A Payment-Based Incentive and Service Differentiation Scheme for Peer-to-Peer Streaming Broadcast

Guang Tan, *Member*, *IEEE*, and Stephen A. Jarvis, *Member*, *IEEE*

Abstract—We propose a novel payment-based incentive scheme for peer-to-peer (P2P) live media streaming. Using this approach, peers earn *points* by forwarding data to others. The data streaming is divided into fixed-length *periods*; during each of these periods, peers compete with each other for good parents (data suppliers) for the next period in a *first-price-auction*-like procedure using their points. We design a distributed algorithm to regulate peer competitions and consider various individual strategies for parent selection from a *game-theoretic* perspective. We then discuss possible strategies that can be used to maximize a peer's expected media quality by planning different bids for its substreams. Finally, in order to encourage off-session users to remain online and continue contributing to the network, we develop an optimal data forwarding strategy that allows peers to accumulate points that can be used in future services. Simulation results show that the proposed methods effectively differentiate the media qualities received by peers making different contributions (which originate from, for example, different forwarding bandwidths or servicing times) and at the same time maintain high overall system performance.

Index Terms—Peer-to-peer, media streaming, incentive, service differentiation, payment.

1 Introduction

The high scalability of peer-to-peer (P2P) systems relies on voluntary resource contributions by individual peers. However, the prevalent *free-riding* phenomenon observed on today's Internet imposes a practical restriction on the performance a P2P system. This problem has recently received a great deal of attention from researchers. Among the various applications, high-bandwidth live media streaming presents unique challenges that differ from those of applications such as P2P file sharing. In P2P streaming, the bandwidth becomes the bottleneck resource, and the peers' quality of service (QoS) depends on the available bandwidth of the overlay network.

Chu and Zhang [5] first consider altruism as a key element of P2P streaming broadcast. They show that the level of altruism has an important impact on the overlay; even a small degree of altruism brings significant benefits to the overall system performance. In [4], the same authors propose a taxation model, in which resource-rich peers are required to contribute more bandwidth to the system and subsidize the resource-poor peers. The social welfare (that is, the aggregate utility of all peers) is hence improved through the redistribution of wealth (that is, individual benefits in terms of the received media rate). This model assumes that a central entity (the content source) has the authority to enforce the taxation. In essence, this mechanism directly relates contribution and benefit in a deterministic and somewhat rigid manner.

Rather than enforcing compulsory contribution from peers, a score-based incentive mechanism [11] provides an indirect mapping between contribution and benefit. In this mechanism, the contribution level of a user is represented by a score, which is used to determine the rank of the user among all those in the system. Peer selection depends on the rank ordering of the requesters and candidate suppliers. For example, a peer may be allowed to select parents of equal or lower ranks. As a result, high-score peers are offered more flexibility in choosing desired data suppliers, whereas low-score peers have limited options in parent selection and hence receive low-quality streaming. In a payment-based [28] system, the peer network is treated as a market, in which each overlay node plays the dual roles of a service consumer and provider. Consumers try to buy the best possible service from service providers at the minimum price, whereas the providers strategically decide their respective prices in a *pricing game*, in order to maximize their economic revenues in the long run. To address the complexity of price setting, the authors use a *reinforcement learning* technique to solve for optimal strategies. This study focuses on the problem of bandwidth allocation and does not consider factors like network latency and packet loss rate, which are also critical to streaming quality.

In this paper, we propose a new incentive mechanism for P2P streaming. Our scheme uses an internal currency called *points* to represent a peer's contribution level, which is implicitly converted to the ability to compete for good parents. Although this paradigm is similar to that of the two approaches mentioned above, our design exhibits

- *G. Tan is with INRIA-Rennes, Campus Universitaire de Beaulieu 35042, Rennes Cedex, France. E-mail: gtan@irisa.fr.*
- *S.A. Jarvis is with the Department of Computer Science, University of Warwick, Coventry, CV4 7AL, UK. E-mail: saj@dcs.warwick.ac.uk.*

Manuscript received 24 Nov. 2006; revised 28 Apr. 2007; accepted 6 Sept. 2007; published online 24 Sept. 2007.
Recommended for acceptance by C. Shahabi.
For information on obtaining reprints of this article, please send e-mail to: tpds@computer.org, and reference IEEECS Log Number TPDS-0381-1106.
Digital Object Identifier no. 10.1109/TPDS.2007.70778.

several salient features that differentiate it from previous schemes:

- The overlay network is modeled as a market, in which peers earn points by selling data transmission service to others, and all trades are carried out through a *first-price-auction*-like procedure—peers bid for resources on desired parents, and the highest bidders win the service, thereby avoiding the complexity of price setting.
- Recognizing that unregulated competitions can result in very inefficient market trading, we design a distributed algorithm to facilitate parent finding. Specifically, we consider various strategies for peers to select targets for bidding from a *game-theoretic* perspective.
- Given a certain number of points, a peer has to decide how to spend them so as to maximize its expected media quality. We model the allocation of points to different substream bids as a combinatorial optimization problem in the context of a large-population and incomplete-information game and discuss possible approaches to addressing it.
- In order to encourage off-session users to continue to make contributions to the network, we design an optimal point accumulation strategy for peers to accumulate wealth, which can be used to improve its competitive power in future media services.

Simulations are conducted to study the performance of the proposed mechanism. The results show that the proposed methods effectively differentiate the QoS received by peers making different contributions and at the same time maintains a high overall system performance.

The remainder of the paper is organized as follows: Section 2 documents the related work. Section 3 presents an overview of the proposed mechanism. Section 4 presents the algorithm of reorganizing the overlay and analyzes strategies for peers to choose their parents. Section 5 discusses possible strategies that can be used to maximize a peer's expected media quality. Section 6 considers how to encourage off-session peers to make contributions through a point-awarding approach. Section 7 evaluates the performance of the proposed schemes through simulations, and Section 8 concludes the paper.

2 Other Related Work

There is a large body of literature on incentive mechanisms for general P2P systems (for example, [7], [12], and [9]). Here, we briefly review the techniques that are more relevant to incentive and differentiated service (DiffServ) schemes for P2P streaming broadcast.

To differentiate users' service quality, the system needs a means to quantify a peer's contribution/participation level in the network. A variety of ways have been proposed to do this. One of the commonly used methods is introducing some kind of currency. The Lightweight Currency Paradigm (LCP) [25] allows users to trade any resource with their own currencies; any entity can introduce their own currency as long as it is acceptable to others in the system. KARMA [27] uses a single type of currency and a set of "bank nodes" to facilitate secure trading. A peer's wealth is increased as resources are contributed and decreased as they are consumed. In view of the possible heavy load imposed on the central bank nodes, PPay [30] uses a *floating self-managed currency* to greatly reduce bank node involvement and thus improve trade efficiency. In the work of Ma et al. [13], peers maintain contribution values and trade with each other using *service receipts*. The contribution values of peers are maintained by a set of nodes called the *auditing authority*, which plays a similar role to the banking system assumed in our framework.

A second approach to differentiating peers' contribution levels is using reputations. With such a mechanism [10], peers earn their reputation by sharing, and highly reputed peers are more likely to obtain better service than peers with a low reputation. Finally, a score-based P2P system [32] scores users in order to differentiate peers of different contributions.

In recent years, game theory has been extensively used to analyze Internet economics and guide system design in DiffServ engineering. Buragohain et al. [2] analyze the optimal strategies of individual peers and the possible Nash Equilibria that can be obtained under different situations in the context of file sharing. Considering the dynamics in a real system, where the supply-demand relationship keeps changing, Wang and Li [29] model the P2P system as a *Cournot Oligopoly* game with dynamic payoff functions and propose a control-theoretic solution to the problem. Both sets of research are focused on economic analysis.

Auctions have been previously used in DiffServ. In [19], Semret et al. propose a decentralized auction-based paradigm for the pricing of the edge-allocated bandwidth in a DiffServ network. Using a game-theoretic model, they explore the feasibility of auctioning capacity in real time on the "demand side" and provisioning stable and consistent service-level agreements across multiple networks on the "supply side." Based on an abstract model for general DiffServ requirements, the work is again focused on economic analysis rather than the design of specific systems. A real system that has implemented auction-based micro-economic resource allocation is Mirage [6], [16]. Mirage is a sensor network testbed management system that employs a *repeated combinatorial auction* to allocate resources. Users send requests containing information about the bidding price, sensor set, duration, and radio frequency to the system, and the system allocates resources in a way that the aggregate utility is maximized. Although used for totally different purposes, Mirage and our system share some interesting features. For instance, in both systems, the auction is first price and is performed periodically; there exists a bank system managing users' credits, and schemes to mitigate the effects of strategic user behaviors are considered.

3 Design Overview

In the proposed mechanism, it is assumed that the system consists of a single source, or content publisher, that delivers data to the peer network. We also assume that a lightweight secure payment mechanism among peers is in place, which has been well studied as a building block of

TABLE 1
Notation and Variable Definitions

Notation	Definition
N	total number of peers in the network
S	number of substreams
T_m	the mth period
L	length of a period
A	bonus for forwarding a substream to a child for a period
C_i	total income (earned points) of peer i in some period
W_i	wealth (accumulated points) of peer i
I_i	number of in slots of peer i
O_i	number of out slots of peer i
l_{ij}	service latency of substream j of peer i
d_{ij}	data loss rate of substream j of peer i
b_{ij}	peer i's bid price for substream j
u_{ij}	utility of substream j of peer i
U_i	total utility of peer i

P2P economics (for example, [13] and [30]). For example, a set of bank servers can be used to manage the user accounts and payment process. There exists some basic bootstrap service that enables a new peer to identify a set of candidate parents. Finally, a lightweight network-time protocol is used to provide approximate overlay time synchronization. For example, NTP is a mature protocol that provides a scalable Internet-scale global time service with accuracies of 50 ms [15].

The data stream is divided into $S \geq 1$ independent and equally important substreams, or stripes, each with a unit bandwidth. A peer i has I_i incoming bandwidth slots (or "in slots" for short) and O_i outgoing bandwidth slots (or "out slots"), with each slot representing the bandwidth capacity of a substream. Since in P2P streaming, the bottleneck resource is the outgoing bandwidth capacity, we focus on the efficient utilization of this type of resource and assume that a peer's incoming bandwidth is always enough to support all substreams. In addition, we do not consider congestion in the core network, as congestion happens mostly at the access links on the Internet. Therefore, a peer can forward traffic to others as long as it has spare outgoing bandwidth. For ease of reference, a list of notation is given in Table 1.

3.1 Market Model

In the dimension of time, the data stream is also divided into fixed-length time periods $T_0, T_1, \ldots$, which can be as long as several minutes (for example, 4 minutes). Each peer possesses a certain number of points (that is, its wealth). Before the start of period T_m, a peer i needs to pay a price P_{ij}^m of points to its parent peer j to buy the data transmission service during T_m; the parent peer j, besides the P_{ij}^m points earned from its child, also receives a *bonus*, a small constant reward of A (for example, 2) points, from the payment system. The purpose of the bonus is to stimulate peers to serve newly arriving peers with 0 points to offer. Fig. 1 gives an example scenario of the market trading. In this example,

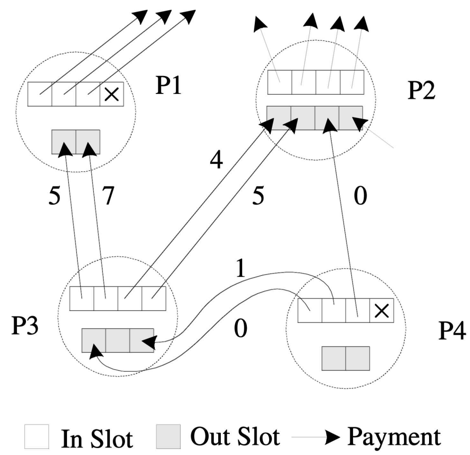


Fig. 1. Illustration of payment between peers.

Peer 3 requests for the transmission of substream 1 from Peer 1 during some period at a price of 5, and Peer 1 earns $5 + A$ points in the same period.

Given a certain number of points, a peer can spend it in many ways. Since the streaming proceeds in periods, a natural earning and consuming scheme is to pay all points earned in T_m for the data transmission in T_{m+1}. Although more economical and sophisticated methods may be devised, this scheme has the advantage of being simple and easy to implement. Hereinafter, we will use this scheme unless we specially target point accumulation rather than QoS optimization.

The start times of periods $T_0, T_1, T_2, \ldots$ actually define the preemption points for peers with different wealth. During each period, peers compete with each other for good parents (for example, those near the source) for the next period in a *first-price-auction*-like procedure: peers submit sealed bids simultaneously to their desired parents, and parents always choose the highest bidders as their children (ties are broken randomly). As a result, wealthy peers are able to choose their desired parents, whereas poor peers are given relatively limited, if any, options in selecting parents. When the whole overlay is short of bandwidth resource, some peers may not be able to find parents for all the substreams, thus receiving a reduced media bit rate. This mechanism stimulates peers to earn points as much as possible, the capability of which in turn depends on their forwarding bandwidths or servicing times contributed to the network. Although large wealth gaps may exist between peers, the probability of resource-poor peers suffering starvation is no higher than that in a nonincentive network, since under the stimulation of the bonus, peers with spare out slots will be more likely to serve peers than they are in a nonincentive network, even though the marginal utility of wealth is decreasing. Note that we do not claim to completely solve the problem of starvation, since this ultimately depends on the aggregate amount of physical resource available in the network; our aim is to achieve a lower probability of starvation by encouraging peers to contribute more usable resource.

3.2 Utility Function

In a free market, the goal of every peer is to maximize its own benefit in terms of media quality in every period. The media quality may be represented by using a utility

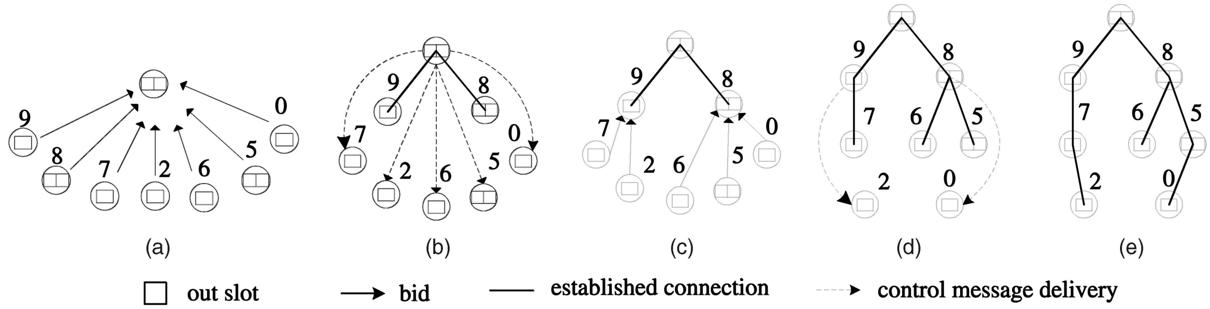


Fig. 2. An example of distributed virtual overlay construction. The numbers represent the bid prices.

function. Let integer $l_{ij} \geq 1$ denote the service latency (the sum of network latencies of all overlay hops from the source to the peer) in milliseconds of substream j of peer i and $d_{ij} \in [0, 1]$ denote the accumulative data loss rate of substream j received by peer i. Then, a simple expression of the utility of a single substream j could be

$$u_{ij} = \frac{\ln(1 + \text{data delivery rate})}{l_{ij}} = \frac{\ln(1 + 1 - d_{ij})}{l_{ij}}. \quad (1)$$

Clearly, the utility increases as the data loss rate or latency decreases. In particular, u_{ij} tends to 0 when d_{ij} tends to 1 or l_{ij} tends to infinity. The concave function $\ln(\cdot)$ captures the diminishing returns of a decreased data loss rate. In other words, when there is serious data loss (that is, d_{ij} is high), the same improvement in data delivery reliability brings more noticeable improvement to the user-perceived media quality than when the streaming is already reliable (that is, d_{ij} is low).

In order to examine the different impacts of the data loss rate and latency on the utility, we add two weight parameters, α and β, and a normalization factor $\ln 2$ to (1) as follows:

$$u_{ij} = \frac{\ln\left\{1 + \max[0, 1 - \alpha \cdot d_{ij}]\right\}}{\ln 2 \cdot (l_{ij})^{\beta}}, \quad (2)$$

where $\alpha > 0$ and $0 < \beta \leq 1$. Parameter α reflects the decreasing speed of utility as the loss rate increases, and β controls the impact of service latency. For applications without user interactions, service latency is less important, and β can be small. In the evaluation, we will use the definition of (2) and assume that all peers have the same α and β. Finally, the collective utility of a peer's substreams can be characterized as follows:

$$U_i = \ln\left(1 + \sum_{j=1}^{S} u_{ij}\right). \quad (3)$$

As in [4], the concave function $\ln(\cdot)$ represents the diminishing benefit for the user-perceived media quality as the overall bit rate increases.

It is worth noting that the above definitions are not the only way of relating the metrics u_{ij}, d_{ij}, and l_{ij}; many other definitions are possible as long as they correctly capture the increasing/decreasing relationship between these metrics.

4 Virtual Overlay Construction

During each period, peers need to find their next-period parents. These peers and the planned parent-child connections thus form a *virtual overlay*, which must be constructed before the next period starts; otherwise, peers have to find parents randomly and receive only random QoS. We design a distributed algorithm to generate the virtual overlay. For simplicity of discussion, we first assume that there is only one substream; the multiple-substream case is then extended on this basis.

4.1 The Distributed Algorithm

The distributed algorithm allows individual peers to contact their candidate parents themselves. This allows them to identify the best parents through real network probing and measurements. A naive way for a peer to find its optimal parent is to bid for the best peer (for example, the source node) first and then try for the second best, the third best, and so on, until it wins one parent. In this way, each peer needs to compete for $O(N)$ times in order to find a parent. In a large network, this may result in excessive competitions and, hence, very inefficient trading. Therefore, the competitions must be regulated.

To do this, our scheme uses the following recursive searching process. When a peer wants to find an ideal parent, it takes part in a competition for that parent. If it wins, it becomes a child of that parent; otherwise, it obtains a list of the winner peers, from which it tries to find a new best parent. It again takes part in the competition for that new parent and continues this process until it wins a parent or has no parents to choose. In the latter case, it tries to find a parent in a best effort manner (with no guarantee of QoS of course). An example of this process is illustrated in Fig. 2.

In Fig. 2a, there are seven peers competing for two spare slots of the top peer. The two highest-bid peers, 9 and 8 (a peer is identified using its unique bid value), are adopted by the top peer, as shown in Fig. 2b. The rejected peers, 7, 2, 6, 5, and 0, then receive a peer list that contains the addresses of 9 and 8. Using this list the rejected peers begin to look for new candidate parents with some criteria, which will be discussed in the next section. As a possible scenario, as shown in Fig. 2c, peers 7 and 2 choose 9, whereas peers 6, 5, and 0 choose 8 as their new targets and begin a new round of competition. As a result, peers 2 and 0 are ruled out, and peers 7, 6, and 5 find their positions in the tree (Fig. 2d). Using the same method of finding new parents, peers 2 and 0 finally find 7 and 5 as their parents, respectively (Fig. 2e), and this finishes the virtual overlay construction.

In each period, the parent finding procedure starts from the source node as this provides the most opportunities for peers to find their desired parents. The distributed algorithm divides a period into multiple *bidding rounds*, during which peers submit bids to their target parents and collect the responses. The length of a bidding round is set to L (the length of a period) divided by the maximum possible number of tree levels, which is a fixed parameter and can be estimated according to the expected network population size and average node bandwidth. For example, if $L = 180$ seconds and the maximum number of tree levels is 30, then the round length is 6 seconds. Generally, the round length should be at least in the order of seconds so that the round-trip times of all peers can be accommodated and the global time errors become negligible. If a peer does not receive a response from a target parent within a bidding round, it tries to contact another target for the next round. Under exceptions such as the early departure of the determined parent before the start of the next period, a peer finds its parent in a best effort manner: it selects a parent from its near neighbors that has spare out slots and wishes to accept it as a child—no bidding is needed, and the parent receives the basic bonus as its income. The same happens if a peer enters the system in the middle of a period. Payments are made at the beginning of the next period. If a peer finds that its parent leaves in the middle of a period, it is entitled to get back all the points it paid for the current period. It reports the parent leaving early to the bank system, which returns the points to it and then notifies the parent of the "refund"; if the parent identifies that the child is lying, it simply stops sending media data to that child. To prevent a peer from consistently requesting for a refund at the end of each period, the refund can be allowed only for a certain time (for example, 10 seconds) before the end of a period.

4.2 Choosing New Parents for Bidding

As described in the preceding section, when a peer fails in the competition for a certain parent, it needs to choose a new target parent from the winners (which become the children of the original parent). We consider two strategies for this, namely, the Shortest Path (SP) strategy and the Balanced Tree (BT) strategy.

4.2.1 Shortest Path (SP) Strategy

With this strategy, a bidding peer contacts all the candidate parents to obtain their service delays and at the same time measures the delays between itself and those candidates. The bidding peer then selects a target parent from the candidates that makes its accumulative service latency to the source the smallest. Note that here we do not consider the loss rate, which is another factor in the utility function, because it is more difficult to obtain an accurate measurement of the loss rate in a short time [20].

The SP strategy may result in a tall tree since a large number of peers may compete for a single well-located peer, making the subtree under the target peer tall. On the other hand, some peers may not even attract enough peers to fill in their slots and become leaf nodes early. In an unbalanced tree, a peer's expected overlay path length (that is, the number of overlay hops from the source to itself) can be much larger than in a BT. A possible scenario is illustrated in Fig. 3a, in which six peers (each with two out slots) are competing for two out slots on a single

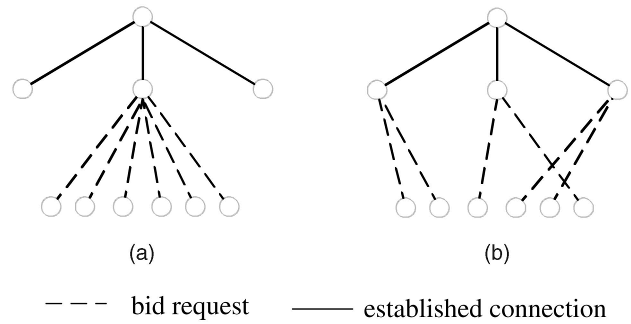


Fig. 3. Illustration of the different parent selection strategies. Each node has two out slots. (a) SP strategy. (b) BT strategy.

parent. As a result, the subtree under the parent will be at least two levels high, whereas the sibling peers cannot attract child peers and themselves become leaf nodes.

4.2.2 Balanced Tree (BT) Strategy

A peer with this strategy chooses a candidate parent probabilistically in an attempt to balance the tree. Given a set of candidates, the probability of one candidate parent being picked is proportional to its number of out slots. See Fig. 3b for an example.

The BT strategy helps to construct a short tree, which translates to a small average overlay path length for peers. This has important implications on the system's performance. Since in an overlay network, the transience of peers becomes the dominant factor that affects the streaming stability, a shortened overlay tree can effectively reduce the probability of streaming disruptions and, hence, the packet loss rate [17], [24], [21] and finally increase the average utility of peers (see (2)). On the other hand, due to the ignorance of locality information, the average service latency may be larger than it is when all peers use the SP strategy, thus impacting on the average QoS. As a result, how system performance benefits from the two strategies ultimately depends on the relative importance of the loss rate and service latency to the streaming quality—when the loss rate is weighted higher, the BT strategy benefits the system more; otherwise, the SP strategy is a better choice.

Although the BT strategy benefits the system as a whole under certain circumstances, it might be important to look at it from the viewpoint of individual peers. One interesting question is why would all rational peers be willing to choose the BT strategy?

To explore this, let us first consider a simplified problem as follows: Suppose that there are I peers competing for m slots on some parent peers, each peer has exactly d out slots, and both the latency and the loss rate between any pair of peers are constant. The question is, does there exist a decision agreement for all peers from which individuals would not make unilateral changes? Alternatively, in terms of game theory [14], does there exists a strategy profile that leads to a Nash Equilibrium for game $\Gamma_N = [I, \{\Delta(S_i)\}, \{u_i(\cdot)\}]$, where I is the player set, $\Delta(S_i)$ denotes player i's mixed strategy over the pure strategy set $S_i = \{1, 2, \ldots, m\}$, and $u_i(\cdot)$ is player i's payoff function as defined in (2)? The following theorem shows that if each peer chooses the out slots with equal probability, then a Nash Equilibrium can be achieved:

Theorem 1. *Define the mixed strategy for player* i $(i = 1, 2, \ldots, I)$, σ_i, *that assigns equal probability* $1/m$ *to all pure strategies in* S_i. *Then, the mixed strategy profile* $\sigma = (\sigma_1, \sigma_2, \ldots, \sigma_I)$ *constitutes a Nash Equilibrium of game* Γ_N.

Proof. Refer to Appendix A. □

Now, we return to the original problem. It can be seen that the BT strategy has the same effect of letting a peer choose any out slot from the candidate parents with equal probability. If we assume that the peer is completely unaware of the latency and loss rate between any peer pairs, including the parent and its rivals, then it can simply assume that each rival has d out slots on average, and the latency and packet loss between any peer pair are both constants, which can be interpreted as their expected values. Theorem 1 therefore applies, and the BT strategy leads to a game equilibrium. To make this assumption hold, we make a small change to the basic bidding procedure: when the peers who failed in the competition for some candidate parent obtain a list of the winners, the identifiers of the winners are only given in the form of some opaque IDs instead of their IP addresses; a peer chooses one of the IDs as its next bidding target and then asks the candidate parent for the target's IP address. This way, the bidding peer will not be able to calculate its own service delay, thus having no motivation to deviate from the strategy that leads to a game equilibrium.

4.3 Security Issues

Under the DiffServ rules, the resource-rich peers generally are more competitive than others and thus are likely to remain high in the tree. This leaves the possibility that adversaries equipped with plenty of points occupy all high-level tree positions and then issue a denial-of-service attack to block critical broadcast services (until their points are exhausted). Indeed, this issue is likely to be common to all incentive schemes for tree-based multicast systems where proximity to the root is a major reward to the contributing peers. A simple solution to this problem is to reserve a certain proportion (for example, 20 percent) of root out slots for serving peers in a nonincentive manner (for example, First Come, First Served). In so doing, a partial non-incentive-based tree will be available for normal data dissemination in the presence of malicious behaviors, although the streaming performance may be degraded. This actually reflects a trade-off between incentive strength and system security: the more root slots open for competition, the more effective the incentive rule is, but the more likely the system is to be exposed to a denial-of-service attack.

Another security issue of the incentive scheme comes from the use of the bonus. If a peer A pretends to provide a service to some conspirator peer B in every period regardless of the bidding rule, then peer A can continuously receive the bonus for the fake service. This problem can be mitigated by enforcing the bidding rule: the highest price bidders are selected first (with ties broken randomly). This rule ensures that the conspirator peer B cannot guarantee a constant connection with peer A for exploiting the bonus at a low price; if peer B is to pay a high price to peer A in order to maintain such a connection, then peer B is effectively transferring its wealth to peer A. We will also describe in the following a method to reduce the possibility of two peers exploiting the bonus by frequently paying a high price to each other.

The bidding rule is enforced in the following way: Every peer maintains a history of its failed bids and periodically sends information including the bidding time, bidding price, and bidding target to the source node via the reverse route of the streaming data; the source node also periodically retrieves the payment records corresponding to the successful bids from the banking system (which we assume is able to provide such data). Combining these data, the source node can run a background program that periodically checks whether peers do not adhere to the bidding rule in selecting their children or whether points are paid in an abnormal way (for example, two peers paying each other in a single period or frequently exchanging points across successive periods). If the source node detects that some peers are obviously, or with a high probability, violating the bidding rule or cheating, then the source node asks the banking system to impose a punishment on those peers by deducting a certain number of points from the their deposits and to stop awarding bonuses to them in the future. Considering that inspecting every peer's behavior can be a nontrivial task, the source node can sample a certain fraction of peers at a time; the sampling process can also be biased toward the wealthy or long-lived peers.

4.4 Extension to Multiple Substreams

So far we have assumed the case of a single substream, in which a peer can use all of its out slots for that substream. When there is more than one substream, peers need to determine how to assign the out slots to each. To simplify the problem and considering that all substreams are symmetric, we choose to allocate the out slots evenly to all substreams. This proves to be a viable approach, as will be demonstrated by the experimental results.

5 In-Session Utility Maximization

During each period T_m, a peer i needs to plan for the bids for each substream using the C points earned before the start of T_m, with the objective of maximizing its expected utility during T_{m+1}. (The index i is omitted for brevity of notation.) We consider three possible strategies.

5.1 The Even Allocation Strategy

The first and simplest strategy is to allocate points to all bids evenly. This strategy is static and easy to implement. Unfortunately, it does not lead to a Nash Equilibrium: consider a simple example of two peers, 1 and 2, with C_1 and C_2 $(C_2 < C_1)$ points, respectively, competing for two slots s_1 and s_2 on some parent peer. It can be seen that the even allocation makes peer 2 fail in every competition (because $\frac{C_2}{2} < \frac{C_1}{2}$), hence becoming a *dominated strategy* for peer 2. Peer 2 may therefore deviate from this strategy.

5.2 The Random Allocation Strategy

The second strategy is to randomly allocate points to the bids. In the above example (here, we assume more generally that $C_2 \leq C_1$), if peer 2 chooses one of its two bids uniformly at random between $[0, C_2]$ and lets the other bid take the rest of the points, then its winning probability would be higher. In particular, if peer 1 chooses its bidding prices in this random way as well, then the expected number of bids won by peer 1 and peer 2 are $\frac{2C_1-C_2+1}{C_1+1}$ and $\frac{C_2+1}{C_1+1}$, respectively, and both peers would not be better off deviating from this randomized strategy, thereby achieving a Nash Equilibrium. Formally, we define a two-player game $\Gamma = [I, \{\Delta(S_i)\}, \{u_i(\cdot)\}]$, where $I = \{1, 2\}$ is the player set, $\Delta(S_i)$ denotes player i's mixed strategy over the pure strategy set $S_i = \{(b_i^1, C_i - b_i^1) : b_i^1 = 0, 1, \ldots, C_i\}$, and $u_i(\cdot)$ is player i's payoff function defined as the expected number of bids won. Then, the following theorem holds:

Theorem 2. *Define the mixed strategy for player i ($i = 1, 2$), σ_i, that assigns equal probability $\frac{1}{C_i+1}$ to all pure strategies $s_i \in S_i$, then the mixed strategy profile $\sigma = (\sigma_1, \sigma_2)$ constitutes a Nash Equilibrium, at which point $u_1(\sigma) = \frac{2C_1-C_2+1}{C_1+1}$ and $u_2(\sigma) = \frac{C_2+1}{C_1+1}$.*

Proof. Refer to Appendix B. □

Generally, for an m-slot, n-player game in which the ith player has C_i points, we can prove that the strategy profile that every peer assigns uniformly random prices (under the constraint of C_i) to the bids still leads to a Nash Equilibrium. The proof is similar to that of Theorem 1, and we omit it in the paper.

5.3 The Estimate-Based Allocation Strategy

Although it has the advantages of being simple and leading to a Nash Equilibrium, the random allocation strategy does not explicitly consider the optimization of utility in terms of (2). We want to see how an individual's utility and system performance are affected if peers take a strategy that considers both service latency l and loss rate d. In the following *estimate-based allocation strategy*, a peer estimates its tree positions and then l and d in the next period and allocates its points in such a way that the expected utility in terms of (2) is maximized. Specifically, let the bids for substream $1, 2, \ldots, S$ be $b_1, b_2, \ldots, b_S$, respectively, and let functions $L_j(\cdot)$ and $D_j(\cdot)$ denote the mappings from bid b_j to the expected service latency and loss rate of substream j, respectively. Then, this problem can be formulated as follows:

$$\text{Maximize } U = \sum_{j=1}^{S} u_j \tag{4}$$

$$= \sum_{j=1}^{S} \frac{\ln\left[1 + \max(0, 1 - \alpha d_j)\right]}{\ln 2 \cdot (l_j)^{\beta}} \tag{5}$$

$$= \sum_{j=1}^{S} \frac{\ln\left[1 + \max(0, 1 - \alpha D_j(b_j))\right]}{\ln 2 \cdot L_j(b_j)^{\beta}} \tag{6}$$

$$\text{subject to } \sum_{j=1}^{S} b_j = C. \tag{7}$$

Unfortunately, both $L_j(\cdot)$ and $D_j(\cdot)$ are not known a priori since one peer has no preview of the overlay to be formed in the next period, which requires at least the knowledge of all other peers' bidding prices and a prediction of the overlay dynamics. We have to let each peer estimate $L_j(\cdot)$ and $D_j(\cdot)$ from past experience. Note that due to the very limited knowledge of individuals about the network, the following estimates may not be accurate; however, at this first step, our aim is only to investigate whether the effort in this direction will be paid off in peers' utilities.

First, a peer i maintains a mapping $\widehat{BH}_j$ from bid b_j to the average tree level number h_j for each substream. Due to the limited experience of individual peers (recall that a peer's lifetime usually lasts only tens of periods, and hence, the samples derived are very limited), it also exchanges this information with other trustworthy peers to obtain more samples (for example, those bids not made by itself but observed on others and the associated results). In order to keep a relatively accurate picture of the external environment, peers only use the information from the past few periods. Second, a peer maintains a mapping $\widehat{HL}_j$ from its tree level number h_j to the service latency l_j for each substream, and when needed, it estimates a latency for a given level number using linear regression. Third, a peer maintains a mapping $\widehat{HD}_j$ from its level number h_j to the loss rate d_j for each substream.

Given a bid b_j, peer i first uses $\widehat{BH}_j$ to estimate the level number it is expected to be at in the tree. The service latency l_j and the data loss rate d_j can then be estimated using $\widehat{HL}_j$ and $\widehat{HD}_j$, respectively. Using these estimates a utility can be computed. So far, we have established the mapping from a bid b_j to the expected utility u_j.

The next step is to solve the optimization problem defined by (6) and (7). We choose to use a genetic algorithm, which provides a general solving framework for combinatorial optimization problems. Being an iterative algorithm, it allows one to conveniently balance the execution time and solution quality and hence is suited to a time-constrained scenario.

In the above method, a peer makes decisions based on the assumption that all other peers would not change their bids and there is no change in the external environment. This is not the case in our network. However, under a highly dynamic environment and with unpredictable individual behavior, it is very difficult to mathematically characterize all the dynamics and then solve for optimal strategies, and therefore, the "myopic" strategy [1] might be a plausible choice; this is in spirit like playing a *fictitious play* [8]. In Section 7, we will examine whether this approach can bring improvement to peers' utilities.

6 Off-Session Point Accumulation

The purpose of off-session point accumulation is to encourage peers that are no longer in the media sessions to continue to make contributions to the network. In return for this, they earn points that can be used in later sessions to improve their media quality. To maximize the individual benefit, such a peer will seek to maximize its wealth instead of media quality. Here, a decision problem during each period is how to allocate points from its current wealth W_i

to the substream bids so that its expected income in terms of points during the next period is maximized. If we let the vectors $(b_1, b_2, \ldots, b_S)$ and $(o_1, o_2, \ldots, o_S)$ denote its bids and out slot allocations for all substreams, respectively, and define the function E_j as the mapping from b_j to the expected income from a single out slot of substream j (again, the index i is omitted), then the wealth maximization problem can be formulated as follows:

$$\text{Maximize} \sum_{j=1}^{S} E_j(b_j) \cdot o_j - \sum_{j=1}^{S} b_j \tag{8}$$

$$\text{subject to} \sum_{j=1}^{S} b_j < W \tag{9}$$

$$\text{and} \sum_{j=1}^{S} o_j < O. \tag{10}$$

By analyzing the market behavior and based on experimental experience, we can make several observations:

1. $E_j(b_1) \geq E_j(b_2)$ for any $b_1 \geq b_2$, which means that a higher bidding price always brings no less than the expected income. This is the case since a higher bidding price leads to a higher utility for a substream, which makes it more attractive to other peers.
2. $E_m(b) = E_n(b)$ for any substream m and n. This means that the same bidding price brings the same expected number of points for different substreams. This is reasonable, given that all substreams are symmetric.
3. A single peer's out slot allocation has negligible influence on the supply-demand relationship of any substream. In a market with a large population, $E_j(\cdot)$ would not change as a single peer adjusts its out slot allocation (which is invisible to others), so $E_j(\cdot) \cdot o_j$ still represents the total expected income from substream j no matter how peer i allocates its out slots.

These observations lead to the following conclusion:

Theorem 3. *The problem defined in expressions (8)-(10) is equivalent to the following problem:*

$$\text{Maximize } E_x(b_x) \cdot O - b_x \tag{11}$$

$$\text{subject to } b_x < W, \tag{12}$$

where x is any integer from $\{1, 2, \ldots, S\}$.

Proof. Refer to Appendix C. □

The above conclusion means that rather than bidding and allocating out slots for multiple substreams, we only need to choose a single substream x and allocate all out slots to it. With the simplification, the remaining problem becomes how to determine the bidding price for substream x so that the objective of (11) is achieved. Central to this problem is to determine $E_j(\cdot)$. Once again, we must resort to historic knowledge. A peer can maintain a mapping from b_j to the average income from an out slot of substream j using the record of the past few periods; at the same time, it exchanges this information with others to supplement its own experience. This way an estimated mapping $\hat{E}_j(\cdot)$ can be generated. Using this, the problem defined in Theorem 3 can be solved in $O(W)$ time.

When a peer enters a session in the future, it has many ways to spend the accumulated points. For example, it can evenly allocate them to an estimated number of periods. The increased points will help it compete for a better QoS than it can obtain without the accumulation process. From the system's perspective, the point accumulation mechanism also helps to increase the overall system resources. Since in this mode, a peer consumes only one substream (that is, it occupies one out slot from another peer) while contributing all of its out slots, the total number of spare bandwidth slots can be effectively increased as more and more off-session peers choose to stay online rather than quit the applications or shut down the hosts.

Note that even if some peers who accumulate a large number of points offline go online suddenly, they could barely impact on the stability of the system, because peers arriving in the middle of a period can only find parents in a best effort manner—the preemption only happens at the beginning of the next period, when all peers are ready for parent changes.

7 Performance Evaluation

To study the performance of different P2P streaming algorithms, we have developed an event-driven simulator based on a carefully configured simulation model. The GT-ITM transit-stub model [31] is used to generate an underlying network topology consisting of 2,592 nodes. Link delays between two transit nodes, transit nodes and stub nodes, and two stub nodes are chosen uniformly between [15, 25] ms, [5, 9] ms, and [2, 4] ms, respectively. A total of 1,800 nodes are randomly selected to be peers participating in the multicast tree. The server's location is fixed at a randomly chosen stub node. The packet loss rate between any two peers is uniformly drawn from [0, 0.06]. These underlying network settings are independent of the upper layer application logic, and we have found that varying these parameters (for example, the network size and the latency ranges) generated consistent results regarding the relative performance of different schemes, so we will only report results for this particular setting.

In all simulations, there are eight substreams. The total stream bandwidth is assumed to be 300 Kbps. The root node has 80 out slots (or 10 full streams). Other nodes' outgoing bandwidths follow a bounded Pareto distribution.[1] Let $\text{BP}(u, v, p)$ denote the bounded Pareto distribution with lower bound u, upper bound v, and scale p. Then, the default number of a peer's out slots is generated from BP(0.4, 15, 1.2) times 300. This generates the bandwidth setting as shown in Table 2, which is comparable to the statistics reported in [21]. The nodes' lifetimes follow a lognormal distribution [26], [22] with the μ (location parameter) and σ (shape parameter) set to 5.5 and 2.0, respectively, which are chosen according to the statistical

1. Previous studies [21], [18] have shown that the (access link) bandwidths of overlay nodes exhibit characteristics similar to that of heavy-tailed distributions, a typical example of which is the Pareto distribution. Considering the practical limits of possible bandwidth values, we use a bounded Pareto distribution to model the members' bandwidths.

TABLE 2
Default Bandwidth Setting Used in the Simulation

Bandwidth range	# Full stream / substream	%
$(0, 300)$ Kbps	0 / 0-7	70.55
$(300, 600)$ Kbps	1 / 8-15	17.45
$(600, 1200)$Kbps	2 / 16-23	7.75
> 1.2 Mbps	3-14 / 24-159	4.25
Total	-	100.0
Average	1.2 / 9.6	-

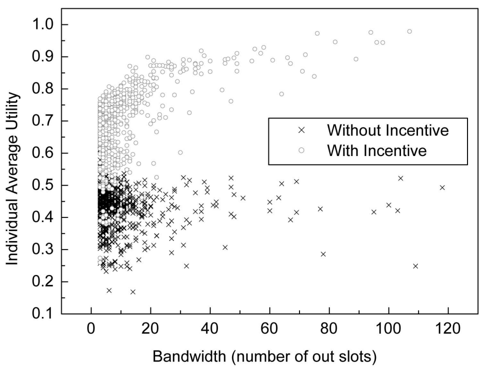


Fig. 4. Individual average utility versus bandwidth.

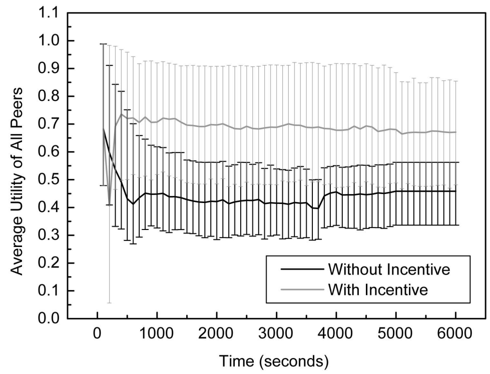


Fig. 5. Average utility of all peers as a function of time.

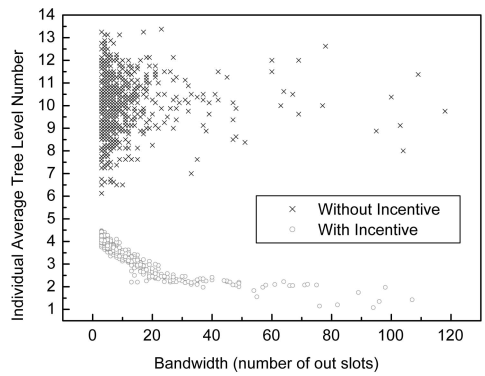


Fig. 6. Individual average tree level number versus bandwidth.

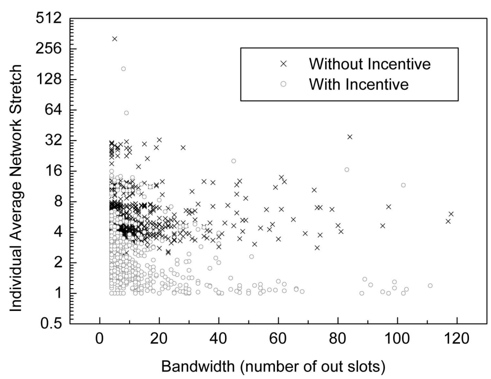


Fig. 7. Individual average network stretch versus bandwidth.

findings in [26]. According to *Little's Law*, the node arrival rate λ is determined from M divided by the mean lifetime, that is, 1,809 seconds. By default, the streaming period is 120 seconds, the α and β parameters in the utility evaluation function are 1 and 0.25, respectively, and the parent selection strategy is SP. Except in Section 7.5, each peer has 0 points when it initially joins the network. The bonus value (see Table 1) is set to 10.

We will compare our scheme against a nonincentive scheme, in which a peer chooses its parents in a best effort manner; that is, for each substream, it tries to find from the candidates the parent that makes the accumulative service delay the smallest; it will give up if no candidate parents with spare slots can be found. This method is similar to the one used in [17] or the minimum-depth algorithm used in [21]. The same bandwidth distribution is assumed in both incentive and nonincentive cases.

7.1 Effectiveness of the Incentive Mechanism

This experiment compares the individual utilities and system performance under situations with and without the proposed incentive mechanism. A metric called *individual average utility* is defined as a peer's utility averaged over all of its periods, which is then normalized over a maximum value obtained when the peer is directly connected to the source and with no packet loss. Fig. 4 plots the results against the peers' bandwidths after the network enters a steady state. Two observations can be made from this figure. First, when there is no incentive mechanism, the peers' utilities exhibit a random distribution, whereas with incentives, there is a clear correlation between the outgoing bandwidth and the utilities—the higher the bandwidth, the higher the utility. This indicates that our mechanism effectively differentiates the service quality of peers with different service capacities. A second observation is that after using an incentive, most peers have substantially higher utilities than they do without it. This can be further seen in Fig. 5, in which the average utility of all peers (with a 95 percent confidence interval) changing over a time interval of 6,000 seconds is plotted. It can be seen that the incentive mechanism provides considerable benefits to the overall system performance.

Figs. 6 and 7 can serve as an explanation for the above phenomena. In Fig. 6, the *individual average tree level number*, that is, the average tree level number of a peers' collective substreams averaged over all periods in its lifetime, is plotted against the outgoing bandwidth. It can be seen that with added incentives, the peers' average tree level numbers are far smaller than they are in a nonincentive network. This is because the high-bandwidth peers are offered high positions in the multicast tree, thus resulting in a wider and shorter tree. This generally means that peers have smaller average service latencies and packet loss rates. Fig. 7 shows the *individual average network stretch* against a peer's bandwidth. A peer's *network stretch* is the ratio of its

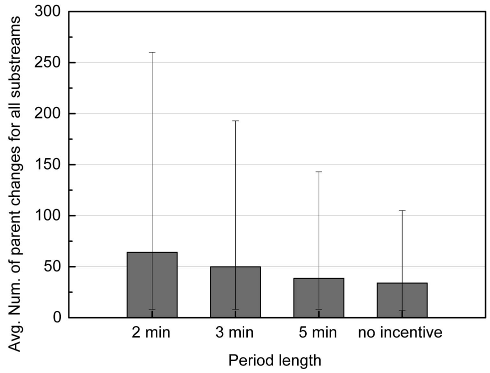


Fig. 8. Overlay maintenance cost in terms of the number of parent changes.

accumulative service latency to its latency from the root along a unicast path in the underlying network [3]. The individual average network stretch is then defined as the average network stretch of all substreams averaged over all periods. The results show that the incentive mechanism effectively reduces the network stretches of most peers, with the average value being reduced by a factor of 56 percent (from 6.32 to 2.78). Note that for both incentive and nonincentive cases, there are a few points with very high stretch values. This is because the corresponding peers are very close to the source server but are placed in a low level due to the randomness of the overlay construction (in the nonincentive case) or their limited contribution levels (in the incentive case).

7.2 Effects of Period Length

The incentive mechanism enforces fairness and improves system performance by offering peers many opportunities to actively switch parents. In this experiment, we examine how this mechanism affects the overlay maintenance cost, which is measured by the average number of substream parent reconnections of all peers, including the reconnections introduced by both the incentive mechanism and the departure of upstream parents. Fig. 8 plots the maintenance costs under different period lengths (the "no-incentive" case is equivalent to the case of an infinite period length). As expected, a longer period leads to fewer parent changes. When the period is 2 minutes long, the average number of parent changes for a single peer is less than 72, which translates to nine changes per substream. Observing that the

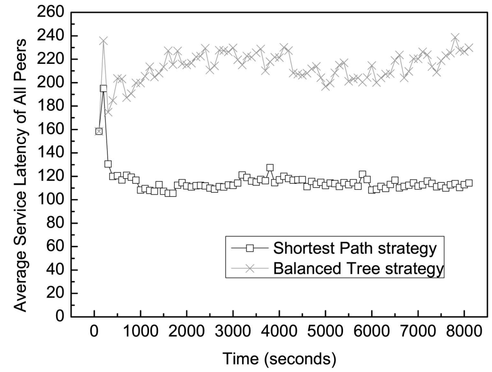


Fig. 10. Average service latency of all peers over time.

average lifetime is more than 1,800 seconds, the parent change frequency is actually less than once every 200 seconds, which is at an acceptable level. Another important observation is that with a period length of 5 minutes, the overlay maintenance cost is indeed very close to that of the nonincentive case (which has an average substream parent change frequency of once every 360 seconds). The reason behind this is that although the incentive mechanism requires peers to make extra parent changes in order to adjust the overlay, the shortened tree helps reduce the parent changes incurred by another major source—that of unexpected upstream parent departures. Given the heterogeneous node out-degrees, a shortened tree causes fewer parent changes for tree nodes because the average number of nodes' descendants in a short tree is smaller than in a tall tree; the same observation has also been made in [17], [24], and [21].

While a longer period reduces the individual overhead, from the system's viewpoint, a shorter period offers more chances for the overlay to be adjusted and hence means better systemwide performance. As shown in Fig. 9, a short period leads to higher and more stable system performance in terms of the average utility of all peers, especially when the bandwidth resource is not so rich.

7.3 Comparison of Parent Selection Strategies

In this experiment, we examine the effect of different parent selection strategies on tree characteristics and media quality received by peers. Figs. 10 and 11 compare the average service latencies of all peers and tree depths under the SP and BT strategies. As expected, the SP strategy results in

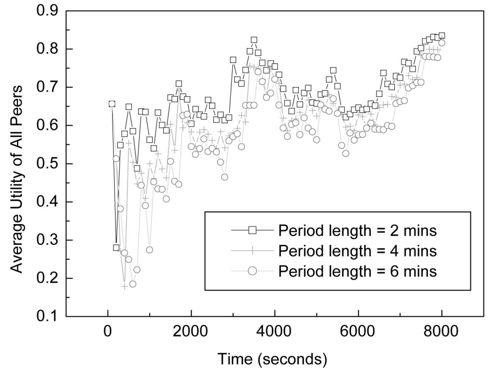


Fig. 9. Effect of period length on system performance.

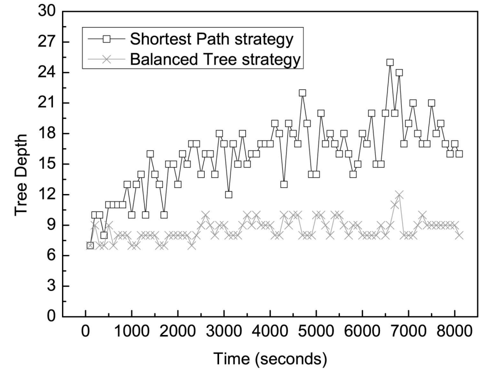


Fig. 11. Tree depth over time.

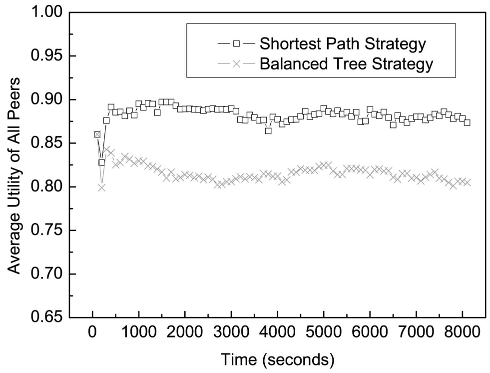


Fig. 12. Average utility of all peers over time. $\alpha = 1$, $\beta = 0.25$.

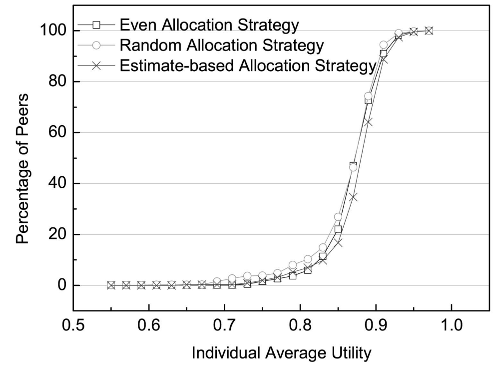


Fig. 14. CDF of individual utilities under the SP strategy.

a much smaller average service latency because peers have the freedom to choose near parents, although the tree under the SP strategy is much higher than that under the BT strategy (see Fig. 11). On the other hand, due to more overlay hops between the source and the peers, the average loss rate under the SP strategy can be much larger than that under the BT strategy. Therefore, how peers' utilities compare under these two strategies depends on the weights of the loss rate (α) and the latency (β) in the utility function. Figs. 12 and 13 show the average utility as a function of time under different settings of α and β. When α is 1 and β is 0.25 (see Fig. 12), the SP strategy is better than the BT strategy, whereas in Fig. 13, where α is changed to 2.5 and β to 0.05, the BT strategy becomes a better choice. Note that the short tree produced by the BT strategy has more implications on streaming quality than what is shown here; for a more thorough discussion of the relationship between tree shape and streaming quality, particularly with respect to reliability, the reader is referred to [23].

7.4 Comparison of Bidding Strategies

We compare the performance of three bidding strategies. For the genetic algorithm used in the estimate-based strategy, we set the population size to 50 and the number of generations to 30. The crossover and mutation probabilities are 0.2 and 0.02, respectively. These settings are not meant to be optimal for finding solutions; we only need the algorithm to produce some good solutions within a short time. On our modern desktop machine, the computation of one solution took only a few milliseconds.

Figs. 14 and 15 plot the cumulative percentages of peers whose average utilities are above certain values under the SP and BT strategies. The results show that the three bidding methods yield almost the same results. This is easy to understand for the random allocation scheme, because under this scheme, peers do not specially optimize the expected utility. The advantage of such a strategy is the equilibrium achieved, as discussed in Section 5. For the estimate-based strategy, the increased complexity actually brings no improvement over the two simple schemes. This is mainly because every peer is optimizing its expected utility based on the assumption that others do not change their decisions in the next period, which is obviously not the case in our framework. The simultaneous changing of point allocations by all the peers make the tree that is formed totally different from the trees envisioned by individuals, and consequently, the peers' calculated allocations do not help in the new configuration. In effect, the new allocations are no more than some random allocation input to the utility optimization problem. In game theory, an interesting problem of this kind of fictitious play [8] is whether the players' best response strategies could converge to equilibria. Unfortunately, for a large-population and incomplete-information game, little can be said about this issue with current game theory, nor have we observed any converging phenomenon in the experiments, even in a static setting where dynamic peer joining/leaving is disabled. In conclusion, in terms of simplicity and obtainable performance, the random allocation strategy appears to be the best choice for individual peers.

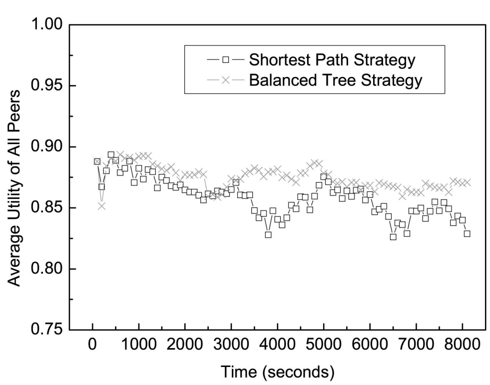


Fig. 13. Average utility of all peers over time. $\alpha = 2.5$, $\beta = 0.05$.

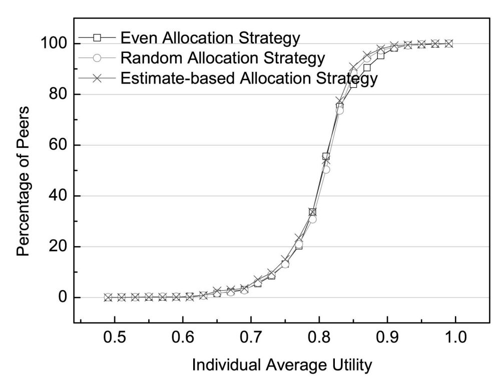


Fig. 15. CDF of individual utilities under the BT strategy.

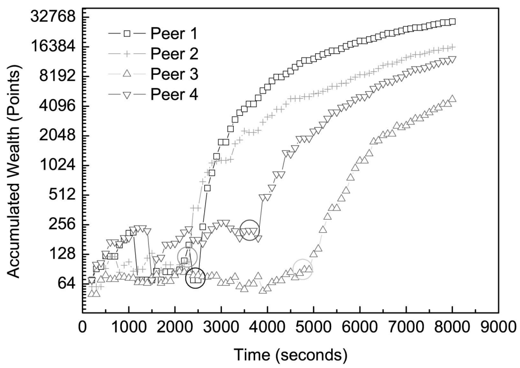


Fig. 16. Point accumulation process of some typical peers.

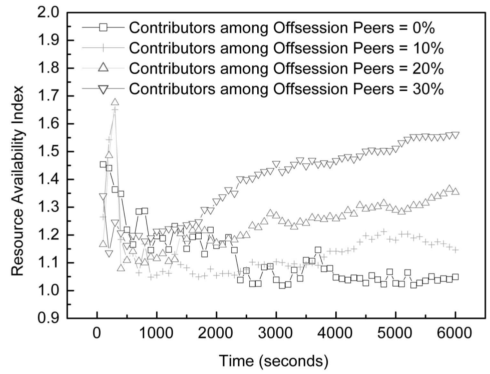


Fig. 17. System resource availability index.

7.5 Effect of Off-Session Point Accumulation

This section examines how the off-session point accumulation mechanism affects individual and system performance. Fig. 16 plots the wealth of four typical peers changing over time. It can be seen that before their departures (denoted by the circles in the figure), a peer's accumulated wealth fluctuates around some constant level. After it leaves the session, the peer's wealth increases at an approximately linear speed.

To measure the system resource, a metric called the *resource availability index* is defined as $\frac{\sum_{i=1}^{N} O_i}{N \times S}$, that is, the average number of bandwidth slots one peer can use. A high index means that a peer can find data suppliers more easily. Fig. 17 shows how this mechanism increases the overall system resource given the different ratios of leaving peers that are willing to continue to make contributions. In this experiment, the original average bandwidth of peers is set to be less than the full media rate. It can be seen that without the contribution from leaving peers, the resource availability index remains below 1 in the steady state (after 2,500 seconds), whereas with a contribution ratio of 20 percent, the resource availability index gradually increases beyond 2 after the network enters a steady state. In addition, the higher the contribution ratio, the more quickly the index increases.

8 Conclusions and Future Work

This paper introduces a payment-based incentive and service differentiation mechanism. The P2P overlay network is viewed as a market, in which peers earn points by forwarding data to others and compete for good parents using these points. We design a distributed algorithm for peers to efficiently find parents. More specifically, we discuss two strategies and analyze their equilibrium properties from a game-theoretic perspective. Bidding strategies are also designed for a peer to maximize its own utility. Finally, a mechanism is provided for off-session peers to continue to make contributions by rewarding them with points that can be used in future services. The experimental results demonstrate the effectiveness of the proposed mechanism.

In our current design, there are some factors that restrict the system's scalability. First, the banking server(s) is still a centralized component for the sake of easy management and security control. We are now looking for a distributed solution that can remove this potential bottleneck. Second, in the current protocol, the starting bidding point is always the root of the tree, which may result in high overheads on the source node when the network population is large. We will consider how to mitigate this problem using the notion of a *local market*, where peers are clustered by proximity, and most peer competitions take place locally. In "local markets," most peers do not need to submit their first bids to the root; instead, they submit to some "superpeers" that are found high in the tree and show good stability. This way, the scalability problem can be alleviated.

Appendix A
Proof of Theorem 1

(Sketch) For any player i, since its opponents choose each slot with equal probability, it will be in a d-ary tree with the same mean number of nodes $(I-1)/m+2$, no matter which slot it initially chooses. Furthermore, the probability that it will be at a given tree level is the same for all trees. This means that the probability that the accumulative service latency l_i or loss rate r_i (which equals the latency or loss rate of a single hop times the tree level number) takes on some given value is the same. Considering that $E(u_i) = \int\int u_i(l_i, r_i)\mathrm{d}l_i\mathrm{d}r_i$, its expected payoff is therefore identical for joining all trees. That is, $u_i(s_i, \sigma_{-i}) = u_i(s'_i, \sigma_{-i})$ for all s_i, $s'_i \in \{1, 2, \ldots, m\}$. This leads to the conclusion that $\sigma = (\sigma_1, \sigma_2, \ldots, \sigma_I)$ constitutes a Nash Equilibrium of game Γ_N using the classic method of identifying the Nash Equilibrium [14].

Appendix B
Proof of Theorem 2

Let a peer's two bids for substreams s_1 and s_2 be the *first bid* and the *second bid*, respectively. Since $C_2 \leq C_1$, the expected number of bids lost by peer 1 will be at most 1 (two peers offering the same price for a bid will each win 1/2 bids in expectation). We first look at the case in which peer 1 loses the first bid. If peer 2 offers a price $x \in [0, C_2]$, peer 1 needs to allocate no more than x points to lose such a bid. Therefore, under the random allocation scheme, we have

$$\Pr\{\text{peer 1 loses the first bid}\} = \sum_{x=0}^{C_2} \frac{1}{C_2+1}\left(\frac{1}{2}\cdot\frac{1}{C_1+1}+\frac{x}{C_1+1}\right) = \frac{C_2+1}{2(C_1+1)}.$$

Since the first and the second bids are symmetric, the probability of peer 1 losing one bid is $\Pr\{\text{peer 1 loses one bid}\} = 2\cdot\Pr\{\text{peer 1 loses the first bid}\} = \frac{C_2+1}{C_1+1}$. Hence, peer 1's expected number of bids won is

$$u_1(\sigma) = \Pr\{\text{peer 1 loses one bid}\}\cdot 1 + (1-\Pr\{\text{peer 1 loses one bid}\})\cdot 2 = \frac{2C_1 - C_2 + 1}{C_1+1}.$$

Clearly, $\Pr\{\text{peer 2 wins onebid}\} = \Pr\{\text{peer 1 loses one bid}\} = \frac{C_2+1}{C_1+1}$, so

$$u_2(\sigma) = \Pr\{\text{peer 2 wins one bid}\}\cdot 1 = \frac{C_2+1}{C_1+1}.$$

Now, given peer 1's mixed strategy σ_1, suppose that peer 2 chooses a pure strategy $(y, C_2 - y), y \in [0, C_2]$. Then, its probability of winning one bid is $\Pr\{\text{peer 2 wins the first bid}\} + \Pr\{\text{peer 2 wins the second bid}\} = (\frac{1}{2}\cdot\frac{1}{C_1+1}+\frac{y}{C_1+1}) + (\frac{1}{2}\cdot\frac{1}{C_1+1}+\frac{C_2-y}{C_1+1}) = \frac{C_2+1}{C_1+1}$, which is independent of y. This means that no matter how it chooses its bidding prices, peer 2's expected payoff remains the same. Thus, peer 2 would be indifferent in choosing any pure strategy $s_2 \in S_2$. The same situation applies to peer 1 given peer 2's mixed strategy σ_2. Therefore, (σ_1, σ_2) constitutes a Nash Equilibrium.

Appendix C

Proof of Theorem 3

Consider a solution $(\bar{b}_1, \bar{b}_2, \ldots, \bar{b}_S)$ and $(\bar{o}_1, \bar{o}_2, \ldots, \bar{o}_S)$ to the original problem defined by expressions (8)-(10). Since $E_1(\bar{b}_1+\bar{b}_2) \geq E_1(\bar{b}_1)$, we have $E_1(\bar{b}_1+\bar{b}_2)\cdot\bar{o}_1 \geq E_1\cdot\bar{o}_1$. Using the observation (3) in Section 6, we have $E_1(\bar{b}_1+\bar{b}_2) = E_2(\bar{b}_1+\bar{b}_2) \geq E_2(\bar{b}_2)$. It follows that $E_1(\bar{b}_1+\bar{b}_2)\cdot(\bar{o}_1+\bar{o}_2) \geq E_1\cdot\bar{o}_1 + E_2\cdot\bar{o}_2$. This means that vectors $(\bar{b}_1+\bar{b}_2, 0, \ldots, \bar{b}_S)$ and $(\bar{o}_1+\bar{o}_2, 0, \ldots, \bar{o}_S)$ also maximize the objective expression (8). Continuing the deduction in this way, we find that vectors $(\sum_{j=1}^{S}\bar{b}_j, 0, \ldots, 0)$ and $(\sum_{j=1}^{S}\bar{o}_j, 0, \ldots, 0)$ are the solution of the original problem, and expressions (8)-(10) reduce to (11) and (12), where $x = 1$, and $b_1 = \sum_{j=1}^{S}\bar{b}_j$. Taking x as any integer from $\{1, 2, \ldots, S\}$, the above analysis still holds, and thus, this theorem is proved.

Acknowledgments

The authors are grateful to the anonymous reviewers for their excellent feedback. This work was funded in part by the Engineering and Physical Sciences Research Council (EPSRC) under grant EP/F000936/1.

References

[1] K. Binmore, *Fun and Games: A Text on Game Theory.* D.C. Heath, Oct. 1991.

[2] C. Buragohain, D. Agrawal, and S. Suri, "A Game Theoretic Framework for Incentives in P2P Systems," *Proc. IEEE Int'l Conf. Peer-to-Peer Computing (P2P),* 2003.

[3] M. Castro, P. Druschel, A.-M. Kermarrec, A. Nandi, A. Rowstron, and A. Singh, "SplitStream: High-Bandwidth Multicast in a Cooperative Environment," *Proc. 19th ACM Symp. Operating Systems Principles (SOSP '03),* Oct. 2003.

[4] Y. Chu, J. Chuang, and H. Zhang, "A Case for Taxation in Peer-to-Peer Streaming Broadcast," *Proc. ACM SIGCOMM Workshop Practice and Theory of Incentives and Game Theory in Networked Systems (PINS '04),* Aug. 2004.

[5] Y. Chu and H. Zhang, "Considering Altruism in Peer-to-Peer Internet Streaming Broadcast," *Proc. 14th ACM Int'l Workshop Network and Operating Systems Support for Digital Audio and Video (NOSSDAV '04),* June 2004.

[6] B.N. Chun, P. Buonadonna, A. AuYoung, C. Ng, D.C. Parkes, J. Shneidman, A.C. Snoeren, and A. Vahdat, "Mirage: A Microeconomic Resource Allocation System for Sensornet Testbeds," *Proc. Second IEEE Workshop Embedded Networked Sensors,* May 2005.

[7] M. Feldman, C. Papadimitriou, J. Chuang, and I. Stoica, "Free-Riding and Whitewashing in Peer-to-Peer Systems," *Proc. ACM SIGCOMM Workshop Practice and Theory of Incentives in Networked Systems (PINS),* 2004.

[8] D. Fudenberg and D.K. Levine, *The Theory of Learning in Games.* MIT Press, 1998.

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

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

[11] A. Habib and J. Chuang, "Incentive Mechanism for Peer-to-Peer Media Streaming," *Proc. 12th IEEE Int'l Workshop Quality of Service (IWQoS),* 2004.

[12] K. Lai, M. Feldman, I. Stoica, and J. Chuang, "Incentives for Cooperation in Peer-to-Peer Networks," *Proc. ACM SIGCOMM Workshop Economics of Peer-To-Peer Systems (P2PECON' 03),* June 2003.

[13] R.T.B. Ma, S.C.M. Lee, J.C.S. Lui, and D.K.Y. Yau, "A Game Theoretic Approach to Provide Incentive and Service Differentiation in P2P Networks," *Proc. ACM SIGMETRICS,* 2004.

[14] A. Mas-Colell, M.D. Whinston, and J.R. Green, *Microeconomic Theory.* Oxford Univ. Press, 1995.

[15] D. Mills, *Simple Network Time Protocol (SNTP) Version 4 for IPv4, IPv6 and OSI,* Network Working Group RFC 2030, Oct. 1996.

[16] C. Ng, P. Buonadonna, B.N. Chun, A.C. Snoeren, and A. Vahdat, "Addressing Strategic Behavior in a Deployed Microeconomic Resource Allocator," *Proc. ACM SIGCOMM Workshop Economics of Peer-To-Peer Systems (P2PECON),* 2005.

[17] V.N. Padmanabhan, H.J. Wang, P.A. Chou, and K. Sripanidkulchai, "Distributing Streaming Media Content Using Cooperative Networking," *Proc. 12th ACM Int'l Workshop Network and Operating Systems Support for Digital Audio and Video (NOSSDAV '02),* May 2002.

[18] S. Saroiu, P. Gummadi, and S. Gribble, "A Measurement Study of Peer-to-Peer File Sharing Systems," *Proc. Multimedia Computing and Networking (MMCN),* 2002.

[19] N. Semret, R.R.-F. Liao, A.T. Campbell, and A.A. Lazar, "Pricing, Provisioning and Peering: Dynamic Markets for Differentiated Internet Services and Implications for Network Interconnections," *IEEE J. Selected Areas in Comm.,* vol. 18, no. 12, Dec. 2000.

[20] J. Sommers, P. Barford, N. Duffield, and A. Ron, "Improving Accuracy in End-to-End Packet Loss Measurement," *Proc. ACM SIGCOMM,* 2005.

[21] K. Sripanidkulchai, A. Ganjam, B. Maggs, and H. Zhang, "The Feasibility of Supporting Large-Scale Live Streaming Applications with Dynamic Application End-Points," *Proc. ACM SIGCOMM,* 2004.

[22] K. Sripanidkulchai, B. Maggs, and H. Zhang, "An Analysis of Live Streaming Workloads on the Internet," *Proc. ACM SIGCOMM Internet Measurement Conf. (IMC),* 2004.

[23] G. Tan and S.A. Jarvis, "Improving the Fault Resilience of Overlay Multicast for Media Streaming," *IEEE Trans. Parallel and Distributed Systems,* vol. 18, no. 6, June 2007.

[24] D.A. Tran, K.A. Hua, and T.T. Do, "A Peer-to-Peer Architecture for Media Streaming," *IEEE J. Selected Areas in Comm.,* special issue on recent advances in service overlay networks, vol. 22, Jan. 2004.

[25] D. Turner and K. Ross, *The Lightweight Currency Protocol,* Internet draft, draft-turner-lcp-00.txt, Sept. 2003.

[26] E. Veloso, V. Almeida, W. Meira, A. Bestavros, and S. Jin, "A Hierarchical Characterization of a Live Streaming Media Workload," *IEEE/ACM Trans. Networking,* vol. 14, no. 1, 2006.

[27] S.C.V. Vishnumurthy and E.G. Sirer, "KARMA: A Secure Economic Framework for P2P Resource Sharing," *Proc. ACM SIGCOMM Workshop Economics of Peer-To-Peer Systems (P2PECON),* 2003.

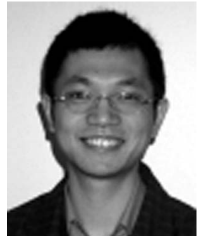
[28] W. Wang and B. Li, "Market-Driven Bandwidth Allocation in Selfish Overlay Networks," *Proc. IEEE INFOCOM,* 2005.

[29] W. Wang and B. Li, "To Play or to Control: A Game-Based Control-Theoretic Approach to Peer-to-Peer Incentive Engineering," *Proc. 11th IEEE Int'l Workshop Quality of Service (IWQoS),* 2003.

[30] B. Yang and H. Garcia-Molina, "PPay: Micropayment for Peer-to-Peer Systems," *Proc. ACM Conf. Computer and Comm. Security (CCS),* 2003.

[31] E.W. Zegura, K. Calvert, and S. Bhattacharjee, "How to Model an Internetwork," *Proc. IEEE INFOCOM,* 1996.

[32] *KaZaA Web Site,* http://www.kazaa.com, 2006.



Guang Tan received the BS degree from the Chongqing University of Posts and Telecommunications, China, in 1999, the master's degree from the Huazhong University of Science and Technology, China, in 2002 and the PhD degree in computer science from the University of Warwick, UK, in 2007. He is currently a postdoc researcher in the ASAP Group, French National Institute for Research in Computer Science and Control (INRIA), Rennes, France. From 2002 to 2004, he worked in the industry as a software engineer. His areas of interest include distributed systems, Internet services, and multimedia computing. He is a member of the IEEE.

Stephen A. Jarvis is the deputy chair and the head of research in the Department of Computer Science, University of Warwick. While previously at the Oxford University Computing Laboratory, he worked on the development of performance tools with Oxford Parallel, Sychron Ltd., and Microsoft Research, Cambridge. He has close research ties with the industry, including current projects with IBM, BT, Hewlett-Packard Research Laboratories, and AWE. He has authored more than 120 refereed publications (including three books) in the area of software and performance evaluation. He is the principal investigator on the supporting EPSRC Grant Optimizing Peer-to-Peer Aggregation Performance through Overlay Adaptation, which draws on research collaboration with the Vrije Universiteit Amsterdam and the Australian Commonwealth Scientific and Industrial Research Organization. He is also co-organizer for one of the UK's high-end scientific computing training centers, the manager of the Midlands e-Science Technical Forum on Grid Technologies and a member of the EPSRC and BBSRC Peer Review Colleges. He is member of the IEEE.