache multiple videos from the same class that have similar popularity. In real systems, we are only interested in the content that a peer caches, but not the physical caching sequences or the specific storage units. For example, if a peer caches two videos, one of V_1 and another of V_2, its state should be independent of which video is cached first and which storage units it uses to cache both videos. Formerly, we use an $m+1$ dimensional vector $\boldsymbol{s} = (s_0, s_1, \dots, s_m)$ to represent the cache state of a peer of type m. The state $\boldsymbol{s}$ is defined upon type m, but we omit the superscript m to make the notation neat. The first element s_0 denotes the number of empty units. For any $j > 0$, s_j denotes the number of videos of V_j that this peer caches. For example, if a type 3 peer has six storage units and caches one video of V_1 and two videos of V_3, then this peer still has three available caching units and its cache state is $(3, 1, 0, 2)$. The state space of a type m peer is $S_m = \{\boldsymbol{s} : \sum_{j=0}^{m} s_j = C, s_j \geq 0\}$. We let $\boldsymbol{s} = (s_0, \dots, s_m) = \sum_{j=0}^{m} s_j \boldsymbol{e}_j$, where $\boldsymbol{e}_j$ is a vector with the jth element being 1 and all other elements being 0. We define $t(\boldsymbol{s}) = \arg\max_i \{s_i : s_i > 0\}$ as the largest class index of the videos cached in state s. We define a *deletion* operation $\boldsymbol{d}(\boldsymbol{s})$ on state $\boldsymbol{s}$ as

$$\boldsymbol{d}(\boldsymbol{s}) = \begin{cases} \boldsymbol{e}_0 & \text{if } s_0 > 0, \\ \boldsymbol{e}_{t(\mathbf{s})} & \text{otherwise.} \end{cases} \tag{1}$$

The deletion operation maps a cache state to the video that will be replaced, if necessary. If the peer has available storage, then the operation maps to the empty slots $\mathbf{e}_0$; otherwise, it maps to the largest class index in s.

Based on the above notation, we illustrate the cache state transition diagram in Fig. 1. Given a current state $\boldsymbol{s}$, three types of transition can happen to a peer. First, with probability p_0, the peer refreshes its local storage and transits to state $C\boldsymbol{e}_0$, where all its C units of storage become empty. Second, with probability $P_m = \sum_{s=m+1}^{M} p_s$, which is the aggregate probability that the peer watches any video from the set $\{V_s : s > m\}$, the cache state remains the same, since the peer watches some video that it is not going to cache. Third, with the probability p_j for $j \leq m$, the peer watches some video of V_j and wants to cache it. Depending on whether the current cache space is full, the deletion operation might be needed to replace the least-priced video in the cache.

Fig. 2 illustrates the complete cache state transition diagram for a peer of type 2 with capacity $C = 2$.

In closing, we mention that this basic model can be extended to more general cases (e.g., intelligent peers) which we will show in later sections.

2.3 Cache State Distribution of Peers of Type m

In this section, we use the mean field model to approximate the fraction of peers in each state. We denote $q_m(\boldsymbol{s})$ as the fraction of type m peers that are in state $\boldsymbol{s}$ in a steady state. The existence and uniqueness of such a stationary distribution is guaranteed since the Markov chain is time homogeneous, irreducible, and that all states are positive recurrent [12].

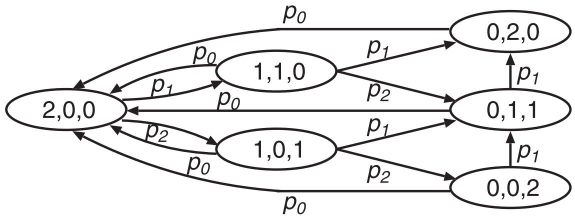


Fig. 2. An example of cache state transitions.

We use the global balance equations [12] to derive the steady state distribution of peers in each cache state. The global balance equation requires that rate at which peers arrive at a state equals the rate the peers depart from this state. We distinguish three scenarios where a peer's local storage is empty, partially occupied and full, and derive $q_m(\boldsymbol{s})$ separately for the three cases. For the state $\boldsymbol{s} = C\boldsymbol{e}_0$ that represents the empty storage, we have

$$q_m(C\boldsymbol{e}_0) \sum_{j=1}^{m} p_j = (1 - q_m(C\boldsymbol{e}_0)) p_0. \tag{2}$$

The left side describes the rate at which the peers depart from state $C\boldsymbol{e}_0$, which equals the fraction of peers in state $q_m(C\boldsymbol{e}_0)$, multiplied by the probability that they watch and cache some video, i.e., $\sum_{j=1}^{m} p_j$. The right side describes the rate at which peers arrive into state $C\boldsymbol{e}_0$, which equals the fraction of peers not in state $C\boldsymbol{e}_0$, multiplied by the refreshing probability p_0. From (2), we can solve $q_m(C\boldsymbol{e}_0)$ as

$$q_m(C\boldsymbol{e}_0) = \frac{p_0}{\sum_{j=0}^{m} p_j} = \frac{p_0}{1 - P_m}. \tag{3}$$

Similarly, for any state $\boldsymbol{s}$ with $s_0 > 0$, i.e., a partially occupied cache, we have the following equation:

$$q_m(\boldsymbol{s}) \sum_{j=0}^{m} p_j = \sum_{j \in J(\boldsymbol{s})} q_m(\boldsymbol{s} + \boldsymbol{e}_0 - \boldsymbol{e}_j) p_j, \tag{4}$$

where $J(\boldsymbol{s}) = \{j : j > 0, s_j > 0\}$, i.e., the set of class indices for which at least one video is cached at state $\boldsymbol{s}$. The right side describes possible transitions to $\boldsymbol{s}$ happen from any cache state $\boldsymbol{s} + \boldsymbol{e}_0 - \boldsymbol{e}_j$ that has one less video of V_j than $\boldsymbol{s}$, with the probability p_j that the peer starts to cache a video of V_j. Starting from the result of (3) serving as the right side of (4), we can progressively and recursively solve the above equation and obtain

$$q_m(\boldsymbol{s}) = \frac{|\boldsymbol{s}|! p_0 \prod_{j=1}^{m} p_j^{s_j}}{(1 - P_m)^{|\boldsymbol{s}|+1} \prod_{j=1}^{m} s_j!}, \qquad \forall \boldsymbol{s}, s_0 > 0, \tag{5}$$

where $|\boldsymbol{s}| = \sum_{j=1}^{m} s_j$ denotes the total number of replicas cached by this peer.[2]

2. The definition of $|\boldsymbol{s}|$ excludes s_0 from the summation since s_0 denotes the number of empty units.

Lastly, for the state $\boldsymbol{s}$ with $s_0 = 0$, i.e., a full storage, the corresponding balancing equation is

$$q_m(\boldsymbol{s}) \sum_{j=0}^{t(\boldsymbol{s})-1} p_j = \sum_{k \in K(\boldsymbol{s})} \sum_{j \in J(\boldsymbol{s})} q_m(\boldsymbol{s} + \boldsymbol{e}_k - \boldsymbol{e}_j) p_j, \tag{6}$$

where $K(\boldsymbol{s}) = \{k : k = 0 \text{ or } t(\boldsymbol{s}) < k \leq m\}$. To calculate the above $q_m(\boldsymbol{s})$, we sort the states $\{\boldsymbol{s} : s_0 = 0\}$ in an increasing order of the sequence $(s_1 s_2 \ldots s_m)$. For the example in Fig. 2, the states are sorted as $(0,0,2), (0,1,1), (0,2,0)$. Then we can solve (6) for the states according to the sorted order so that the right-hand side quantities will already be available.

Let N be the total number of peers and N_m be the number of peers of type m. Define $r_m(j)$ as the per peer average number of videos of V_j cached by type m peers:

$$r_m(j) = \sum_{\forall \boldsymbol{s} \in S_m} s_j q_m(\boldsymbol{s}). \tag{7}$$

2.4 Cache State of the System

Based on the cache state distribution $\{q_m(\boldsymbol{s}) : \boldsymbol{s} \in S_m\}$, we now derive the cache state of the entire system. This depends on the number of peers of each type, which is further determined by the prices of the videos. For example, if v_M is large, then more peers will be of type M; if v_1 is small, then more peers will be of type 0. The distribution of peer types also depends on how sensitive the peers are toward prices. We start with a simplified *linear sensitivity model*, which is generalized in later sections, to characterize the impact of prices on the distribution of peer types. This linear model assumes that the fraction of peers willing to cache any video is *proportional* to the price of that video. In particular, define V as the lowest price for which all peers are willing to cache the video. By proposing price v_j, a fraction $\min\{v_j/V, 1\}$ of the peers are willing to cache V_j.

Naturally, the content provider can set up any nonnegative price for videos; however, setting the price higher than V cannot be more beneficial than setting at V. Hence, we focus on the design space $v_j \in [0, V], \forall j$. Under our linear model, by defining $v_0 = V$ and $v_{M+1} = 0$, we can express the number of type m peers in the system as

$$N_m = \frac{v_m - v_{m+1}}{V} N. \tag{8}$$

In particular, $N_0 = (1 - v_1/V)N$ denotes the number of peers unwilling to cache any video, and $N_M = v_M N/V$ denotes those willing to cache all videos. Hence, the number of videos of V_j in the system is

$$\begin{aligned} R_j(\boldsymbol{v}) &= \sum_{m=0}^{M} N_m r_m(j) = \frac{N}{V} \sum_{m=0}^{M} (v_m - v_{m+1}) r_m(j) \\ &= \frac{N}{V} \sum_{m=1}^{M} [r_m(j) - r_{m-1}(j)] v_m. \end{aligned} \tag{9}$$

For the ease of notation, we express $R_j(\boldsymbol{v})$ as

$$R_j(\boldsymbol{v}) = \sum_{m=1}^{M} l_{mj} v_m, \tag{10}$$

where $l_{mj} = \frac{N}{V}(r_m(j) - r_{m-1}(j))$.

The above equation shows that, under the linear model, the number of videos of any class V_j is a linear combination of all the prices. Hence, given a set of prices, we can characterize the number of video replicas of each class V_j in the system. Our design space is to strategically set the prices to achieve certain objectives for the system.

2.5 Design Objectives of the Pricing Scheme

The content provider proposes the incentive scheme to reduce its operational cost. We define two kinds of pricing strategies that aim at different objectives.

Conservative pricing problem (CPP): A major part of the content provider's operational cost is the upload cost for delivering data to the peers that cannot be satisfied by other peers' contribution, due to the lack of video replicas in the system. Therefore, the content provider would like to set the prices such that the number of cached replicas can satisfy all peers' demand. In this paper, we assume that we know the desired number of replicas in this system, which has been addressed by the previous work [21]. We denote $\hat{R}_j$ as the desired number of replicas for V_j. The conservative pricing problem tries to find a price vector $\boldsymbol{v} = (v_1, \ldots, v_M)$ that satisfies the following constraints:

$$\hat{R}_j = R_j(\boldsymbol{v}) \quad \text{and} \quad 0 \leq v_j \leq V, \quad \forall j = 1, \ldots, M. \tag{11}$$

In other words, the content provider wants to find the prices for the videos such that the supplied number of video replicas would be the exact desired amount. The content provider is *conservative* since it ensures no upload consumption at the content server (provided that the peers' uplink bandwidth is enough), despite that the prices for some videos might be high.

Strategic pricing problem (SPP): The operational cost comes not only from the upload cost of the servers, but also from the reward payable to all the peers that cache videos. Therefore, the content provider might not want to guarantee the desired amount of cached videos in peers' storage. In reality, it is sometimes the best interest for the content provider to set lower prices so as to reduce the reward cost and balance the overall utility. Formally, we denote $C(\boldsymbol{v})$ as the content provider's operational cost, which consists of an upload cost $C_u(\boldsymbol{v})$ as well as the reward cost $C_p(\boldsymbol{v})$ it pays to all peers.

If $R_j < \hat{R}_j$, the replicas of V_j are not enough in the system, then the server incurs an upload cost of C_u that is proportional to the deficit number of replicas, defined by $C_u(\boldsymbol{v}) = c_u \sum_{j=1}^{M} (\hat{R}_j - R_j(\boldsymbol{v}))^+$, where c_u is the unit cost, and $(x)^+ = \max(0, x)$. The cost of reward C_p is the total rewards that the content provider pays to all peers, defined by $C_p(\boldsymbol{v}) = \sum_{j=1}^{M} v_j R_j(\boldsymbol{v})$. Thus, the operational cost of the content provider $C(\mathbf{v})$ is

$$\begin{aligned} C(\boldsymbol{v}) &= C_u(\boldsymbol{v}) + C_p(\boldsymbol{v}) \\ &= c_u \sum_{j=1}^{M} \left(\hat{R}_j - \sum_{m=1}^{M} l_{mj} v_m \right)^+ + \sum_{j=1}^{M} \sum_{m=1}^{M} v_j l_{mj} v_m. \end{aligned} \tag{12}$$

The strategic pricing strategy requires the content provider to find $\boldsymbol{v}$ to minimize its operational cost, i.e.,

$$\begin{aligned} \min_{\boldsymbol{v}} \quad & C(\boldsymbol{v}) \\ \text{subject to} \quad & 0 \leq v_j \leq V, \quad \forall j = 1, \ldots, M. \end{aligned} \tag{13}$$

To close this section, we relate the two pricing problems as follows: When c_u is very large and there is a deficit of replicas, then the upload cost is significantly larger than the reward cost, i.e., $C_u(\boldsymbol{v}) \gg C_p(\boldsymbol{v}), \forall \boldsymbol{v}$. In this case, the content provider would try to keep the replicas enough for each video. If CPP has a solution, then the solution to SPP converges to the solution to CPP when $c_u \to \infty$.

3 Asymptotic Analysis

In this section, we analyze a practical asymptotic case of the P2P-VoD systems where either the local storages are refreshed quite frequently, i.e., a large value for p_0, or they have large capacities, i.e., a large value for C. Most of the asymptotic results are derived under the limiting condition: $(1-p_0)^C \to 0$. Physically, the above condition means that the probability a peer keeps watching C videos without refreshing its local storage approaches zero. We emphasize that this limiting condition is practical for real systems where peers do not often watch many videos continuously. As a result, when a peer stops and rejoins the system, most likely the data in its local storage become stale and not so useful for other peers. This reflects the evidence of a high refreshing probability p_0 being reasonable. Meanwhile, the storage capacity of a peer (i.e., its local hard disk resource) is typically large, although a typical peer might want to cache only a limited number of videos. As an example, if a peer has a probability $p_0 = 0.3$ for refreshment, and caches at most eight videos, then the quantity $(1-p_0)^C = 0.7^8$ is quite close to zero.

3.1 Cache State of Peers

We characterize the asymptotic number of V_j cached by a peer of type m, i.e., $r_m(j)$, by the following theorem.

Theorem 1. *The average number of V_j cached by a type m peer approaches p_j/p_0 when $(1-p_0)^C \to 0$, i.e.,*

$$r_m(j) \to p_j/p_0 \quad \text{when} \quad (1-p_0)^C \to 0, \quad \forall j \leq m.$$

Due to page limit, proofs are in the online supplemental material.

In the following, we call a system as *an asymptotic system* when $r_m(j) = p_j/p_0, \forall 1 \leq j \leq m \leq M$. We are interested in solving both the conservative and strategic pricing problems in an asymptotic system.

3.2 Conservative Pricing Problem

We first discuss the conservative pricing problem (CPP) which ensures enough replicas of all videos in the system. In the following, we will derive the *order* and *value* of prices for a given set of video popularity. We start from the following lemma, which establishes the relationship between the cache state of a video and its popularity:

Lemma 1. *If two classes of videos are priced the same, then the ratio of the numbers of replicas in an asymptotic system equals the ratio of their popularity. Formally, labeling them as V_i and V_{i+1} with $v_i = v_{i+1}$, then $R_i : R_{i+1} = p_i : p_{i+1}$.*

The above lemma implies that, when setting all the prices equal, the number of cached copies for each video would be *proportional* to its popularity, resulting a suboptimal solution for the system. It was pointed out in [21] that one needs to be "greedy" in replicating unpopular videos in a P2P-VoD system. Formally, we have the following assumption:

Assumption 1. *For any two videos V_i and V_j with $p_i < p_j$, the desired number of replicas, $\hat{R}_i$ and $\hat{R}_j$, satisfy the following condition: $p_i/p_j < \hat{R}_i/\hat{R}_j < 1$.*

An underlined physical reasoning of this assumption is that, a larger group of peers watching one particular video can cooperate more effectively than a smaller group. Hence, the desired number of replicas increases sublinearly with respect to the video popularity. We have the following lemma:

Lemma 2. *Under Assumption 1, if CPP in an asymptotic system has a solution $\boldsymbol{v}$, then for any two videos V_i and V_j with popularity $p_i < p_j$, we have $v_i > v_j$.*

The above lemma indicates the important fact that in CPP, the *order* of prices is the *reverse order* of video popularity. This implies that we need to set higher prices for less popular videos so that more peers would like to cache them so as to meet the *greedy* cache requirement [21]. We assumed previously that $V_1, \ldots, V_M$ are arranged in a *nonincreasing* order of prices, and based on the above result, the popularity of the videos would be in a *nondecreasing* order, i.e., V_1 is the most unpopular video and is priced the highest, whereas V_M is the most popular one and is priced the lowest.

Given the video popularity, we can now determine the *order* of prices. Next, we need to decide the *value* of each video, which is shown in the following theorem.

Theorem 2. *Under Assumption 1, a necessary and sufficient condition that CPP in an asymptotic system has a solution is $p_0\hat{R}_1 \leq p_1 N$. If this condition is satisfied, then the solution is $v_j = \frac{p_0 \hat{R}_j}{p_j N} V, \forall j$.*

3.3 Strategic Pricing Problem

In this section, we solve the strategic pricing problem (SPP) where the content provider sets prices to minimize its operational cost. The following theorem shows the existence and the closed form of the solution.

Theorem 3. *There always exists a solution to SPP. The solution to SPP for an asymptotic system is $v_j = \min\{\frac{1}{2}c_u, \frac{p_0\hat{R}_j}{p_j N} V, V\}$, $\forall 1 \leq j \leq M$.*

Theorems 2 and 3 point out the condition under which CPP and SPP have solutions. Furthermore, we can get closed-form solutions for these pricing problems in an asymptotic system. In the real systems which satisfy $(1-p_0)^C \to 0$, we can apply Theorems 2 and 3 in the incentive mechanism design, by setting the video prices using the above asymptotic solutions. By doing this, we can approach to the system design objectives, i.e., keeping enough replicas and minimizing the operational cost. It is also worth noting that the solutions to both pricing problems indicate the "reverse order" phenomenon, i.e., for any $p_i \leq p_j$, we have $v_i \geq v_j$, which is an important guideline for designing the pricing schemes in practice.

4 Nonasymptotic Analysis

In this section, we show that our pricing scheme can be extended to a general case where the condition $(1-p_0)^C \to 0$ does not necessarily hold. This can apply to scenarios where C is small, for example, VoD on the mobile devices. One can hardly obtain closed-form solutions like in Section 3. We start from the following important theorem, which simplifies our analysis for a given order of video prices.

Theorem 4. *Given a fixed order of prices for the videos, the per peer average number of replicas for V_j by type m peers equals to the number by type $m-1$ peers ($\forall j \leq m-1$), i.e., $r_m(j) = r_{m-1}(j)$, $\forall j \leq m-1$.*

In short, this theorem implies that the value of $r_m(j)$ is independent of m as long as $m \geq j$. This can greatly simplify the problem for a given order of prices. In particular, $l_{mj} = \frac{N}{V}(r_m(j) - r_{m-1}(j)) = \mathbf{1}_{\{j=m\}} \frac{N}{V} r_j(j)$. Hence, CPP becomes

$$\hat{R}_j = \frac{N}{V} r_j(j) v_j, \quad 0 \leq v_j \leq V, \quad \forall j = 1, \ldots, M, \tag{14}$$

and SPP becomes

$$\begin{aligned} \min_{\mathbf{v}} \quad & c_u \sum_{j=1}^{M} \left(\hat{R}_j - \frac{N}{V} r_j(j) v_j \right)^+ + \sum_{j=1}^{M} \frac{N}{V} r_j(j) v_j^2 \\ \text{s.t.} \quad & 0 \leq v_j \leq V, \quad \forall 1 \leq j \leq M, \\ & v_i - v_j < 0, \quad \forall i < j. \end{aligned} \tag{15}$$

Note that for any fixed order of prices, we can calculate $r_i(j)$ from the close form or recursive algorithm stated in Section 2. Therefore, we can solve our pricing problems by the following approach: we exhaustively explore every possible order of prices, given which we solve the pricing problem. By comparing or verifying the results for all these possible orders, we get the final pricing solution. In particular, for CPP, we can solve $v_j = \frac{\hat{R}_j V}{N r_j(j)}$ for any given price order. If the solution $v_j, \forall j$ satisfies the preset order, then it is a solution to CPP. Similarly, for SPP, we can solve the above convex optimization (15) for any given price order; these pricing schemes are the optimal candidates among which we select the one that minimizes the operational cost.

Trying every possible order of prices can be computationally expensive, in particular, when the number of videos is large. However, one can categorize the videos of similar popularity into one class, and design the pricing strategy based on this limited number of classes.

Example. We use the following example to illustrate the procedure. Due to page limit, we only present CPP. We have two video classes: popular and unpopular, and peers' storage capacity is 2. Based on Fig. 2 and balancing equations, we can calculate the fraction of peers in each state, which are

$$r_1(1) = \frac{p_0 p_1 + 2p_1^2}{(p_0 + p_1)^2},$$

$$r_2(2) = \frac{p_0 p_2}{(p_0 + p_1 + p_2)^2} \frac{p_0^2 + 3p_1^2 + 4p_0 p_1 + 2p_0 p_2 + 3p_1 p_2}{(p_0 + p_1)^2}.$$

We consider the following two cases.

Case 1: We set $N = 10{,}000, V = 1, \text{and } p_0 = 0.2$, The two class popularities are 0.6 and 0.2, and the corresponding $\hat{R}_j$ are 2,000 and 1,000, respectively. We can verify there does not exist any solution to CPP: no matter we set $p_1 = 0.6$, $p_2 = 0.2$, (where we have $r_1(1) = 1.31, r_2(2) = 0.13$) or $p_1 = 0.2, p_2 = 0.6$ (where we have $r_1(1) = 0.75, r_2(2) = 0.69$), the prices p_1 and p_2 calculated from $v_j = \frac{\hat{R}_j V}{N r_j(j)}$ do not satisfy $v_1 \geq v_2$.

Case 2: We set $N = 10{,}000, V = 1, \text{and } p_0 = 0.6$, The two class popularities are 0.3 and 0.1, and the corresponding $\hat{R}_j$ are 1,000 and 500, respectively. We can verify there exists a unique solution to CPP: we set $p_1 = 0.1 \text{ and } p_2 = 0.3$ (where we have $r_1(1) = 0.16, r_2(2) = 0.40$), the prices are $v_1 = 0.31 \text{ and } v_2 = 0.25$, which satisfies $v_1 \geq v_2$.

This example is for illustration only. In fact, one can apply our methodology to any nonasymptotic analysis.

5 Generalizations and Extensions

In the previous section, we have analyzed the conservative and strategic pricing problems under the asymptotic system and showed some important implications. In real system implementations, our simple model may need to be extended so as to be adaptive to some practical issues. We deal with the following problems in this section:

- What is the impact if some peers do not watch a video but cache the data only for sake of reward?
- How to deal with a general price sensitivity model, i.e., the fraction of peers willing to cache the video is not necessarily proportional to the price?

In what follows, we extend our model to answer the above questions, respectively. For the simplicity of presentation, we make each extension separately on the basis of the simple model in Section 3. However, readers may note that these extensions are independent and can be easily combined in practical system designs.

5.1 Viewing-Caching Decoupling

In the previous analysis, we assume that a peer caches a video only if it watched the video previously. In reality, some intelligent peers may cache the video data only for the sake of earning reward, even if they are not interested in watching it at all. In such a case, the viewing and caching behaviors of a peer can be totally decoupled. Such behavior can impact on the distribution of replicas in the system.

We call the peers as "ordinary peers" if they cache the data only if they watched the video (which we discussed in the previous sections), and we call those as "intelligent peers" if they totally decouple the viewing and caching behaviors. Let α be the fraction of ordinary peers in the system, and $1-\alpha$ be the fraction of intelligent peers. For ordinary peers, the number of replicas cached for V_j is $\alpha \sum_{m=1}^{M} l_{mj} v_m$, where $l_{mj} = \frac{N}{V}(r_m(j) - r_{m-1}(j))$ as defined in

Section 2. For intelligent peers, the replica distribution does not depend on the video popularity, but only depends on the video prices. If a video is with a higher price, then it is expected to have more replicas in the system. In here, we assume a simple linear model, i.e., the number of replicas for V_j is proportional to its price v_j, and let it be $\beta \frac{v_j}{V} N$ where β is the coefficient. Then, the number of replicas in the whole system is

$$R_j(\boldsymbol{v}) = \alpha \sum_{m=1}^{M} l_{mj} v_m + \beta \frac{v_j}{V} N. \tag{16}$$

Taking this formula into the definition of conservative and strategic pricing problems, we can solve the prices accordingly. In particular, we still focus on the asymptotic system.

Theorem 5. *Under Assumption 1, a necessary and sufficient condition that CPP has a solution is* $p_0 \hat{R}_j \le (\alpha p_j + \beta p_0) N, \forall j$. *If this condition is satisfied, then the solution is*

$$v_j = \frac{p_0 \hat{R}_j V}{(\alpha p_j + \beta p_0) N}, \forall j.$$

Theorem 6. *There always exists a solution to SPP. The solution to SPP for an asymptotic system is* $v_j = \min\{\frac{1}{2} c_u, \frac{p_0 \hat{R}_j V}{(\alpha p_j + \beta p_0) N}, V\}$, $\forall 1 \le j \le M$.

A key observation is that when considering the intelligent peers, the "reverse order pricing" phenomenon may not apply any more. In particular, if peers are ordinary peers, i.e., $\alpha = 1$ and $\beta = 0$, then the solutions are the same as in Theorems 2 and 3 where the order of prices is the "reverse" order of popularity. However, if all peers are intelligent peers, i.e., $\alpha = 0$, then the order of prices is the "same" order of popularity. For general cases that lie in between, the order of prices depends on the fraction of ordinary/intelligent peers, or the order of values of $\frac{p_0 \hat{R}_j V}{(\alpha p_j + \beta p_0) N}$.

5.2 General Sensitivity Model

In Section 2, we applied a linear sensitivity model to characterize the impact of pricing on the peers' type distribution. In real systems, the sensitivity model can be complicated. We will show that under realistic assumptions, one can still effectively find the desired pricing schemes. Assume that given any nonnegative price v_j for video V_j, a fraction $f(v_j)$ of peers ($f(v_j) \in [0,1]$) are willing to cache this video provided that they watched it. Based on this model, the number of replicas for V_j becomes

$$R_j(\boldsymbol{v}) = N \sum_{m=0}^{M} (f(v_m) - f(v_{m+1})) r_m(j) = V \sum_{m=1}^{M} l_{mj} f(v_m), \tag{17}$$

where $l_{mj} = \frac{N}{V}(r_m(j) - r_{m-1}(j))$ as defined in Section 2. Correspondingly, the conservative pricing problem becomes

$$\hat{R}_j = V \sum_{m=1}^{M} l_{mj} f(v_m), \quad v_m \ge 0, \quad \forall j = 1, \ldots, M. \tag{18}$$

The strategic pricing problem becomes

$$\begin{aligned} \min \quad & c_u \sum_{j=1}^{M} \left(\hat{R}_j - V \sum_{m=1}^{M} l_{mj} f(v_m) \right)^{+} + V \sum_{j=1}^{M} \sum_{m=1}^{M} v_j l_{mj} f(v_m), \\ \text{s.t.} \quad & \boldsymbol{v} \ge 0. \end{aligned} \tag{19}$$

In real systems, the sensitivity function can be complicated, and system designers need to perform various measurements to approximate this function. How to observe this sensitivity function is beyond the scope of this paper. Here, we make the following assumption for the sensitivity function.

Assumption 2. $f(v)$ *is a concave, continuous, and nondecreasing function in* v. *Furthermore,* $f(0) = 0$, $f(\infty) = 1$.

Assumption 3. $v f(v)$ *is a convex, continuous, and increasing function in* v.

The above assumptions are realistic in practical systems: Assumption 2 means the fraction of peers willing to cache increases sublinearly with respect to the price, implying the diminishing return to scale effect observed in many similar economic scenarios. Assumption 3 means the total reward cost increases superlinearly with respect to the price (or unit reward), since the fraction of peers receiving the reward increases with respect to the price. Based on these assumptions, we will show that it is still efficient to get the solutions to CPP and SPP. In the following theorem, we first solve CPP.

Theorem 7. *Under Assumptions 1 and 2, a necessary and sufficient condition that CPP for an asymptotic system has a solution is* $p_0 \hat{R}_1 \le p_1 N$. *If this condition satisfies, then the solution is* $v_j = f^{-1}(\frac{p_0 \hat{R}_j}{p_j N})$, *where* $f^{-1}(\cdot)$ *is the reverse function of* $f(v_j)$.

The next theorem shows that SPP is a convex optimization problem, which could be solved efficiently.

Theorem 8. *Under Assumptions 1, 2, and 3, SPP for an asymptotic system is a convex optimization.*

The above two theorems indicate that, even under general sensitivity models, it is still efficient to calculate the optimal prices for the conservative/strategic pricing problems. Hence, our pricing scheme is practical in real system designs.

In the closing of this section, we point out that we also consider the pricing strategy before the system reaches the steady state. Due to page limit, we provide the material in the online supplemental material.

6 Performance Evaluation

In this section, we use simulation-based experiments to evaluate the performance of our pricing schemes. In the simulation, we have fixed number of videos and peers. Time is divided into T slots. In each slot, each peer randomly chooses a particular video with a certain probability (i.e., video popularity); each peer randomly leaves the system with probability p_0, and if this happens, we have a new peer

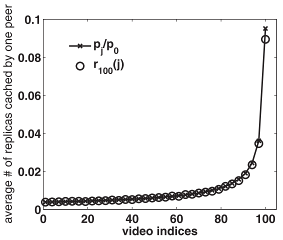


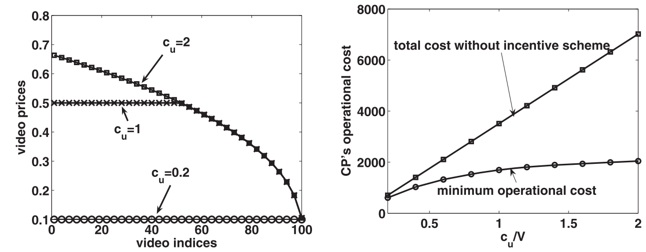
Fig. 3. Verification of Theorem 1.

joining the system with refreshed (thus empty) storage. In each case, we run the simulation for $T = 100$ slots and show the average value in the figures. We consider a fully connected P2P system and omit the effect of various issues (e.g., firewalls). We first run the simulations based on the basic model in Section 3 (see Figs. 3, 4, and 5), and then extend our results in general cases (see Fig. 6).

In particular, we have the following settings:

- We set $N = 10{,}000$ peers and $M = 100$ video classes. This is to model a medium scale P2P-VoD system.
- Each peer can cache up to $C = 6$ videos, i.e., a few GB storage space which a normal peer can afford.
- In each slot, a peer has a probability $p_0 = 0.5$ to refresh its storage.
- Videos' popularity p_j follows a Zipf distribution with parameter $\gamma = 0.7$. We apply this setting because Zipf distribution is observed in many video popularity measurements.
- The desired number of replicas $\hat{R}_j$ follows a Zipf distribution with parameter $\gamma = 0.3$, and maximum value $\hat{R}_{100} = 100$. We apply this setting to cope with the shape of p_j and to follow Assumption 1.
- We normalize the maximum price $V = 1$.

We apply the pricing mechanism derived for the asymptotic system, and evaluate the effectiveness of conservative and strategic pricing schemes. We first validate our result in Theorem 1. In Fig. 3, we plot the average number of replicas of V_j (i.e., $r_M(j)$) cached by a single peer of type $M = 100$. In comparison, we also plot the value p_j/p_0 for each video V_j. From the figure, we can verify $r_M(j)$ is very near to p_j/p_0 under the simulation settings. Note that there is a bit gap between $r_M(j)$ and p_j/p_0 for the few most popular videos. This is due to the effect of replacement: when the peers reach their storage capacity and perform replacement, the videos with larger indices (or low prices) are more easily to be replaced. Replacement is not considered in the asymptotic case, and hence leads to this difference.



(a) Video prices (b) Cost with/without incentive

Fig. 5. Strategic pricing scheme.

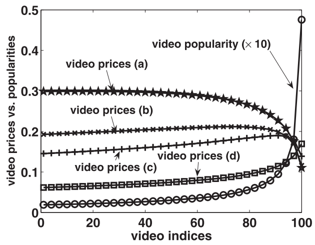
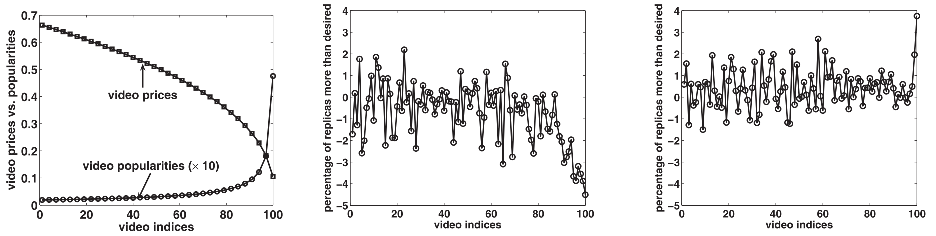


Fig. 6. Viewing-caching decoupling.

We next apply the conservative pricing scheme, i.e., $v_j = \frac{p_0 \hat{R}_j}{p_j N} V$. In Fig. 4a, we plot the video popularities versus the prices proposed for each video. It shows that the order of prices is the reverse order of popularity. We also compare the number of replicas desired (i.e., $\hat{R}_j$) versus cached (i.e., R_j) in the system using the conservative pricing scheme. In Fig. 4b, we plot $(R_j - \hat{R}_j)/\hat{R}_j$, i.e., the relative difference between R_j and $\hat{R}_j$. A positive value represents the percentage of replicas cached in the system but is more than desired, while a negative value indicates the deficit. The figure shows that using the conservative pricing scheme, the number of replicas for each video is very close to the value desired. We also note that the popular videos lack a few percentage of replicas due to the similar reasons stated above. A natural way to fill up this gap is to propose a bit higher prices for the popular videos. We apply a heuristic amendment by setting $v_j' = \frac{p_0 \hat{R}_j V}{(p_0 + p_j) N}$, and the corresponding result is shown in Fig. 4c. We can see that using the heuristic amendment, the number of replicas cached by peers is a bit more than desired for the popular videos.

We also apply the strategic pricing scheme, i.e., $v_j = \min\{\frac{c_u}{2}, \frac{p_0 \hat{R}_j}{p_j N} V, V\}$. The unit upload cost c_u can have a major impact on the prices. In Fig. 5a, we apply three typical values of c_u: $c_u = 0.2$, 1, and 2, and plot the prices which solve the strategic pricing problem. When c_u is



(a) Price vs. popularity (b) $\frac{R_j - \hat{R}_j}{\hat{R}_j}$ in **CPP** (c) $\frac{R_j - \hat{R}_j}{\hat{R}_j}$ using amendment

Fig. 4. Conservative pricing scheme.

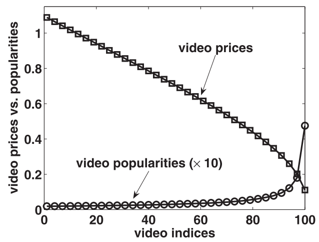


Fig. 7. Nonlinear sensitivity model.

small, i.e., the unit upload cost of the content server is small, the prices are upper bounded by $\frac{c_u}{2}$; whereas when c_u is large, the solution to the strategic pricing problem is the same as the conservative pricing problem. In Fig. 5b, we vary $c_u \in [0.2, 2.0]$, and compare the content provider's operational cost using the strategic pricing scheme versus that without using any incentive scheme. The figure shows that when the upload cost is high, the strategic pricing scheme earns a high cost reduction, which validates the effectiveness of our incentive scheme.

In Fig. 6, we illustrate a major difference on the order of prices when the system consists of certain amount of "intelligent" peers that decouple viewing and caching behaviors. In particular, we consider CPP and the prices are determined according to $v_j = \frac{p_0 \hat{R}_j V}{(\alpha p_j + \beta p_0) N}, \forall j$. We set (a) $\alpha = 0.9, \beta = 0.002$, (b) $\alpha = 0.8, \beta = 0.01$, (c) $\alpha = 0.6, \beta = 0.015$, and (d) $\alpha = 0.2, \beta = 0.04$, respectively. We can observe that when the majority are ordinary peers, the order of prices is the *reverse* order of popularity (see "a"); on the other hand, if there are a number of intelligent peers, then the order of prices is the *same* order of popularity (see "d"). In the cases in between, the price may not be monotonic with respect to the change of popularity (see "b" and "c"). We also consider the nonlinear sensitivity model, in particular, we assume $f(v) = 1 - e^{-v}$. According to previous analysis, the solution to CPP is $-\ln(1 - (\frac{p_0 \hat{R}_j}{p_j N}))$ which we plot in Fig. 7. We observe a similar trend as in Fig. 4a; the differences are 1) the largest price is not upper bounded by V; and 2) the decreasing trend of the prices seems more like linear.

7 Related Work

There have been number of research works on incentive issues for P2P systems. In [24], the authors presented a general framework to characterize the system performance and robustness for a class of adaptive incentive protocols. Park and Van Der Schaar [17] proposed a game theoretic framework and analyzed the content production and sharing model under different incentive schemes in a P2P network. Several particular methods were proposed for incentive schemes. Based on the peers' historical contributions to the community, service differentiation models [10], [14], and reputation systems [9], [11] were proposed to provide incentives. Aperjis et al. [1] and Freedman et al. [7] proposed a multilateral exchange scheme for content distribution networks. Recently, Misra et al. [15] proposed a Shapley value approach which serves as a new mechanism for incentives in P2P systems.

While earlier works [5], [6], [8] are mainly for file sharing systems, recent works have been focusing on P2P streaming/VoD systems, for example, modified tit-for-tat protocol [16], [18], punishment-based [13] and reward-based [4], [22] mechanisms were proposed to provide incentives. Recently Wu et al. [19] proposed an auction-based incentive protocol for P2P VoD streaming. These works incentivized the peers to provide their upload bandwidth to serve other peers, but did not address how to ensure the peers have cached the proper data so that they are able to upload. In [20], [21], [25], the authors discussed the replication strategies in P2P-VoD systems; nevertheless, up till now, we have not found any work on incentivizing the peers in P2P-VoD systems to participate in the distributed caching. Our work differs from the previous in that our incentive scheme stimulates the peers to contribute the local storage resources and cache the desired data so that they can effectively upload the content to one another in P2P-VoDs.

8 Conclusion

In this paper, we propose a reward-based incentive mechanism to stimulate the peers to contribute their local storage resources in a P2P-VoD system. In particular, the content provider proposes a price for each video. The most interesting finding is that, the order of video prices should be the reverse order of video popularity. More precisely, in an asymptotic system, we can get closed-form solutions to the conservative/strategic pricing problems that keep enough video replicas in the system and minimize the content provider's operational cost. We also show that our pricing scheme can be adaptive to various system environments, for examples, viewing-caching decoupling peers, the nonlinear price sensitivity model, nonasymptotic system, and highly dynamic video popularity. We evaluate the performance and validate the effectiveness of our incentive scheme via extensive simulations.

There are a number of issues to consider for our future work. First, as we mentioned earlier, incentive designs include local storage and upload bandwidth. We have analyzed the latter issue in another paper [22], but it remains an open question on how to perfectly combine these two aspects of design, and how to incentivize distributed caching in lack of bandwidth. Second, it is interesting (yet difficult) to learn the real sensitivity function of peers in practical systems so as to make the incentive scheme more profitable. Third, peers may be classified into different "taste groups" according to their intrinsic preference toward the video contents, and based on this classification, different groups of peers have certain watching preference, so an incentive scheme can be specialized based on the group. Last but not the least, the P2P-VoD system is highly dynamic, not only in video popularity, but also in peers' churn and their type distribution. The content provider may not have complete information all the time, and it is important yet challenging to learn the feature of system dynamics and adjust the pricing scheme correspondingly.

Acknowledgments

The research work of John C.S. Lui was supported in part by the Hong Kong GRF grant 2150634 (Ref# 415309).

References

[1] C. Aperjis, M.J. Freedman, and R. Johari, "Peer-Assisted Content Distribution with Prices," *Proc. ACM CoNEXT Conf. (CoNEXT)*, 2008.

[2] M. Benaïm and J.-Y.L. Boudec, "A Class of Mean Field Interaction Models for Computer and Communication Systems," *Performance Evaluation*, vol. 65, pp. 823-838, 2008.

[3] J.-Y.L. Boudec, D. McDonald, and J. Mundinger, "A Generic Mean Field Convergence Result for Systems of Interacting Objects," *Proc. Fourth Int'l Conf. Quantitative Evaluation of Systems (QEST)*, 2007.

[4] Y.-F. Chen et al., "Towards Capacity and Profit Optimization of Video-on-Demand Services in a Peer-Assisted IPTV Platform," *Multimedia System*, vol. 15, no. 1, pp. 19-32, 2009.

[5] B. Cohen, "Incentives Build Robustness in Bittorrent," *Proc. First Workshop Economics of Peer-to-Peer Systems*, 2003.

[6] M. Feldman, C. Papadimitriou, J. Chuang, and I. Stoica, "Free-Riding and Whitewashing in Peer-to-Peer Systems," *IEEE J. Selected Areas in Comm.*, vol. 24, no. 5, pp. 1010-1019, May 2006.

[7] M.J. Freedman, C. Aperjis, and R. Johari, "Prices are Right: Managing Resources and Incentives in Peer-Assisted Content Distribution," *Proc. Seventh Int'l Conf. Peer-to-Peer Systems (IPTPS)*, 2008.

[8] P. Golle, K. Leyton-Brown, and I. Mironov, "Incentives for Sharing in Peer-to-Peer Networks," *Proc. Third ACM Conf. Electronic Commerce (EC)*, 2001.

[9] M. Gupta, P. Judge, and M. Ammar, "A Reputation System for Peer-to-Peer Networks," *Proc. 13th Int'l Workshop Network and Operating Systems Support for Digital Audio and Video (NOSSDAV)*, 2003.

[10] A. Habib and J. Chuang, "Service Differentiated Peer Selection: An Incentive Mechanism for Peer-to-Peer Media Streaming," *IEEE Trans. Multimedia*, vol. 8, no. 3, pp. 610-621, June 2006.

[11] S.D. Kamvar, M.T. Schlosser, and H. Garcia-Molina, "The Eigentrust Algorithm for Reputation Management in P2P Networks," *Proc. 12th ACM Int'l Conf. World Wide Web (WWW)*, 2003.

[12] F.P. Kelly, *Reversibility and Stochastic Networks.* John Wiley and Sons, 1979.

[13] Z. Liu, Y. Shen, S.S. Panwar, K.W. Ross, and Y. Wang, "Using Layered Video to Provide Incentives in P2P Live Streaming," *Proc. Workshop Peer-to-Peer Streaming and IP-TV (P2P-TV)*, 2007.

[14] R.T.B. Ma, S.C.M. Lee, J.C.S. Lui, and D.K.Y. Yau, "Incentive and Service Differentiation in P2P Networks: A Game Theoretic Approach," *IEEE/ACM Trans. Networking*, vol. 14, no. 5, pp. 978-991, Oct. 2006.

[15] V. Misra, S. Ioannidis, A. Chaintreau, and L. Massoulié, "Incentivizing Peer-Assisted Services: A Fluid Shapley Value Approach," *Proc. ACM SIGMETRICS*, 2010.

[16] J. Mol, J. Pouwelse, M. Meulpolder, D. Epema, and H. Sips, "Give-to-Get: Free-Riding-Resilient Video-on-Demand in P2P Systems," *Proc. Ann. Multimedia Computing and Networking (MMCN)*, 2008.

[17] J. Park and M. Van Der Schaar, "A Game Theoretic Analysis of Incentives in Content Production and Sharing over Peer-to-Peer Networks," *IEEE J. Selected Topics in Signal Processing*, vol. 4, no. 4, pp. 704-717, Aug. 2010.

[18] F. Pianese, D. Perino, J. Keller, and E.W. Biersack, "Pulse: An Adaptive, Incentive-Based, Unstructured P2P Live Streaming System," *IEEE Trans. Multimedia*, vol. 9, no. 8, pp. 1645-1660, Dec. 2007.

[19] C. Wu, Z. Li, X. Qiu, and F.C.M. Lau, "Auction-Based P2P VoD Streaming: Incentives and Optimal Scheduling," *ACM Trans. Multimedia Computing, Comm., and Applications*, vol. 8, no. 1S, p. 14, 2012.

[20] J. Wu and B. Li, "Keep Cache Replacement Simple in Peer-Assisted VoD Systems," *Proc. IEEE INFOCOM*, 2009.

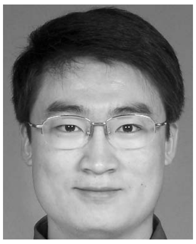
[21] W. Wu and J.C.S. Lui, "Exploring the Optimal Replication Strategy in P2P-VoD Systems: Characterization and Evaluation," *IEEE Trans. Parallel Distributed Systems*, vol. 23, no. 8, pp. 1492-1503, Aug. 2012.

[22] W. Wu, J.C.S. Lui, and R.T.B. Ma, "Incentivizing Upload Capacity in P2P-VoD Systems: A Game Theoretic Analysis," *Proc. Second Int'l Conf. Game Theory for Networks (GameNets)*, 2011.

[23] W. Wu, R.T.B. Ma, and J.C.S. Lui, "On Incentivizing Caching for P2P-VoD Systems," *Proc. Workshop the Economics of Networks, Systems, and Computation (NetEcon)*, 2012.

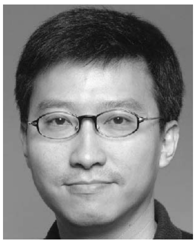
[24] B.Q. Zhao, J.C.S. Lui, and D.-M. Chiu, "Analysis of Adaptive Incentive Protocols for P2P Networks," *Proc. IEEE INFOCOM*, 2009.

[25] Y. Zhou, T.Z.J. Fu, and D.-M. Chiu, "Statistical Modeling and Analysis of P2P Replication to Support VoD Service," *Proc. IEEE INFOCOM*, 2011.



Weijie Wu received the BSc degree in electronics from Peking University in 2008 and the PhD degree in computer science from The Chinese University of Hong Kong in 2012. He is currently an assistant professor at the School of Information Security Engineering, Shanghai Jiao Tong University. Before that, he was a postdoctoral research fellow in the Department of Computer Science and Engineering at the Chinese University of Hong Kong. During his PhD study, he worked as a research intern at the National University of Singapore. His current research interests include computer networks and security, including incentive mechanism design, application of game theory on communication networks and security issues, performance evaluation, and network economics.

Richard T. B. Ma received the BSc (first-class honors) degree in computer science and the MPhil degree in computer science and engineering from the Chinese University of Hong Kong, in July, 2002 and 2004, respectively, and the PhD degree in electrical engineering from Columbia University, in May, 2010. During his PhD study, he worked as a research intern at IBM T.J. Watson Research Center in New York and Telefonica Research in Barcelona. He is currently a research scientist in the Advanced Digital Science Center, University of Illinois at Urbana-Champaign and an assistant professor in the School of Computing at the National University of Singapore. His current research interests include distributed systems, network economics, game theory, and stochastic processes.



John C.S. Lui received the PhD degree in computer science from the University of California, Los Angeles. He is currently a professor in the Computer Science and Engneering Department at the Chinese University of Hong Kong (CUHK). His research interests include data networks, system and applied security, multimedia systems, network sciences, and cloud computing. He received various departmental teaching awards and the CUHK Vice-Chancellor's Exemplary Teaching Award, as well as the corecipient of the Best Student Paper Awards in the IFIP WG 7.3 Performance 2005 and the IEEE/IFIP Network Operations and Management (NOMS) Conference. He is an associate editor in the*Performance Evaluation Journal*, *IEEE-TC*, *IEEE-TPDS*, and *IEEE/ACM Transactions on Networking*. He is a fellow of the IEEE and the ACM, and the chair of ACM SIGMETRICS.