

On Incentivizing Caching for P2P-VoD Systems

Weijie Wu*, Richard T.B. Ma^{§+} and John C.S. Lui*

*The Chinese University of Hong Kong

§National University of Singapore

+ Advanced Digital Sciences Center, Illinois at Singapore

Email: wjwu@cse.cuhk.edu.hk, tbma@comp.nus.edu.sg, cslui@cse.cuhk.edu.hk

Abstract—Peer-to-peer (P2P) systems consist of selfish peers which 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. This issue is particularly important for P2P Video-on-Demand (VoD) services. Content providers of P2P-VoD services want to incentivize peers to cache video data so that the peers can transfer data to one another and alleviate the workload of the content server. In this paper, we design a practical mechanism 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. Via extensive simulations, we validate the effectiveness and efficiency of our incentive scheme.

I. Introduction

The adoption of Peer-to-Peer (P2P) technologies into Video-on-Demand (VoD) systems has been greatly successful in the recent years. 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 the data to one another when they have available upload bandwidth. 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 are selfish in nature and would not be willing to contribute their resources by default. Hence, designing an effective incentive scheme is critical. Unlike traditional P2P applications in which plenty of incentive schemes have been proposed, very limited work has been focusing on the P2P-VoD applications. In P2P-VoDs, traditional built-in incentives like tit-for-tat are not applicable due to lack of mutual upload between peers. Because P2P-VoD system is large scale and stochastic in nature, one can hardly predict, or even describe the peers' requests and resources at a particular time. Unlike the P2P file sharing system where peers are able to download a file from a single replica, P2P-VoDs need to keep *enough* replicas so as to guarantee the download rate to satisfy peers' viewing requirements. Moreover, peers in P2P-VoDs are heterogeneous and may have various responses to particular incentive schemes. All the above make the design of an effective incentive mechanism for P2P-VoDs very challenging.

Each peer needs to be incentivized to contribute (1) their local storage space to cache video data, and (2) their upload

bandwidth for uploading data to other peers. Both of the above are equally important because a peer cannot contribute if it fails to contribute either of them. Our previous work [22] has been focusing on incentivizing the upload bandwidth of peers. In this paper, we focus on the incentive scheme that stimulates peers to cache the needed video data. We propose a reward-based incentive scheme for caching, where the content provider decides the *reward price* of each video and peers decide what videos to cache in a distributed manner. In particular, we apply a *mean field model* to characterize the steady-state of the caches in a large scale P2P-VoD system. Based on this, we formulate a pricing problem using an optimization framework and solve the optimal prices that minimize the content provider's operational cost. Although most P2P-VoDs are free for ordinary service, our pricing scheme is practical as the reward can be in forms of credits or service fee rebate for premium services (e.g. high definition channels) for which commercial systems do charge users. Our contributions are:

- We develop a stochastic model to characterize the peers' caching behaviors, and use the mean field technique to characterize the system state in a limiting steady state.
- We formulate an optimal pricing problem of the content provider and derive the optimal reward prices for each video. Our scheme incentivizes the peers to cache enough replicas of various videos in the system and minimizes the content provider's operational cost.
- We validate the effectiveness of our incentive scheme by extensive simulations.

We organize the paper as follows. In Section II, we develop a mathematical model to characterize the system's cache state, and present an optimization framework for the pricing scheme. In Section III, we analyze the pricing schemes in a practical asymptotic case where we give closed-form solutions for the optimal pricing schemes. We perform extensive simulations to evaluate our pricing schemes in Section IV. Section V states related work and conclusions.

II. Model

A. Preliminaries

We consider a P2P-VoD system which consists of peers and videos. In a realistic P2P-VoD 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 .

In order to design an effective incentive mechanism for caching, we first characterize the system's state in caching each video. A straight-forward way is using a stochastic model where the system's state is determined by *all* peers' cache states. However, describing such a system can be computationally expensive, in particular, when the number of peers is large, the state space of the system becomes huge and thus intractable. To overcome this difficulty, we model the peers in the system 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 explain the applicability of this technique to our problem in our technical report [23].

We denote p_0 as the probability that a peer refreshes its local storage with an empty cache space after finishing watching a video. This could happen for various reasons: this peer may restart or update the software, or its cached data are not popular any longer. A peer can also choose the next video to watch. Denote p_j as the probability that it chooses a video 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 reward-based scheme, the content provider provides an external reward of v_j (dollars for example) for each video of V_j cached. We call v_j the *price* of V_j and define the price vector as $\mathbf{v} = (v_1, v_2, \dots, v_M)$. Our core design problem is to decide the optimal prices for videos from the content provider's point of view. For any fixed price vector, without loss of generality, we index the videos in a non-increasing order of prices, i.e., $v_1 \geq \dots \geq v_M$. Later on, we will explore its relationship with the order of popularity of the videos.

Each peer decides whether or not to cache the video it watches 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 more about their local storage costs so that they do not want to cache even the highest-priced video, while peers of type M are the most insensitive to their storage costs and are willing to cache any video. In what follows, we first characterize the caching behaviors of the peers of various types, and then explore the impact of the pricing scheme on the distribution of the number of video replicas in the whole system.

B. 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. 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. Formally, we use an $m+1$ dimensional vector $\mathbf{s} = (s_0, s_1, \dots, s_m)$ to represent the cache state of a peer of type m . 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 $S_m = \{\mathbf{s} : \sum_{j=0}^m s_j = C, s_j \geq 0\}$. We let $\mathbf{s} = (s_0, \dots, s_m) = \sum_{j=0}^m s_j \mathbf{e}_j$, where \mathbf{e}_j is a vector with the j th element being 1 and all other elements being 0. We define $t(\mathbf{s}) = \operatorname{argmax}_i \{i : s_i > 0\}$ as the largest class index of the videos cached in state \mathbf{s} . We define a *deletion* operation $\mathbf{d}(\mathbf{s})$ on state \mathbf{s} as

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

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 (also the lowest price) in \mathbf{s} .

Based on the above notation, we illustrate the cache state transition diagram in Fig. 1. Given a current state \mathbf{s} , three

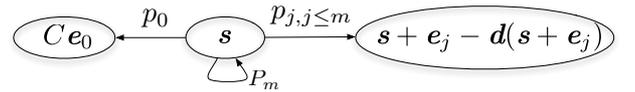


Fig. 1: Transition diagram

types of transitions can happen to a peer. First, with probability p_0 , the peer refreshes its local storage and transits to state $C\mathbf{e}_0$, where all its C units of storage become empty. Second, with probability $P_m = \sum_{j=m+1}^M p_j$, which is the aggregate probability that the peer watches any video from the set $\{V_j : j > m\}$, the cache state remains the same, since the peer watches some video that it is not going to cache. Third, with 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.

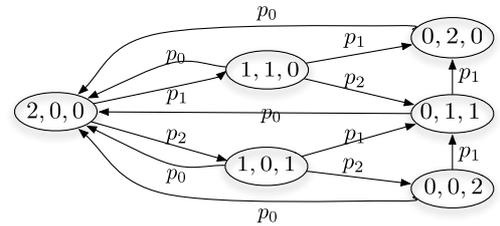


Fig. 2: An example of cache state transitions

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

C. Cache state distribution of peers of type m

In this subsection, we use the mean field model to approximate the fraction of peers in each state. We denote $q_m(\mathbf{s})$ as the

fraction of type m peers that are in state \mathbf{s} in a steady state¹. 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 [11].

We use the global balance equations [11] 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, respectively, and derive $q_m(\mathbf{s})$ separately for the three cases. For the state $\mathbf{s} = C\mathbf{e}_0$ that represents the empty storage, we have

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

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

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

Similarly, for any state \mathbf{s} with $s_0 > 0$ that represents a partially occupied cache, we have the following balance equation:

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

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

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

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

Lastly, for a state \mathbf{s} with $s_0 = 0$ that represents a full storage, the corresponding global balance equation is

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

where $K(\mathbf{s}) = \{k : k = 0 \text{ or } t(\mathbf{s}) \leq k \leq m\}$. To calculate the above $q_m(\mathbf{s})$, we sort the states $\{\mathbf{s} : s_0 = 0\}$ in an increasing

¹In reality, the video popularity is also time varying, but its changing rate is typically much slower than that of the caching states. Hence, in most cases, we can approximate the system's state by calculating the steady state distribution using the current video popularity. We also note it worth an in-depth discussion for the boosting interest at a new publishing instance.

order on the sequence $(s_1 s_2 \dots s_m)$. For the example in Fig. 2, the states are sorted as $(0, 0, 2), (0, 1, 1), (0, 2, 0)$. Then we can solve Eq. (6) for the states according to the sorted order so that the right hand side quantities will already be available.

Denote N as the total number of peers in the system and N_m as the number of peers of type m . Based on the cache state distribution $\{q_m(\mathbf{s}) : \mathbf{s} \in S_m\}$, denote $r_m(j)$ as the per peer average number of videos of V_j cached by type m peers:

$$r_m(j) = \sum_{\mathbf{s} \in S_m} s_j q_m(\mathbf{s}). \quad (7)$$

D. Cache state of the system

Based on the cache state distribution $\{q_m(\mathbf{s}) : \mathbf{s} \in S_m\}$, we now derive the cache state of the entire P2P-VoD system. The system state depends on the number of peers of each type, which, in real systems, is further determined by the prices of the videos. For example, if v_M is large, then more peers would be of type M ; if v_1 is small, then more peers would 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 our technical report [23], 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 under 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 non-negative price for videos; however, setting a price higher than V cannot be more beneficial than setting the price at V . Hence, we focus on the design space of $v_j \in [0, V], \forall j$. Under our linear model, by defining $v_0 = V$ and $v_{M+1} = 0$, we can express the number of peers of type m in the system as

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

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

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

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

$$R_j(\mathbf{v}) = \sum_{m=1}^M l_{mj} v_m. \quad (10)$$

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

We would like to mention that the above stationary cache state derived from the mean field limit is a result of optimal caching decisions made by individual peers. In fact, the

viewing preference of a type m peer is independent of the video prices, whereas its best caching decision is always to cache the top m highest-priced videos. Eq. (10) also 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 so as to achieve certain objectives for the system.

E. Design objectives of the pricing scheme

The content provider proposes the incentive scheme in order to reduce its operational cost. We formally 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 $\mathbf{v} = (v_1, \dots, v_M)$ that satisfies the following constraints:

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

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 $C(\mathbf{v})$ as the content provider's operational cost, which consists of an upload cost $C_u(\mathbf{v})$ as well as the reward cost $C_p(\mathbf{v})$ 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 $C_u(\mathbf{v}) = c_u \sum_{j=1}^M (\hat{R}_j - R_j(\mathbf{v}))^+$, 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 $C_p(\mathbf{v}) = \sum_{j=1}^M v_j R_j(\mathbf{v})$. Thus, the operational cost of the content provider $C(\mathbf{v})$ is

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

The strategic pricing strategy requires the content provider to find \mathbf{v} which minimizes its operational cost, i.e.,

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

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., $C_u(\mathbf{v}) \gg C_p(\mathbf{v}), \forall \mathbf{v}$. 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 $c_u \rightarrow \infty$.

III. 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: $(1 - p_0)^C \rightarrow 0$. Physically, the above condition means that the probability a peer keeps watching C videos without refreshing its local storage approaches zero. We explain the practicality of this condition in our technical report [23].

A. 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 peer of type m approaches p_j/p_0 when $(1 - p_0)^C$ approaches 0, i.e.,

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

Due to page limit, we omit all the proofs. Interested readers may refer to our technical report [23] for details.

This theorem provides a neat form for $r_m(j)$ and greatly simplifies the analysis for the asymptotic case. More importantly, it also points out the fact that under a limiting case, the average number of videos cached by a single peer is *proportional* to the popularity of the videos. We will validate the above result by simulations in later sections.

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.

B. Conservative and strategic 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. 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 of these two videos, \hat{R}_i and \hat{R}_j , satisfy the following condition: $p_i/p_j < \hat{R}_i/\hat{R}_j < 1$.

An underline 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 sub-linearly with respect to the video popularity. We have the following lemma:

Lemma 1: Under Assumption 1, if CPP in an asymptotic system has a solution \mathbf{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 popularities. 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, \dots, V_M are arranged in a non-increasing order of prices, and based on the above result, the popularity of the videos would be in a non-decreasing 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.

Now we show how to decide the value of each video.

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 $v_j = \frac{p_0 \hat{R}_j}{p_j N} V, \forall j$.

Next we show the solution to the strategic pricing problem.

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

Theorem 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 $(1-p_0)^C \rightarrow 0$, we can apply Theorem 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.

IV. Performance Evaluation

In this section, we use simulation-based experiments to evaluate the performance of our pricing schemes. In particular, we have the following settings:

- The system contains $N = 10,000$ peers and $M = 100$ kinds of videos. Each peer can cache up to $C = 6$ videos.
- A peer has a probability $p_0 = 0.5$ to refresh its storage.
- Popularity p_j follows a Zipf distribution with $\gamma = 0.7$.
- \hat{R}_j follows a Zipf distribution with $\gamma = 0.3$ and the maximum value $\hat{R}_{100} = 100$.
- The minimal price to ensure all peers willing to cache a particular video is 1, i.e., $V = 1$.

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 $M = 100$.

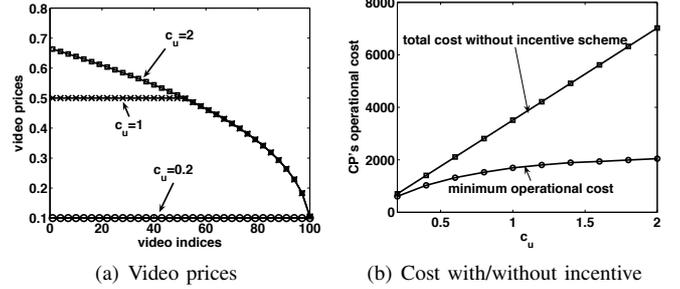
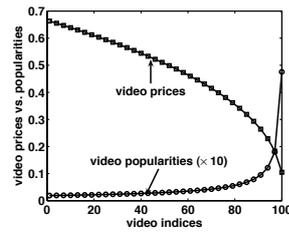
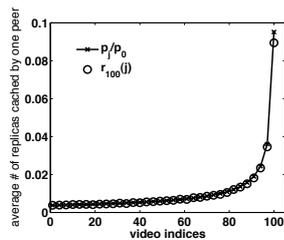


Fig. 5: Strategic pricing scheme

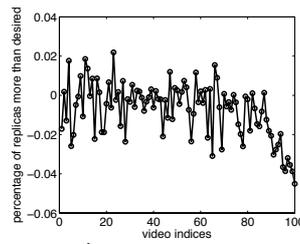
In comparison, we also plot the value p_j/p_0 for each video V_j . From the figure, we can verify $r_M(j)$ is very near to p_j/p_0 under the simulation settings. Note that there is a bit gap between $r_M(j)$ and p_j/p_0 for the few most popular videos. This is due to effect of replacement: when the peers 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.

We next apply the conservative pricing scheme, i.e., $v_j = \frac{p_j \hat{R}_j}{p_0 N} V$. In Fig. 4(a), we plot the video popularities vs. 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. 4(b), 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 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, in particular, for the many unpopular videos which has a long tail effect. 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 $v'_j = \frac{p_j \hat{R}_j V}{(p_0 + p_j) N}$, and the corresponding result is shown in Fig. 4(c). 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.

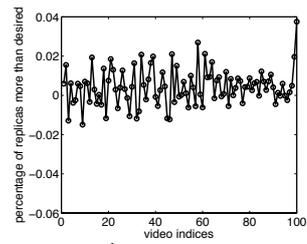
Lastly, we apply the strategic pricing scheme, i.e., $v_j = \min \left\{ \frac{c_u}{2}, \frac{p_0 \hat{R}_j}{p_j N} V, V \right\}$. The unit upload cost c_u can have a major impact on the prices. In Fig. 5(a), we apply three typical values of c_u : $c_u = 0.2, 1$ and 2 , and plot the prices which solve the strategic pricing problem. When c_u is 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. 5(b), we vary $c_u \in [0.2, 2.0]$, and compare the content provider’s operational cost using the strategic pricing scheme vs. 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.



(a) Video prices vs. video popularities



(b) $\frac{R_j - \hat{R}_j}{\hat{R}_j}$ in conservative pricing



(c) $\frac{R_j - \hat{R}_j}{\hat{R}_j}$ using amendment

Fig. 3: Verification of Theorem 1

Fig. 4: Conservative pricing scheme

V. Related Work and Conclusion

There have been number of research works on incentive issues for P2P systems, e.g., general framework [24], [16], service differentiation models [13], [9], reputation systems [8], [10], multilateral exchange systems [1] and Shapley value approach [14]. While the earlier works [7], [4], [6] are mainly for file sharing systems, recently some research works have been focusing on P2P streaming/VoD systems, e.g., modified tit-for-tat protocol [15], [17], punishment based [12] and reward based [22] mechanisms were proposed. These works incentivized the peers to upload and serve other peers, however, in a large scale distributed system, it is very hard for the peers to be smart enough to know what are the proper data that they should cache. The authors in [21], [25], [20] discussed the replication strategies in P2P-VoD systems but did not address the incentive issues.

Our work differs from all the existing results in that we propose a reward-based incentive mechanism for caching. A most interesting finding is that, the order of video prices should be the reverse order of video popularity. More precisely, we can get closed-form solutions to conservative/strategic pricing problems, and validate their effectiveness via simulations.

There are some issues for our future work. First, there might be a boosting increase of interest at a video's publishing instance. The steady state analysis may not apply to this specific scenario since the video popularity may have changed before reaching the steady state. It is interesting to discuss how to adjust the pricing scheme dynamically with the change of popularity. Second, a strategic peer may choose to download and cache a video only for the sake of earning reward, even if it is not interested in watching the video at all. It is worth for a discussion on the impact of such a strategic behavior, and whether it is beneficial from the system operator's point of view. Lastly, cheating-prevention mechanism is critical in implementing our incentive scheme, which we did not address in our paper. Existing literatures [5], [19], [18] have developed many enforcement methodologies to guarantee that the peers truthfully upload to others, and we can potentially use the similar ideas to ensure that the peers truthfully cache the correct data in the system under our incentive scheme.

Acknowledgement

The authors would like to thank the anonymous reviewers for their constructive comments and suggestions.

REFERENCES

- [1] C. Aperjis, M. J. Freedman, and R. Johari. Peer-assisted content distribution with prices. In *Proceedings of ACM CoNEXT*, 2008.
- [2] M. Benaïm and J.-Y. Le Boudec. A class of mean field interaction models for computer and communication systems. *Performance Evaluation*, 65:823–838, 2008.
- [3] J.-Y. Le Boudec, D. McDonald, and J. Mundinger. A generic mean field convergence result for systems of interacting objects. In *QEST*, 2007.
- [4] B. Cohen. Incentives build robustness in bittorrent. In *Proceedings of the 1st Workshop on Economics of Peer-to-Peer Systems*, 2003.
- [5] L. P. Cox and B. D. Noble. Samsara: honor among thieves in peer-to-peer storage. In *Proceedings of ACM SOSP*, 2003.
- [6] M. Feldman, C. Papadimitriou, J. Chuang, and I. Stoica. Free-riding and whitewashing in peer-to-peer systems. *IEEE Journal on Selected Areas in Communications*, 24(5):1010 – 1019, 2006.
- [7] P. Golle, K. Leyton-Brown, and I. Mironov. Incentives for sharing in peer-to-peer networks. In *Proceedings of ACM EC*, 2001.
- [8] M. Gupta, P. Judge, and M. Ammar. A reputation system for peer-to-peer networks. In *Proceedings of NOSSDAV*, 2003.
- [9] A. Habib and J. Chuang. Service differentiated peer selection: an incentive mechanism for peer-to-peer media streaming. *IEEE Transactions on Multimedia*, 8(3):610–621, 2006.
- [10] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina. The eigentrust algorithm for reputation management in P2P networks. In *WWW*, 2003.
- [11] F. P. Kelly. *Reversibility and stochastic networks*. Wiley NY, 1979.
- [12] Z. Liu, Y. Shen, S. S. Panwar, K. W. Ross, and Y. Wang. Using layered video to provide incentives in P2P live streaming. In *P2P-TV*, 2007.
- [13] 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 Transactions on Networking*, 14:978–991, October 2006.
- [14] V. Misra, S. Ioannidis, A. Chaintreau, and L. Massoulié. Incentivizing peer-assisted services: a fluid shapley value approach. In *Sigmetrics'10*.
- [15] J. Mol, J. Pouwelse, M. Meulpolder, D. Epema, and H. Sips. Give-to-get: Free-riding-resilient video-on-demand in P2P systems. In *MMCN'08*.
- [16] J. Park and M. van der Schaar. A game theoretic analysis of incentives in content production and sharing over peer-to-peer networks. *IEEE Journal of Selected Topics in Signal Processing*, 4(4):704–717, 2010.
- [17] F. Pianese, D. Perino, J. Keller, and E. W. Biersack. Pulse: An adaptive, incentive-based, unstructured P2P live streaming system. *IEEE Transactions on Multimedia*, 9(8):1645–1660, 2007.
- [18] M. Piatek, T. Isdal, A. Krishnamurthy, and T. Anderson. One hop reputations for peer to peer file sharing workloads. In *NSDI*, 2008.
- [19] M. Sirivianos, J. H. Park, X. Yang, and S. Jarecki. Dandelion: cooperative content distribution with robust incentives. In *Usenix ATC'07*.
- [20] J. Wu and B. Li. Keep cache replacement simple in peer-assisted vod systems. In *INFOCOM'09*.
- [21] W. Wu and J. C. S. Lui. Exploring the optimal replication strategy in P2P-VoD systems: Characterization and evaluation. In *INFOCOM'11*.
- [22] W. Wu, J. C. S. Lui, and R. T. B. Ma. Incentivizing upload capacity in P2P-VoD systems: a game theoretic analysis. In *GameNets'11*.
- [23] W. Wu, J. C. S. Lui, and R. T. B. Ma. On incentivizing caching for p2p-vod systems. Technical report. Available at www.cse.cuhk.edu.hk/~7Ewjwu/cachingTR.pdf
- [24] B. Q. Zhao, J. C. S. Lui, and D.-M. Chiu. Analysis of adaptive incentive protocols for P2P networks. In *Proceedings of 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. In *INFOCOM'11*.