Incentive Scheme for P2P Live Streaming Systems
Being Aware of the Upload Capability of the Participants

Shogo KANDA$^1$ and Satoshi FUJITA$^1$
$^1$Department of Information Engineering, Hiroshima University
Higashi-Hiroshima, 739-8527, Japan

Abstract—In this paper, we propose an incentive scheme for P2P live streaming systems being aware of the upload bandwidth of the participants. The basic idea of the scheme is to combine a point-based incentive scheme with the notion of minimum guaranteed services. The performance of the proposed scheme is evaluated by simulation. The result of simulation indicates that: 1) the proposed scheme increases the utility of poor peers by 30% compared with Sepidar which is a typical auction-based incentive scheme and 2) compared with the Chu’s scheme, which is a typical taxation-based scheme, the proposed scheme gradually increases the utility of rich peers as the amount of contributions increase, whereas the utility under the Chu’s scheme does not change after reaching the limit determined by the taxation rate.

Keywords: Peer-to-Peer live streaming systems, auction-based incentive scheme, taxation scheme.

1. Introduction

Recently, Peer-to-Peer (P2P) technology has been used in many fields as a way of realizing scalable network services. Voice over IP such as Skype$^1$ and video streaming services such as PPLive$^2$ and PPStream$^3$ are representatives of such applications. Among those applications, in this paper, we focus on the live streaming based on the P2P technology which is generally referred to as the P2P live streaming. In P2P live streaming systems, each peer (node) participating in the system is encouraged to contribute its communication bandwidth to the system, so that a copy of data received from an adjacent peer is uploaded to another adjacent peer by using the upload bandwidth. In other words, the system is designed so that the increase of the number of participants also increases the total amount of upload bandwidth.

However, the performance of such P2P systems is severely affected by the behavior of each participant, since it causes an overload of specific peers and the degradation of provided services if the number of free-riders which do not contribute to the system increases. To overcome such a situation, P2P systems should have an incentive scheme which strongly encourages the participants to provide their resources as much as possible. Conventional incentive schemes for P2P live streaming systems are designed so that a peer contributing to the system can receive a high quality service in return for the contribution. Such a mechanism works well for the peers to have enough resources. However, for the peers to have small amount of resources, it is not user-friendly since it (strictly) differentiates available services by the amount of resources held by the peer, i.e., it discourages many “poor” peers to participate in the system to contribute as an uploader.

In this paper, we propose an incentive scheme for P2P live streaming systems such that all contributors become happy. In other words, we determine the return of contribution so that every peer can receive a return even if the amount of contributions is small as long as it repeats contributions, and it can receive a larger return as the amount of contributions increases. The basic idea of the scheme is to combine a point-based incentive scheme with the notion of minimum guaranteed services. More concretely, we design the scheme so that: 1) the given live stream is divided into $k$ sub-streams which are delivered to the participants using different trees, where we assume the existence of an appropriate encoding method such that the original stream is decoded from any subset of sub-streams while the quality of the decoded stream depends on the number of sub-streams; 2) we differentiate the number of sub-streams acquired by each peer according to the amount of contributions; and at the same time, 3) we guarantee the minimum service for every peer by allowing peers to participate in at least $R_d$ trees, where the value of $R_d$ is dynamically updated according to the change of the status, such as the number of participants and the total amount of bandwidth.

The performance of the proposed scheme is evaluated by simulation. The result of simulation indicates that: 1) the proposed scheme increases the utility of poor peers by 30% compared with Sepidar which is a typical auction-based incentive scheme for P2P live streaming, and 2) compared with the Chu’s taxation scheme, which is a typical taxation-based scheme, the proposed scheme gradually increases the utility of rich peers as the amount of contributions increase, whereas the utility under the Chu’s scheme does not change after reaching the limit determined by the taxation rate.

The remainder of this paper is organized as follows. Section 2 describes preliminaries and Section 3 overviews related works. Section 4 describes the details of the pro-
posed scheme. Section 5 describes the result of simulations. Finally, Section 6 concludes the paper with future works.

2. Preliminaries

In P2P live streaming systems, a given stream is delivered to the subscribers through a logical network called P2P overlay. In this paper, we are particularly interested in the multi-tree structure used in SplitStream [1] as the underlying P2P overlay, because of the ease of the maintenance compared with mesh structured overlays and the high fault tolerance compared with a single tree. In multi-tree structured P2P live streaming systems, each stream is divided into several sub-streams called stripes, and those stripes are delivered through different trees. Figure 1 illustrates the delivery of a stream through multi-trees. The split of a stream into stripes is done by encoding schemes such as MDC [3], in such a way that: 1) any subset of stripes can be decoded into a stream and 2) the quality of the resulting stream monotonically increases as the number of stripes increases.

In order to decode a high quality stream, each peer tries to participate in as many trees as possible, while it can participate in at most one tree as an internal peer, where a peer with a child in a tree is called an internal peer and a peer with no child is called a leaf peer. In each tree, each internal peer forwards a stripe received from the parent to the children by using its upload bandwidth. In this paper, we assume that the upload of one stripe consumes one unit of upload bandwidth called upload slot, and we call the maximum number of upload slots used by a peer the fan-out of the peer. The reader should note that the fan-out of a peer is calculated by dividing the upload bandwidth of the peer by the bandwidth required for the upload of a stripe.

3. Related Work

3.1 Auction-Based Incentive Schemes

There are many auction-based incentive schemes proposed for tree-based P2P live streaming systems. Sepidar [5] and Tan’s scheme [8] are two representatives in this category.

The bidding procedure adopted in Sepidar is based on the notions of money and price. The money $m[i]$ held by peer $i$ is the (declared) fan-out of the peer. The price $p[i]$ of peer $i$ is set to be zero if it has an unused upload slot, and otherwise, $p[i]$ is set to be the minimum money over all children of $i$. For example, in Figure 2, the price of peer $b$ is two since it has three slots connected to children with money 2, 3 and 4, respectively. Suppose that a peer $a$ wishes to become a child of peer $b$. Then peer $a$ initiates an auction and bids all of its money $m[a]$ for an upload slot of $b$. If the bid received from $a$ is greater than $p[b]$, then peer $b$ accepts $a$ as a new child, and turns out a child with the least amount of money from the tree (by this operation, all descendants of the removed child will also be turned out from the tree and they should initiate another auction to subscribe to the stream). The reader should note that the above bidding mechanism gives incentives to all peers to declare a large fan-out so that they are not turned out by the other peers.

Tan’s scheme conducts such a bidding periodically. It divides the playback time of the video stream into periods of fixed length as $T_0, T_1, T_2, \ldots$, and during each period, each peer earns reward points from its children as a consideration for the upload of the video stream. More concretely, in the $\ell^{th}$ period, a peer $j$ receives all reward points from child $i$ which were earned by peer $i$ in the $(\ell - 1)^{st}$ period. Reward points earned at the beginning of the $\ell^{th}$ period are used for the bidding for an upload slot for the $(\ell + 1)^{st}$ period. The bidding within the $\ell^{th}$ period proceeds by repeating bidding round in the following manner:

1) Participants of each bidding round are peers which did not find the parent for the next period.
2) Each participant $a$ randomly selects an internal peer $b$ which has already found the parent and has enough upload slots, and bids all reward points earned at the beginning of the period for an upload slot of $b$.

3) Among submitted bids, peer $b$ selects bidders with the largest bids as the winners and makes them children in the next period.

4) All losers of the bidding receive a list of winners from peer $b$, so that it could be used as the candidate for internal peers in the next bidding round.

In the Tan’s scheme, each peer can participate in any number of trees as an internal peer, while after exhausting the upload slots, it should participate in each of the remaining trees as a leaf peer. This indicates that it may happen a situation such that a loser of an action cannot find a peer to have enough upload slot within a period. In such a case, it should try to find an internal peer by traversing the tree from the root to leaves in a best effort manner.

3.2 Taxation Scheme

Chu et al. point out that auction-based incentive schemes described in the last subsection would cause a significant gap of the quality of received services due to the difference of the amount of resources intrinsically held by each participant, and propose a taxation scheme to overcome this issue [2].

The key idea of the Chu’s taxation scheme is to combine the taxation with a fixed tax rate $t$ ($>1.0$) with the notion of basic income called demogrant. The maximum number of trees in which a peer can participate is a linear function of the amount of contribution of the peer. More concretely, to participate in $r_i$ different trees, peer $i$ must contribute at least $f_i = t \times r_i$ upload slots in a tree. By consuming $\sum_i \lfloor f_i / t \rfloor$ upload slots among collected $\sum_i f_i$ slots, there remain internal reserves of amount $\sum_i (f_i - \lfloor f_i / t \rfloor)$ in the system, which are equally redistributed over all participants (independent of the magnitude of contribution) as a demogrant. With such a redistribution mechanism, the quality of services received by poor peers increases, which relaxes the gap of the quality of services caused in the auction-based schemes. An apparent drawback of the Chu’s taxation scheme is the lack of incentives for rich peers, because they will not contribute more than $f_i = t \times r_i$ slots, even if they have more resources.

4. Proposed Scheme

4.1 Overview

Most of existing incentive schemes for P2P live streaming systems are designed so that a heavy contributor can enjoy high quality services. In other words, the quality of received services increases as the amount of contribution increases. However, a naive application of such a survival of the fittest approach discourages peers with few resources to participate in the system since they always miss high quality services.

To overcome such an issue, in the proposed scheme, we combine the notion of minimum guaranteed services with a point-based incentive scheme. More concretely, the basic idea of the proposed scheme is: 1) to differentiate the quality of service according to the amount of contribution using a point-based scheme, and at the same time, 2) to guarantee the amount of minimum service for every peer by allowing peers to participate in at least $R_d$ trees (the role of parameter $R_d$ is similar to the demogrant used in the Chu’s taxation scheme, but it is calculated in a different manner). This idea is inspired by the Weber-Fechner law which states that the just-noticeable difference between two stimuli is proportional to the magnitude of the stimuli, or an increment is judged relative to the previous amount. In our case, the judged quality of a live stream is proportional to the logarithm of the number of subscribed stripes concerned with the stream, which indicates that in order to increase the overall utility, we need to increase the number of stripes subscribed by “poor” peers enjoying low quality streams, by decreasing the number of stripes subscribed by “rich” peers enjoying high quality streams.

In the following, after clarifying the underlying P2P architecture in Section 4.2, we describe the details of the point-based bidding procedure in Section 4.3. We then describe the way of tuning the value of $R_d$ in Section 4.4. and describe the way of carrying over unused points in Section 4.5.

4.2 P2P Architecture

We consider a P2P system consisting of $N$ homogeneous peers, a content server and a point server. Peers are assigned a unique ID and can subscribe to a stripe by participating in a tree associated with the stripe. Each peer periodically earns $k$ reward points from the point server by forwarding stripes to $k$ children, and as will be described later, such earned points are used for the bidding for the upload slot of an internal peer in another tree. The interval of earning points is called the earning interval. Reward points unused in an earning interval are stored at the point server, to encourage poor peers to earn reward points for the future use. We use symbol $\sigma[i]$ to denote the reward points currently held by peer $i$. The way of managing stored reward points, which is mandatory to avoid the inflation of the points, will be described in Section 4.5.

For each tree, the content server keeps tracks of the ID of the root of the tree, and maintains the following two sets $\text{InP}$ and $\text{LP}$ representing the availability of the upload slots in the tree, which can be referred by all peers in the system:

- $\text{InP}$ is the set of internal peers to have an available upload slot; i.e., peers whose fan-out is not exhausted.
- $\text{LP}$ is the set of leaf peers which have already participated in more than $R_d$ trees, where $R_d$ is the minimum number of trees guaranteed by the system. Peers in $\text{LP}$ are ordered in a non-increasing order of $\sigma[i]$. 
4.3 Bidding Procedure

In the proposed scheme, each peer can participate in at most one tree as an internal peer, while it can participate in any number of trees as a leaf as long as no conflict occurs. Each peer $a$ which wishes to join a tree first refers to the InP of the tree. If it is not empty, it completes the join after becoming a child of a peer in the InP, and if it is empty, it conducts one of the following operations depending on the (expected) role in the tree:

1) If peer $a$ joins as an internal peer, then after identifying a leaf peer $b$ in the tree, it inserts itself between $b$ and its parent, i.e., it becomes the parent of $b$ and a child of the former parent of $b$.

2) If $a$ joins the tree as a leaf peer, then it refers to the LP of the tree, and if it is empty, $a$ gives up the join at this moment (it retries to join after waiting a certain time from 10 to 15 sec). Otherwise, after selecting a peer with the smallest $\sigma[i]$ from the LP, $a$ initiates an auction for the upload slot of the parent $b$ of the peer.

The auction proceeds as follows. For each peer $i$, let $r(i)$ denote the number of trees in which peer $i$ currently participates and $C(i)$ denote the set of children of $i$ in the tree we are currently considering. The procedure separately considers the case of $r(a) \geq R_d$ (Case 1) and the case of $r(a) < R_d$ (Case 2).

Case 1) In this case, in order to keep that all children of peer $b$ participate in at least $R_d$ trees, peer $a$ competes with peers in set $\{i \in C(b) : r(i) > R_d\}$ for the upload slots of $b$. Each player $i$ bids $\sigma[i]$ reward points, and after receiving all bids, $b$ selects peers which submitted the largest bids as the winners, and makes them as its new children. Each winner $i$ pays 5% of its reward points to the system and each looser tries to join a tree after passing a certain time.

Case 2) In this case, we modify the procedure for Case 1 so that peer $a$ is removed from the set of competitors. In other words, peer $a$ can always join the tree as a leaf peer without paying reward points unless the LP of the tree is not empty.

4.4 Minimum Guaranteed Trees

The above procedure is designed to guarantee that every peer participates in at least $R_d(\geq 1)$ trees. However, an appropriate value of $R_d$ would change according to the join and the leave of peers. In order to reflect such a dynamic change to $R_d$, in the proposed scheme, the value of $R_d$ is periodically updated by the content server in the following manner:

- The server identifies the set of internal peers with their fan-out and the set of leaf peers with their reward points for each tree. Such an identification is realized by collecting information from participants through tree edges. The reader should note that InP and LP concern with each tree can also be constructed using the collected information.

- Let $F$ be the estimated fan-out of the overall system which is obtained by summing the fan-out over all trees. Let $N$ be the estimated number of participants and $\Delta N$ be the number of peers which newly joined the system during the current update period.

- Using those values, the server updates $R_d$ according to the following rule: if $F > 1.5 \times (R_d + 1)N$ then increment $R_d$ by one and if $F \leq 1.25 \times R_d(N + \Delta N)$ then decrement $R_d$ by one.

After that, the content server broadcasts a message containing the new value of $R_d$ to all peers to update the variable locally held by each peer. The behavior of each peer after receiving the update is as follows: When $R_d$ increases, a peer participating in less than $R_d$ trees tries to join a new tree (to become the participant of at least $R_d$ trees) after waiting a certain time. When $R_d$ decreases, it does nothing until it becomes a looser of an auction.

4.5 Carry-Over of Earned Points

In the proposed scheme, each peer periodically earns reward points from the system by serving as an uploader and bids those points for an upload slot of the other peers in the same bidding period. If it wins, it pays 5% of the bidded points to the system, and it carries over the remaining points for the future use. However, if we allow an unlimited carry-over of the remaining points, it causes the inflation of reward points which enlarges the gap between rich peers and poor peers. To overcome such an issue, in the proposed scheme, we bound the amount of carry-over of each peer $i$ by the following value:

$$\theta_i \overset{\text{def}}{=} 10 \times \log_{1.1} p[i].$$

With this mechanism, even poor peers to have few upload slots could earn sufficient amount of points by repeating contributions, and simultaneously, it could avoid rich peers to be a sole winner for a long time.

5. Evaluation

5.1 Setup

We evaluate the performance of the proposed scheme by simulation. In the simulation, all peers are synchronized to the global clock, and within a step of the clock corresponding to one second, each peer completes any procedure.
including the join to a tree and the bidding for an upload slot. Parameters used in the simulation are summarized in Table 1. According to the observation shown in [7], we assume that the upload bandwidth (and fan-out) of each peer follows the distribution shown in Table 2. Peers arrive at the system according to a Poisson distribution with mean 1 [peer/sec], and upon arrival, each peer is assigned a longevity $t$ according to the following distribution [6]:

$$F(t) = 1 - 1.23 \times e^{-\left(\frac{t}{1379}\right)^{0.99}}.$$  

Each peer does not leave until longevity exhausts, and the behavior of the peer is not affected by the rest of its life.

Each peer tries to participate in all trees, and if it participates in a tree as an internal peer, it uses all of its fan-out (determined as in Table 2) for the other peers in the tree. Let $s_{i,j}$ denote the number of stripes subscribed by peer $i$ in the $j^{th}$ step. Then, the utility $u_i$ of peer $i$ during simulation of length $T$ is defined as

$$u_i \overset{\text{def}}{=} \log_8 \left( \sum_{j=1}^{T} \frac{s_{i,j}}{T} \right),$$

where the reason of taking logarithm is that we are assuming the Weber-Fechner law as was described in Section 4.1 and we use eight as the base of the logarithm to normalized it in the range of $[0, 1]$.

Each run of the simulation consists of five consecutively executed sessions of 3600 sec and we conducted 50 independent runs to take an average of them. Carry-over of reward points (shown in Section 4.5) takes place within each run and the assignment of upload bandwidth is fixed in each run.

### 5.2 Comparison with Other Schemes

At first, we compare the performance of the proposed scheme with previous schemes described in Section 3. A comparison with Sepidar is illustrated in Figure 3. The horizontal axis is the fan-out of peers and the vertical axis is the average utility over all peers to have a given fan-out. This result indicates that compared with Sepidar, the proposed scheme certainly increases the utility of poor peers and the amount of increase is about 30% when the number of upload slots is small. In addition, the average utility over all peers is 0.87 in the proposed scheme, which significantly increases the average utility of 0.75 in Sepidar.

Next, we compare the performance of the proposed scheme with the Chu’s taxation scheme. Figure 4 illustrates the result. In the taxation scheme, the utility of rich peers does not change after the fan-out reaches 14. In other words, under this scheme, there are no incentives for rich peers for the further contribution when the number of contributed slots exceeds 14. In contrast to that, in the proposed scheme, the utility of rich peers gradually increases as the fan-out increases, which indicates that it certainly gives incentives to rich peers for the contribution of their resources.

This figure also shows that the absolute value of the utility of rich peers is smaller than the utility of rich peers in the taxation scheme, which is because of the following reasons. In the proposed scheme, the number of trees in which a peer can participate is determined by the bidding of reward points held by the peer. Hence even if it has a large fan-out, it could not join a sufficient number of trees if it has less reward points than the other peers. On the other hand, under

<table>
<thead>
<tr>
<th>Table 2: Distribution of upload bandwidth.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Upload bandwidth [KB/sec]</td>
</tr>
<tr>
<td>---------------------------</td>
</tr>
<tr>
<td>50-599</td>
</tr>
<tr>
<td>400-599</td>
</tr>
<tr>
<td>600-799</td>
</tr>
<tr>
<td>800-999</td>
</tr>
<tr>
<td>1000-1199</td>
</tr>
</tbody>
</table>

Fig. 3: Comparison with Sepidar.

Fig. 4: Comparison with the Chu’s taxation scheme.
the Chu’s scheme, the participation in a tree is controlled by the content server, and each peer contributing $f_i$ upload slots participates in $r_i(=\lfloor f_i/t \rfloor)$ trees with high probability.

### 5.3 Impact of the Number of Participants

Next, we evaluate the impact of the number of participants to the performance of the proposed scheme. In the simulation, we vary the number of participants from 500 to 3000 by keeping the distribution of peers for each fan-out as shown in Table 2. The result is illustrated in Figure 5. The vertical axis of the figure is normalized by the utility in the case of 2000 participants. This figure indicates that for any fan-out, the utility of peers increases as the number of participants increases, and the amount of increase is large for the peers to have small fan-out. The increase of the utility is apparently because of the increase of the number of available upload slots caused by the increase of the number of participants. In addition, a large increase for the peers with small fan-out is due to the low utility compared with the peers with large fan-out as was shown in Figure 3.

### 5.4 Impact of Two Intervals to the Performance

In this subsection, we evaluate the impact of two intervals used in the proposed scheme to the performance, i.e., earning interval described in Section 4.2 and the update interval of parameter $R_d$ described in Section 4.4.

Figure 6 shows the impact of the update interval to the utility (recall that in previous subsections, this interval was fixed to 30 sec as shown in Table 1). Each curve in the figure corresponds to an update interval, which is ranged from 30 sec to 1200 sec, and the vertical axis is the utility normalized by the result for 30 sec, i.e., “value 100” corresponds to the utility at 30 sec. As the interval increases, the utility of peers with small fan-out rapidly decreases, while that of peers with moderate fan-out slightly increases. Such a badness for poor peers is due to the way of updating $R_d$. More specifically, even if there are enough upload slots due to the arrival of new peers, the scheme increments $R_d$ one-by-one in each interval, which significantly loses the benefit of the minimum guarantee in the proposed scheme.

Finally, we evaluate the impact of the earning interval to the performance. The result is shown in Figure 7. The vertical axis of the figure is the utility normalized by the result for 10 sec, as before. The increase of the earning interval decreases the utility of most of the peers to have fan-out of more than four, while it increases the utility of peers to have small fan-out of less than four. This phenomenon could be explained as follows. A long earning interval reduces the chance of earning reward points for all peers. In addition, each winner should pay 5% of the reward points and the points which can be carried over to the next earning interval.
is bounded by a logarithm of the current point (see Section 4.5). Thus by increasing the earning interval, the competitive power of a peer reduces if it has large fan-out and increases if it has small fan-out.

6. Concluding Remarks

This paper proposes an incentive scheme for P2P live streaming system which is aware of the upload bandwidth of each participant. The proposed scheme is a combination of an auction-based incentive scheme and a taxation scheme based on the notion of minimum guaranteed services. The result of simulation indicates that it could increase the utility of poor peers by 30% compared with a conventional auction-based scheme and could improve the shape of utility function of rich peers so that it gradually increases as the amount of contribution increases.

An important future work is to refine the scheme by considering the cost, such as the maintenance cost of data structures and the cost for securely exchanging reward points. Another issue is to refine the model of simulation so that it reflects the dynamic change of the environment.

Acknowledgements

This paper was supported by the Grant-in-Aid for Scientific Research No.24500082 of the JSPS.

References


