A Payment-based Incentive and Service Differentiation Mechanism for Peer-to-Peer Streaming Broadcast

Guang Tan, and Stephen A. Jarvis
Department of Computer Science, University of Warwick
Coventry, CV4 7AL, United Kingdom
Email: {gtan,saj}@dcs.warwick.ac.uk

Abstract— **We proposes a novel payment-based incentive mechanism 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 which 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 keep staying 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. Simulations results show that 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 maintaining a high system-wide performance.**

I. Introduction

The high scalability of Peer-to-Peer (P2P) systems strongly 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 attentions from researchers. Among the various application domains, high-bandwidth live media streaming presents special challenges that differ from those of applications such as P2P file-sharing. In P2P streaming, bandwidth becomes the bottleneck resource, and peers' quality of service (QoS) highly depends on the available bandwidth of the overlay network.

Chu et al. [3] first consider altruism as a key element of P2P streaming broadcast. They show that the level of altruism has very important impact on the overlay; even a small degree of altruism brings significant benefits to overall system performance. In [2], 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 (i.e., the aggregate utility of all peers) is hence improved through the redistribution of wealth (i.e., individual benefits in terms of 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 [6] 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 users 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, while low-score peers have limited options in parent selection, and hence receive low quality streaming. In a payment-based [19] system, the peer network is treated as a market, in which each overlay node plays the dual roles of service consumer and provider. Consumers try to buy the best possible service from service providers at a minimum price, while 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 both critical to streaming quality.

In this paper, we propose a new incentive mechanism for P2P streaming. Our mechanism uses a internal currency called *points* to represent a peer's contribution level, which is implicitly converted to the ability to compete for good parents. While this paradigm is similar to that of the two approaches mentioned above, our design exhibits 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, while at the same time maintaining a high overall system performance.

The rest of the paper is organized as follows. Section II documents some related work; Section III presents an overview of the proposed mechanism; Section IV presents the algorithm of reorganizing the overlay, and analyzes strategies for peers to choose their parents; Section V discusses possible strategies that can be used to maximize a peer's expected media quality; Section VI considers how to encourage off-session peers to make contributions through a point-awarding approach; Section VII evaluates the performance of the proposed schemes through simulations and Section VIII concludes the paper.

II. Related Work

There are a variety of ways to quantify a peer's contribution/participation level in the network, based on which an incentive mechanism can be designed. A commonly approach method is using some kind of currency. The Lightweight Currency Paradigm (LCP) [16] 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 [18] 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 [21] uses the *floating, self-managed currency* to greatly reduce bank node involvement, and thus improve trade efficiency. This paradigm can serve as a basic payment mechanism for our system.

A second approach for differentiating peers is to exploit the peers' different reputations. With such a mechanism [5], peers earn 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 [23] 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 differentiated service (DiffServ) engineering. Buragohain et al. [1] analyze the optimal strategies of individual peers

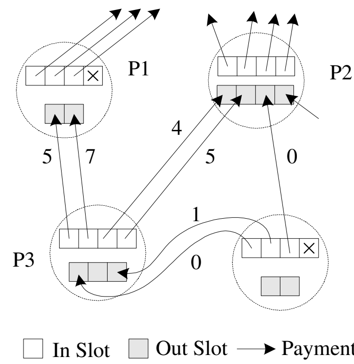

Fig. 1. System Overview.

and the possible Nash Equilibria that can be obtained under different situations in the context of file sharing. Considering the high degree of dynamics in a real system where the supply-demand relationship keeps changing, Wang et al. [20] 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.

It should be noted that we are not the first to use the auction-based approach for DiffServ. In [11], Semret et al. propose a decentralized auction-based paradigm for pricing of 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". Their work is again focused on economic analysis for general DiffServ requirements; in contrast, our work pays more attention to the problems specific to the P2P streaming broadcast.

III. 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. As in [19], it is also assumed that a lightweight, secure payment mechanism among peers is in place, which has been well studied as a building block of peer-to-peer economics (e.g., [21]). 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 scalable Internet-scale global time service with accuracies of 50 ms [8].

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"), 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 convenience of discussion, the list of notation is given in Table I.

Notation	Definition
N	total number of peers in the network
S	number of substreams
T_m	the mth 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_i^j	service latency of substream j of peer i
d_i^j	data loss rate of substream j of peer i
b_i^j	peer i's bid price for substream j
u_i^j	utility of substream j of peer i
U_i	total utility of peer i

TABLE I

NOTATION AND VARIABLE DEFINITIONS.

A. Market Model

In the dimension of time, the data stream is also divided into fixed length time periods $T_0, T_1, \cdots$, which can be as long as several minutes (e.g., 4 minutes). Each peer possesses a certain number of points (i.e., 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 constant reward of A (e.g., 5) points, from the the source node, which is assumed to have an infinite number of points. The purpose of the bonus is to stimulate peers to serve newly arriving peers with zero points to offer. Figure 1 gives an example scenario of the market trading. In this example, 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 specifically target wealth accumulation rather than QoS optimization.

The start times of periods $T_0, T_1, T_2, \cdots$ actually define the preemption points for peers with different wealth. During each period, peers compete with each other for good parents (e.g., those near the source) for the next period in a *first-price auction* like procedure: peers submit sealed-bids simultaneously to their desired parents, while parents always choose the highest bidders as their children. As a result, wealthy peers are able to choose desired parents, while 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. While large wealth gaps may exist between peers, the probability of resource-poor peers suffering starvation is no higher than in a non-incentive network since under the stimulation of the bonus, peers with spare out slots are always more likely to serve more peers than in a non-incentive network, even though the marginal utility of wealth is decreasing. Note that we do not claim to completely solve the starvation problem, since this ultimately depends on the aggregate amount of resource (either contributed or reserved by individuals) in the network; our aim here is to achieve a lower probability of starvation by encouraging peers to contribute more usable resource.

B. 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 a utility value using a utility function. Let integer $l_i^j(t) \geq 1$ 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 at time t, $d_i^j(t) \in [0, 1]$ denote the accumulative data loss rate of substream j received by peer i at time t, then the utility of a single substream j at time t can be expressed as

$$u_i^j(t) = \frac{\ln\left\{1 + \max[0, 1 - \alpha \cdot d_i^j(t)]\right\}}{\ln 2 \cdot [l_i^j(t)]^\beta}, \tag{1}$$

Where $\alpha > 0$ and $0 < \beta \leq 1$. The numerator part considers the impact of loss rate on the media quality. Parameter α reflects the decreasing speed of utility over the increase of loss rate. The concave function $\ln(\cdot)$ indicates that the utility decreases more slowly at a small loss rate than when the loss rate approaches $1/\alpha$, where the utility becomes zero. The denominator part of Eq. 1 uses another concave function to include the factor of service latency, with parameter β indicating the importance of service latency to utility. For applications without user interactions, service latency is less important and β can be fairly small (e.g., 0.1). Note that we choose not to use a $\log_x(\cdot)$ function for the latency because we want to normalize the utility to $[0, 1]$, after which the effect of base x would be eliminated. In this paper we assume that all peers have the same α and β.

Further, the collective utility of a peer's all substreams can be characterized as follows:

$$U_i(t) = \ln(1 + \sum_{j=1}^{S} u_i^j(t)). \tag{2}$$

The use of $\ln(\cdot)$ again captures the diminishing returns of an increased media bit rate on the perceived media quality.

IV. 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.

A. 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 (e.g., 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 finds 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 Figure 2.

In Figure 2 (a), there are 7 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 Figure 2 (b). 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 next subsection. As a possible scenario as shown in Figure 2 (c), peers 7 and 2 choose 9, while peers 6, 5 and 0 choose 8 as their new targets and begin a new round of competition. As a result, peer 2 and 0 are ruled out and peer 7, 6, 5 find their positions in the tree (Figure 2 (d)). Using the same method of finding new parents, peer 2 and 0 finally find 7 and 5 as their parents respectively (Figure 2 (e)), and this finishes the virtual overlay construction.

In fact, 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 $\max(x/y, z)$, where x is the length of a period, y the maximum possible number of levels in the tree, and z a threshold value often set to a few 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. In the case that some determined parent leaves the overlay before the next period starts, its children peers re-find their future parents as follows. They query the source node for possible joining opportunities. If there are no spare out slots available, the source node redirects their requests to peers at the next level; and if no chances of joining again, they are further redirected down the tree until they find their parents or have reached the leaf level; in the latter case the peers randomly choose their parents in a best-effort manner. The real payments are made at the beginning of the next period. A parent peer leaving in the middle of a period should return all the points earned in preceding period; or its child peers can report this cheating behavior to others, hurting the parent's reputation and finally affect its income.

The parent finding procedure always starts from a certain node, called the *starting point*. Ideally, the starting point for every peer is the source node as this provides the most opportunities to find desired parents. However, when the population is very large, the bidding/response message traffic may overwhelm the source node; and the bidders themselves may experience response timeout and have to choose future parents randomly, possibly resulting in worse QoS than they deserve. This problem can be avoided using a simple heuristic. Each peer maintains the addresses of a certain number (e.g., 30) of peers at levels between $[a + 1, a + b]$, where a is the level number of itself and b a small integer (e.g., 2). A peer then chooses the nearest parent among this set as its starting point. The rationale behind this heuristic is that low-level peers with very limited wealth are not likely to win parents at high levels, so it is more realistic for them to choose peers not so far from itself up in the tree.

B. 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 one new target parent from the winners (which become the children of the original parent). There are three possible strategies to do this, namely the Shortest Path (SP) strategy, the Balanced Tree (BT) strategy and the Shortest Path and Balanced Tree (SP-BT) strategy, as will be introduced as follows.

a) Shortest Path (SP) Strategy: With this strategy, a peer selects a parent from the candidates that make the accumulated service latency 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 loss rate within a short time window [12].

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 very tall. On the other hand, some peers may not even attract enough peers to fill their slots and become leaf nodes early. In a highly unbalanced tree, a peer's expected overlay path length, defined as the number of overlay hops from the source to itself, can be much larger than in a balanced tree. An example of this strategy is illustrated in Figure 3(a), in which six peers are competing for two out slots on a single parent. As a result, the subtree under the parent will be at least two levels high, while the sibling peers cannot attract child peers and become leaf nodes.

b) Balanced Tree (BT) Strategy: A peer with this strategy chooses a candidate parent probabilistically in an attempt to

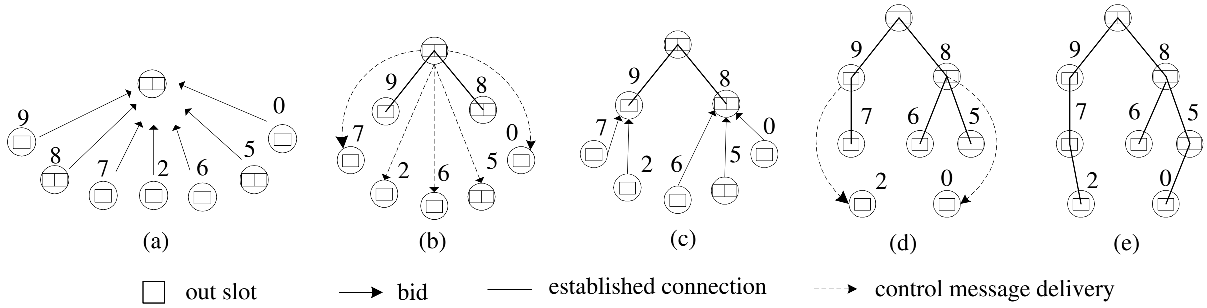

Fig. 2. An example of distributed virtual overlay construction. The numbers represent the bid prices.

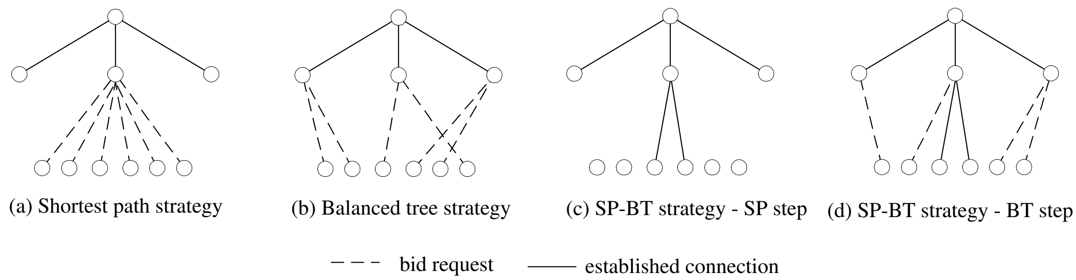

Fig. 3. Illustration of the different parent selection strategies. Each node has two out slots.

balance the tree. Given a set of candidates, the probability of one peer being picked in proportional to its number of out slots. See Figure 3(b) for 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 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 packet loss rate, and finally increase the average utility of peers (see Eq. (1)). 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, thereby reducing the average utility. As a result, how system performance benefits from the two strategies ultimately depends on the relative importance of loss rate and service latency to 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 more important to look at it from the viewpoint of individual peers. One important 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? Or in terms of game theory [7], 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 $S_i = \{1, 2, \cdots, m\}$, and $u_i(\cdot)$ is player i's payoff function as defined in Eq. (1) ? The following theorem shows that if each peer chooses the out slot with equal probability, then a Nash Equilibrium can be achieved.

Theorem 1: Define mixed strategy for player i, $\sigma_i : S_i \rightarrow [0, 1]$ as the probabilities assigned to all pure strategies $s_i \in S_i$, and let $\sigma_i(j) = 1/m \ \forall j \in \{1, 2, \cdots, m\}$ and $\forall i \in \{0, 1, \cdots, I\}$, then mixed strategy profile $\sigma = (\sigma_1, \sigma_2, \cdots, \sigma_I)$ constitutes a Nash Equilibrium of game Γ_N.

Proof: Refer to [15]. ■

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 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 respectively l and r, which are both constants and can be interpreted as their expected values. Theorem 1 therefore applies and the BT strategy leads to a game equilibrium. However, the assumption does not necessarily hold in practise – a peer can easily measure the latency between itself and all the parents, making the expected payoff unequal for choosing different parents. That is, a peer would not be indifferent between the parents; instead it tends to choose a near parent using the SP strategy when others choose the BT strategy, thus deviating from the desired equilibrium. This problem can be resolved by a combination of the SP and BT strategies, as introduced in the next section.

c) Shortest Path and Balanced Tree (SP-BT) Strategy: A peer first chooses a parent using the SP strategy (see Figure 3(c)). If it fails to win a slot on that parent, it uses the BT strategy to choose a parent (see Figure 3(d)).

The SP-BT strategy is intended to achieve an equilibrium based on the BT strategy. Peers (referred to as *seekers*) first bid for their nearest parents from a subset of peers (referred to as *targets*). The seekers that win the competitions become the children of the targets, and the list of winners is temporarily held by these targets. Now a failed seeker can only choose a target as its grandparent (which may turn out to be the seeker's parent if the target has not attracted enough children before, but the seeker would be unaware of the fact), and it no longer knows the distance between itself and the target because some unknown parents are now situated between itself and the target. Consequently, the seeker will be indifferent in its choice of target when choosing a grandparent. At this point the BT strategy becomes a sensible choice for all seekers.

C. Extensions 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 are more than one substreams, peers need to determine how to assign the out slots to the different substreams. 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.

D. Security Issue

Under the DiffServ rules, the resource-rich peers generally are more competitive than others and thus are likely to keep staying high in the tree. This leaves the possibility that hackers equipped with plenty of points occupy all high-level tree positions and then issue a denial-of-service attack to block critical broadcast services, although the blocking may only last a short period of time. Indeed, this issue is common to all incentive schemes for tree-based multicast systems where end-to-end latency is one of the major approaches to awarding individuals. Our current solution is to reserve a certain proportion of root out slots (which we assume are nontrivial) for serving peers in a non-incentive manner (e.g., First-Come, First-Served). This is actually a tradeoff choice between encouraging individual contributions and guaranteeing system security: the more root slots open for competition, the more effective the incentive rule is, but the more likely the system is exposed to denial-of-service risk. A deeper investigation of this issue is left as a future subject of research.

V. In-Session Utility Maximization

During each period T_m, a peer i needs to plan for the bids for each substream using the C_i points earned before the start of T_m, with the objective to maximize its expected utility during T_{m+1}. Let the bids for substream $1, 2, \cdots, S$ be $b_i^1, b_i^2, \cdots, b_i^S$ respectively, and let functions $L_i^j(\cdot)$ and $D_i^j(\cdot)$ denote the mappings from bid b_i^j to the expected service latency and loss rate of substream j respectively, then this issue can be formulated as the following optimization problem:

$$\text{Maximize } U_i = \sum_{j=1}^{S} u_i^j \tag{3}$$

$$= \sum_{j=1}^{S} \frac{\ln\left[1+\max(0, 1-\alpha d_i^j)\right]}{\ln 2 \cdot (l_i^j)^\beta} \tag{4}$$

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

$$\text{Subject to } \sum_{j=1}^{S} b_i^j = C_i. \tag{6}$$

Unfortunately, both $L_i^j(\cdot)$ and $D_i^j(\cdot)$ 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. One solution to this problem is to estimate $L_i^j(\cdot)$ and $D_i^j(\cdot)$ from past experience, as described as follows.

First, a peer i maintains a mapping $\widehat{BH}_i^j$ from bid b_i^j to the average tree level number h_i^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 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 $\widehat{HL}_i^j$ from its tree level number h_i^j to the service latency l_i^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 $\widehat{HD}_i^j$ from its level number h_i^j to the loss rate d_i^j for each substream.

Given a bid b_i^j, peer i first use $\widehat{BH}_i^j$ to estimate the level number it is expected to be at in the tree. The service latency l_i^j and the data loss rate d_i^j can then be estimated using $\widehat{HL}_i^j$ and $\widehat{HD}_i^j$ respectively. Using these estimates a utility can be computed. So far we have established the mapping from a bid b_i^j to the expected utility u_i^j.

The next step is to resolve the optimization problem defined by Eq.s (5) and (6). 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 between the execution time and solution quality, and hence is suited to a time-constrained scenario.

The above method is essentially a myopic strategy; that is, 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. 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 becomes a plausible choice [4].

In view of the strong assumptions of the myopic optimization method, we also develop a simple static strategy: that allocates points to all bids evenly. This strategy however does not lead to a Nash Equilibrium: consider a simple example of two peers, 1 and 2 with C_1 and C_2 ($C_1 > C_2$) points respectively, competing for two substreams s_1 and s_2 on some parent node. It can be seen that the even allocation will make peer 2 fail in each competition (because $\frac{C_2}{2} < \frac{C_1}{2}$), hence becoming a *dominated strategy* for peer 2. Clearly, a better choice for 2 would be randomly choosing its two bids under the point constraint C_2, making its winning probability higher. Despite this, we will use this strategy as a baseline scheme for performance comparison in later experiments.

VI. 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 individual benefit, such a peer will seek to maximize its wealth instead of media quality. Here a decision problem is: during each period, how to allocate points from its current wealth W_i to the substream bids so that its expected income in term of points during next period is maximized? If we let the vectors $(b_i^1, b_i^2, \cdots, b_i^S)$ and $(o_i^1, o_i^2, \cdots, o_i^S)$ denote its bids and out slot allocations for all substreams respectively, and define the function E_i^j as the mappings from b_i^j to the expected income from a single out slot of substream j, then the wealth maximization problem can be formulated as follows:

$$\text{Maximize } \sum_{j=1}^{S} E_i^j(b_i^j) \cdot o_i^j - \sum_{j=1}^{S} b_i^j \tag{7}$$

$$\text{Subject to} \sum_{j=1}^{S} b_i^j < W_i \tag{8}$$

$$\text{and } \sum_{j=1}^{S} o_i^j < O_i. \tag{9}$$

By analyzing the market behavior and based on experimental experience, we can make several observations: (1) $E_i^j(b_1) \geq E_i^j(b_2)$ for any $b_1 \geq b_2$, which means that a higher bidding price always brings no less expected income. This is the case since a higher bidding price leads to higher utility for a substream, which makes it more attractive to other peers. (2) $E_i^m(b) = E_i^n(b)$ 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, $E_i^j(\cdot)$ would not change as a single peer adjusts its out slot allocation (which is invisible to others), so $E_i^j(\cdot) \cdot o_i^j$ 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 2: The problem defined in expressions (7)-(9) is equivalent to the problem:

$$\text{Maximize } E_i^x(b_i^x) \cdot O_i - b_i^x \tag{10}$$

$$\text{Subject to } \ b_i^x < W_i, \tag{11}$$

Where x is any integer from $\{1, 2, \cdots, S\}$.

Proof: Refer to [15]. ∎

The above conclusion means that rather than bidding and allocating out slots for multiple substreams, we only need 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 (10) is achieved. Center to this problem is to determine $E_i^j(\cdot)$. Once again, we must resort to historic knowledge. A peer can maintain a mapping from b_i^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 $\hat{E}_i^j(\cdot)$ can be generated. Using this, the problem defined in Theorem 2 can be solved in $O(W_i)$ 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 better QoS than it can obtain without the accumulation process. From the system's perspective, the point accumulation mechanism also helps to increase overall system resources. Since in this mode, a peer consumes only one substream (i.e., 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 shutdown the hosts.

VII. Performance Evaluation

We developed an event-driven simulator based on a carefully configured simulation model. The simulator uses the GT-ITM transit-stub model [22] to generate an underlying network topology consisting of 15600 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. Of all the 15360 stub nodes, 2000 peers are randomly selected to participate 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 $[0, 0.06]$.

In all simulations, there are 8 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 [13] [10]. Let

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 II. The nodes' lifetimes follow a lognormal distribution [17] [14] with the μ (location parameter) and σ (shape parameter) set to 5.5 and 2.0 respectively, which are chosen according to the statistical findings in [17]. According to *Little's Law*, the node arrival rate λ is determined from M divided by the mean lifetime, i.e. 1809 seconds. The streaming period is by default 120 seconds; the α, β parameters in the utility evaluation function are by default 1 and 0.25, respectively. The bonus value (see Table I) is set to 10.

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	-

TABLE II
DEFAULT BANDWIDTH SETTING USED IN THE SIMULATION.

A. 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. Figure 4 plots the results against 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 QoS of peers with different service capacities. A second observation is that after using incentive, most peers have substantially higher utilities than they do without it.

Figures 5 and 6 can serve as an explanation for the above phenomena. In Figure 5, *individual average tree level number,* defined as the average tree level number of a peers' all substreams averaged over all periods in its lifetime, is plotted against the outgoing bandwidth. It can be seen that with incentive, peers' average tree level numbers are far smaller than they are in a non-incentive 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. Figure 6 shows the *individual average network stretch* against peer's bandwidth. A peer's *network stretch* is the ratio of its accumulative service latency to its latency from the root along a unicast path in the underlying network. 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% (from 6.32 to 2.78).

B. 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. Figure 7 plots the maintenance costs under different period lengths (the "no-incentive" case is equivalent to the case of 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 9 changes per substream. Observing that the average lifetime is more than 1800 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 non-incentive 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.

While a longer period reduces individual overhead, from the system's viewpoint a shorter period offers more chances for the overlay to be adjusted and hence means better system-wide performance. As shown in Figure 8, a short period leads to higher and stabler system performance in terms of average utility of all peers, especially when the bandwidth resource is not so rich.

C. Comparison of Distributed Parent Selection Strategies

This section compares the three distributed parent selection strategies under different parameter settings. Figures 9 and 10 show the average utility as a function of time. As expected, the performance of these strategies depends on the weights of service latency (α) and loss rate (β) in the utility. When α is 1 and β 0.25 (see Figure 9), the shortest path strategy is better than the other two strategies in terms of overall performance, whereas in Figure 10, where α is changed to 2.5 and β to 0.05, the SP-BT strategy becomes the best choice.

D. Comparison of Bidding Strategies

We compare the performance of the proposed "myopic optimization" based biding strategy and the baseline strategy where a peer allocates its points to all bids evenly. For

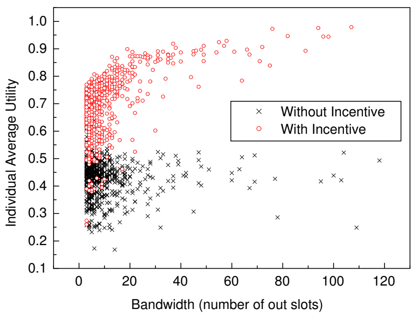

Fig. 4. Individual average utility vs. bandwidth.

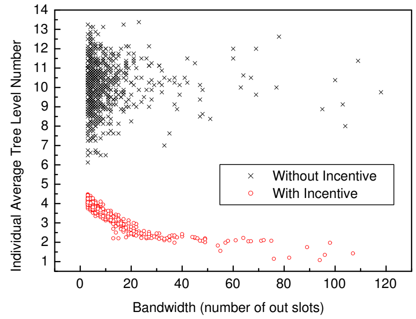

Fig. 5. Individual average tree level number vs. bandwidth.

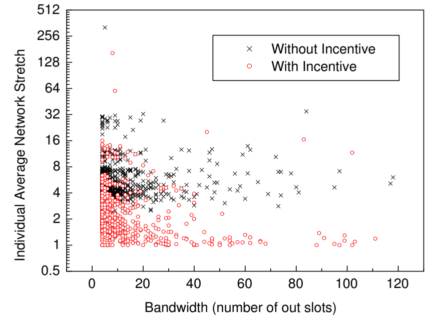

Fig. 6. Individual average network stretches vs. bandwidth.

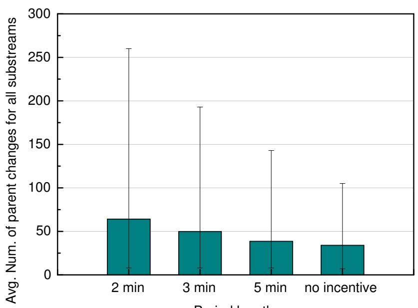

Fig. 7. Overlay maintenance cost in terms of number of parent changes.

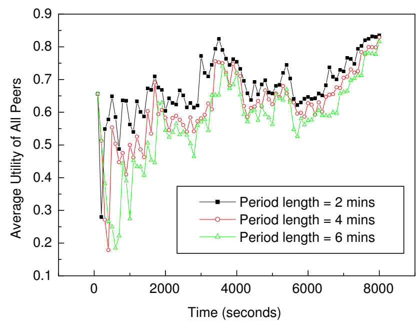

Fig. 8. Effect of period length on system performance.

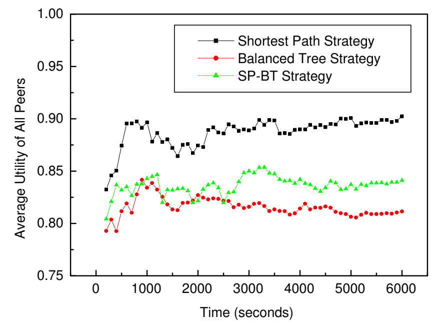

Fig. 9. Average utility of all peers over time. $\alpha = 1, \beta = 0.25$.

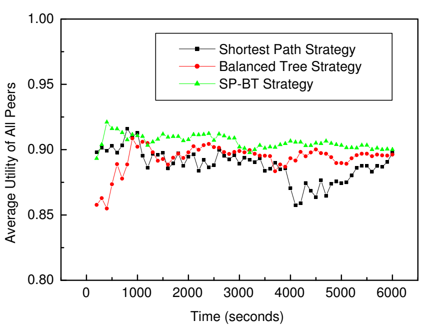

Fig. 10. Average utility of all peers over time. $\alpha = 2.5, \beta = 0.05$.

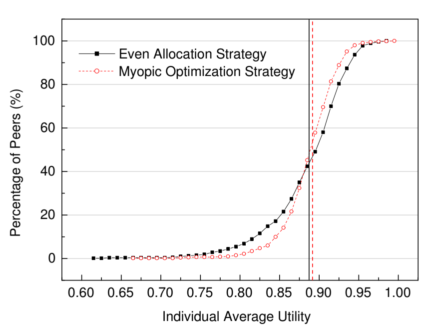

Fig. 11. CDF of individual utilities: shortest path strategy.

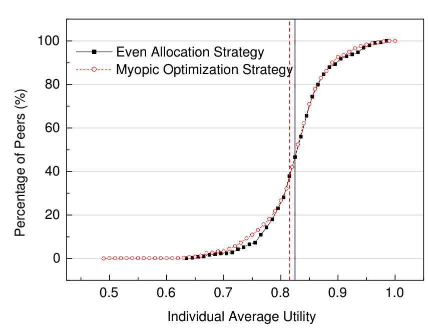

Fig. 12. CDF of individual utilities: SP-BT strategy.

the genetic algorithm used in the first 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. On a modern desktop machine, the computation takes only a few milliseconds.

Figures 11 and 12 plot the cumulative percentages of peers whose average utilities are above certain values under the shortest path and the SP-BT strategies. The results show that the myopic optimization method actually has no advantage over a simple even allocation scheme. The reason for using such a complex strategy may lie in the equilibrium achieved. With very limited knowledge about others' decisions and unpredictable system dynamics, a peer can neither find the actual optimal strategy nor take advantage of others (as in the case where all other peers use an even allocation strategy). A sensible choice for any peer is to play a best strategy according to its historic knowledge. This is in spirit like playing a *fictitious play* [4]. One should also note that a decision made on this basis is unknown by others, so there is no need for a peer to worry about being taken advantage of by others.

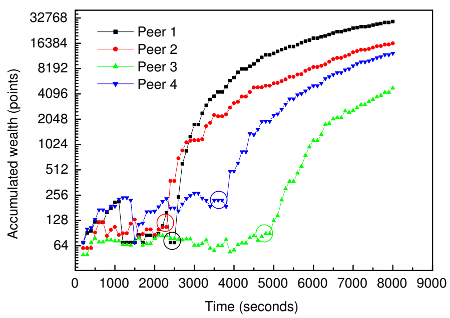

Fig. 13. Point accumulation process of some typical peers.

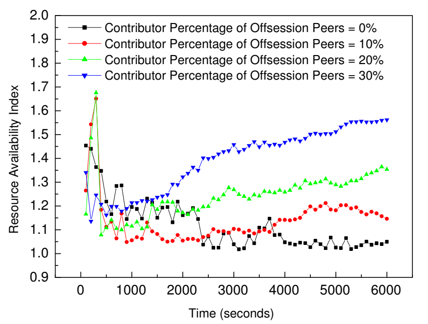

Fig. 14. System resource availability index

E. Effect of Off-Session Point Accumulation

This section examines how the off-session point accumulation mechanism affects individual and system performance. Figure 13 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 approximately a linear speed.

To measure 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 supplier more easily. Figure 14 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 keeps below 1 in the steady state (after 2500 seconds); while with a contribution ratio of 20%, 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.

VIII. Conclusions

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 three strategies 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 points which can be used in future services. The experimental results demonstrate the effectiveness of the proposed mechanism.

References

[1] C. Buragohain, D. Agrawal and S. Suri. A game theoretic framework for incentives in P2P systems. *Proc. of P2P 2003*.

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

[3] Y. Chu and H. Zhang. Considering Altruism in Peer-to-Peer Internet Streaming Broadcast. *Proc. of ACM NOSSDAV*, June, 2004.

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

[5] M. Gupta, P. Judge and M. Ammar. A reputation system for peer-to-peer networks. *Proc. of ACM NOSSDAV*, 2003.

[6] A. Habib and J. Chuang. Incentive Mechanism for Peer-to-Peer Media Streaming. *Proc. of International Workshop on Quaility of Service (IWQoS) 2004*.

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

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

[9] V. N. Padmanabhan, H. J. Wang, P. A. Chou, and K. Sripanidkulchai. Distributing Streaming Media Content Using Cooperative Networking. *Proc. of ACM NOSSDAV*, May 2002.

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

[11] 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 Journal on Selected Areas of Communications*, VOL. 18, No. 12, Dec. 2000.

[12] J. Sommers, P. Barford, N. Duffield and A. Ron. Improving accuracy in end-to-end packet loss measurement. *Proc. of SIGCOMM 2005*.

[13] K. Sripanidkulchai, A. Ganjam, B. Maggs and H. Zhang. The feasibility of supporting large-scale live streaming applications with dynamic application end-points. *Proc. of ACM SIGCOMM*, 2004, Portland, Oregon, USA.

[14] K. Sripanidkulchai, B. Maggs and H. Zhang An analysis of live streaming workloads on the Internet. *ACM SIGCOMM IMC*, 2004.

[15] G. Tan, and S. A. Jarvis. A Payment-based Incentive and Service Differentiation Mechanism for P2P Streaming Broadcast. *TR-2005-087*, Dept. of Computer Science, University of Warwick, UK.

[16] D. Turner and K. Ross. The lightweight currency protocol. Internet Draft, draft-turner-lcp-00.txt. Sept. 2003.

[17] E. Veloso, V. Almeida, W. Meira, A. Bestavros, and S. Jin. A Hierarchical Characterization of A Live Streaming Media Workload. *IEEE/ACM Transactions on Networking*, 12(5), 2004.

[18] S. C. V. Vishnumurthy and E. G. Sirer. Karma: A secure economic framework for p2p resource sharing. *Proc. of Workshop on the Economics of Peer-to-Peer Systems*, 2003.

[19] W. Wang and B. Li. Market-driven Bandwidth Allocation in Selfish Overlay Networks. *Proc. of INFOCOM 2005*.

[20] W. Wang and B. Li. To play or to control: a game-based control-theoretic approach to peer-to-peer incentive engineering. *Proc. of IWQoS 2003*.

[21] B. Yang and H. Garcia-Molina. PPay: Micropayment for peer-to-peer systems. *Proc. of ACM Conference on Computer and Communications Security (CCS) 2003*.

[22] E. W. Zegura, K. Calvert and S. Bhattacharjee. How to Model an Internetwork. *Proc. of IEEE INFOCOM '96*, San Francisco, CA.

[23] KaZaA website http://www.kazaa.com.