Service Differentiated Peer Selection: An Incentive Mechanism for Peer-to-Peer Media Streaming

Ahsan Habib and John Chuang

School of Information Management and Systems

University of California, Berkeley.

{habib,chuang}@sims.berkeley.edu

Abstract— We propose a service differentiated peer selection mechanism for peer-to-peer media streaming systems. The mechanism provides flexibility and choice in peer selection to the contributors of the system, resulting in high quality streaming sessions. Free-riders are given limited options in peer selection, if any, and hence receive low quality streaming. The proposed incentive mechanism follows the characteristics of rank-order tournaments theory that considers only the relative performance of the players, and the top prizes are awarded to the winners of the tournament. Using rank-order tournaments, we analyze the behavior of utility maximizing users. Through simulation and wide-area measurement studies, we verify that the proposed incentive mechanism can provide near optimal streaming quality to the cooperative users until the bottleneck shifts from the streaming sources to the network.

I. INTRODUCTION

Peer-to-peer (P2P) systems rely on voluntary resource contributions by individual peers. However, the inherent tension between individual rationality and collective welfare produces a misalignment of incentives in the grassroots provisioning of P2P services. Empirical studies have shown *free-riding* (consuming resources without contributing) to be prevalent in P2P file-sharing networks [1], [2], legal threats from copyright owners notwithstanding. Researchers have proposed various incentive mechanisms, based on payment, punishment, or service differentiation, to encourage cooperative behavior among peers [3], [4], [5], [6], [7].

Peer-to-peer media streaming systems present additional challenges, and opportunities, that are different from those of traditional P2P file-sharing systems. Users derive utility from P2P streaming systems not only from the availability of files per se, but from the ability to obtain high quality streams of these files. Since the quality of a streaming session depends on a combination of factors, ranging from the characteristics of the streaming sources (e.g., link capacity, availability, offered rate) to the characteristics of the network paths (e.g., available bandwidth, packet loss rate, overlap of paths from multiple sources to receiver), the challenge is to design good peer selection strategies to realize high quality streaming sessions.

We believe that peer selection for P2P streaming offers a unique opportunity to tackle both the free-riding and the streaming quality of service (QoS) challenges in a synergistic manner. We propose an incentive mechanism that provides service differentiation in peer selection for P2P streaming based on relative contribution of the peers. Contributors to the system are rewarded with flexibility and choice in peer selection, resulting in high quality streaming sessions. Free-riders are given limited options in peer selection, if any, and hence receive low quality streaming. The proposed incentive mechanism follows the characteristics of rank-order tournaments. Lazear and Rosen [8] proposed the rank order tournaments as a competition among several workers who perform similar tasks for optimum labor contracts. This scheme pays prize to the winners and losers differently, and the prize depend on the rank order of the contestants instead of their actual contributions. Many applications require only a ranking or ordinal measurement of outputs, and the tournament is the best choice for them. For example, tournaments are commonly used in industry to award promotion. In academics, the tournaments can be used in the awarding of assistantships and scholarships to graduate students. The inherent nature of P2P media streaming application makes rank order tournaments as a viable choice. Because the peer selection depends on relative contribution among the peers. Using the rank order tournaments theory, we analyze the behavior of utility maximizing users so that they can compute the optimal contribution to the system to achieve their desirable quality.

We motivate our work with a case study of the PROMISE P2P streaming system [9] in the presence of free-riding behavior in Section II. We obtain several findings. First, at low levels of cooperation, system performance is poor even when the overall load is low. Second, when a peer contributes to many streaming sessions, its streaming quality and that of its remote receivers suffer degradation. Therefore, in contrast to traditional P2P file-sharing systems, a P2P streaming system cannot be sustained by a small fraction of altruistic users. Furthermore, we confirm that streaming quality is strongly dependent on the choice of suppliers (streaming sources). A random peer selection scheme results in highly variable streaming quality, whereas an informed peer selection scheme produces high quality streaming with low variance.

In Section III, we describe the proposed incentive mechanism where a peer's contribution level determines its score and its relative ranking in the system, which in turn determines its ability to select good peers, and ultimately its

streaming quality. An analytical model is provided in this section to compute the optimal contribution of each user. We evaluate the mechanism in Section IV and find that it provides near optimal quality to the cooperative users until the bottleneck shifts from the suppliers to the network. We discuss related work in Section V, before concluding the paper in Section VI.

II. MOTIVATION

We study a P2P streaming system to understand the requirements to provide high quality streaming. Then, we evaluate the system under free riding to investigate the impact of non-cooperation on the streaming quality.

A. P2P streaming system case study

A streaming system is required to encompass the key functions of object lookup, peer-based aggregated streaming, and dynamic adaptations to network and peer conditions. The quality of a peer depends on its availability, offered rate, and capacity of its outgoing link. To avoid paths sharing among multiple suppliers, a streaming system should leverage the underlying network topology and performance information for the selection of suppliers. To cope with fluctuations in the service received from the network, several techniques can be used such as forward error correction (FEC) coding, multi-description coding, and sending rates adjustment. The system monitors the status of peers to react to peer/connection failure. A dynamic switching mechanism is required to replace a failed peer without disrupting a streaming session.

In this paper, we use PROMISE [9] as an example system that satisfies the requirements of a streaming system. PROMISE uses topology-aware peer selection to obtain the best suppliers. The topology-aware technique infers the underlying topology and its characteristics, and considers the available capacity and packet loss ratio of each link of a path. Consider the example shown in Figure 1. The figure shows a network topology of a set of potential suppliers to a receiver. The topology is annotated with available capacity of each segment of a path, the offered rate of each peer, and its availability. A random technique may choose peers P_1, P_3, P_4 as suppliers, even though some of these peers have low availability (P_1), and others share a congested path (P_3, P_4). However, the topology-aware technique selects the best set: P_2, P_3, P_6 because they provide the best aggregate rate considering their availability, offered rate, and the sender to the receiver path dynamics [9]. Later, we show that a certain level of cooperation is required to receive high quality streaming even with the topology-aware supplier selection.

A streaming session in PROMISE is established as follows: A peer requesting a media file issues a lookup request to the underlying P2P substrate, which will return a set of *candidate* peers who have the content. Usually, a streaming session requires more suppliers than a file downloading session, which implies more cooperation among users is required in a streaming environment. The candidate set of PROMISE typically contains 10 to 20 peers. A topology is constructed and annotated to connect the candidate peers
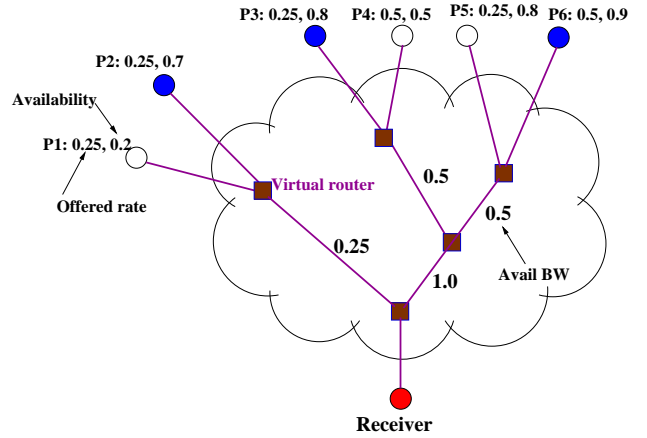

Fig. 1. Peer selection by PROMISE. P_2, P_3, and P_6 are selected because they provide the best aggregate rate considering their availability, offered rate, and the network dynamics of the paths from the senders to the receiver.

with the receiver. Using the annotated topology, the selection algorithm determines the *active* sender set that is likely to yield the best quality for this streaming session. The rest of the candidate peers are kept in a *standby* sender set, from which replacement peers will substitute failed or degraded peers from the active set. The receiver assigns a sending rate to each of the active senders. The streaming session continues as far as there is no need to *switch* to a different active sending set. A switch is needed if a peer fails or the network path becomes congested. At that time, the topology is updated with new values.

B. Impact of non-cooperation

Using packet-level simulation and Internet experiments, we study the impact of peer non-cooperation on the performance of a P2P media streaming system (PROMISE). We analyze this impact from three angles. First, how user cooperation can enhance the average quality of streaming. Second, what negative effects cooperation might bring to a peer that shares resources. Third, what a peer would gain from sharing its resources.

To quantify the performance of media streaming system, we define quality of a streaming session as:

$$Q = \frac{\sum_{i=1}^{T} Z_i}{T}, \tag{1}$$

where T is the number of packets in a streaming session and Z_i is a variable that takes value 1 if packet i arrives at the receiver before its scheduled play-out time, and 0 otherwise. The quality is different from throughput because it considers the deadline of each packet. The parameter Q captures other performance parameters such as packet delay, packet loss, and jitter. A packet that misses its deadline is discarded and, therefore, does not contribute to the quality of a streaming session, similar to a lost packet. The system quality is defined as the average quality of all receivers in the system.

Cooperation brings quality. We simulate a streaming system with 1,200 peers who are connected by a Internet-like topology generated using the Georgia Tech Internet Topology
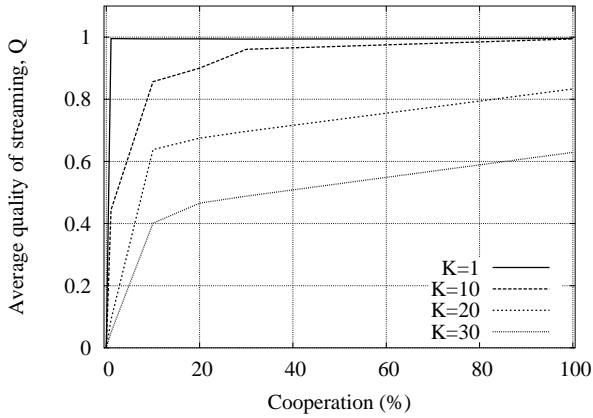
Fig. 2. Simulation data to show how the level of cooperation affects the quality Q of the system. Higher cooperation provides better quality of service. If the level of cooperation is low, the streaming quality is also low even when the network is not heavily congested. When the system has limited capacity, a large number of concurrent sessions degrade the quality of the system even when the cooperation level is 100%.

Mapping tool. Figure 2 shows that the quality of streaming sessions changes with level of cooperation among different peers. The number of concurrent sessions (denoted by K) is varied in this simulation.

When the system is almost idle ($K = 1$), the cooperation level does not affect the streaming quality. The streaming quality is the maximum at all cooperation levels because there is no contention for resources. However, the lack of cooperation negatively impacts the streaming quality even if the network is not heavily loaded. For example, when the system has 10 concurrent sessions ($K = 10$), the streaming quality is excellent as long as the level of cooperation is above 30%. The quality degrades sharply when the level of cooperation drops below this level. The performance degrades even more for larger K when the level of cooperation low. Furthermore, when the system has limited capacity, a large number of concurrent sessions degrade the quality of the system even when the cooperation level is 100%. It is clear from this experiment that cooperation is necessary to maintain the system performance within an acceptable range, and the cooperation might not be effective if the bottleneck is shifted from the suppliers to the network.

Simultaneous uploading hurts quality. We study whether there is a significant cost of sharing resources, especially the bandwidth. The term bandwidth is used to indicate the capacity of a link. We ask two questions: First, does simultaneous uploading affect the downloading quality of the users? Second, does the concurrent uploading reduce the downloading quality of the remote peers? If answer of the first question is yes, a user is even less motivated to share its bandwidth with others without any incentive. If the answer of the second question is yes, a peer should not supply to so many peers to maintain the quality of the remote receivers.

We use PROMISE implementation as a streaming system in the PlanetLab [10] test-bed to conduct experiments in the wide area Internet. We use nodes at Berkeley, Caltech, UCSD, Rice, Duke, MIT, BU, Purdue, CMU, UMass,

UTexas, Arizona, Stanford, Germany, UK, France, Italy, Sweden, and Taiwan in our experiments. Figure 3 shows the quality Q of the Berkeley node that downloads from several suppliers and uploads to several remote receivers. The X-axis is the uploading rate from Berkeley, which is varied from $0-4$ Mbps. Figure 3(a) shows that the quality of the downloading session is good even though the peer is supplying at a rate higher than 2 Mbps. The quality goes down when the rate is 3 Mbps or higher. Thus, there is a limitation about how much a peer should supply without hurting its own quality.

We measure the quality obtained by remote users downloading from Berkeley. Figure 3(b) shows the average quality and Figure 3(c) shows the individual quality of the remote users. Most of the remote users experience poor quality when Berkeley is supplying to many users. Thus, there is a cost associated with uploading: too much uploading from a peer hurts the performance of its own quality and the quality of the remote receivers. Thus, while a traditional file sharing system can be sustained with a low level of cooperation, a P2P streaming system cannot provide high streaming quality to its users if only a small fraction of users cooperate.

Random peer selection provides random quality. We observe in our experiments that the proper selection of peers is important for a P2P streaming session. Arbitrary peer selection yields unpredictable quality, which might be acceptable in file sharing, but not in a streaming session. On the other hand, quality-aware peer selection can provide stable and predictable quality which is a pre-condition for video applications. We leverage this observation to design an incentive mechanism by providing quality-aware peer selection to the cooperative users and arbitrary peer selection to others.

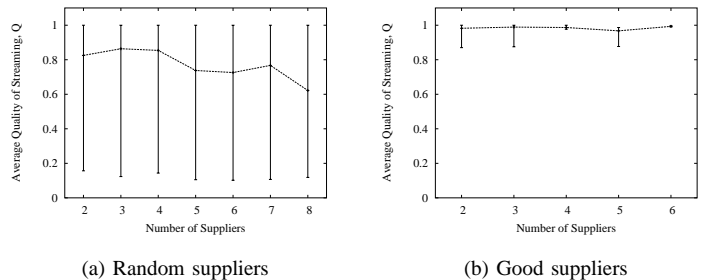(a) Random suppliers   (b) Good suppliers

Fig. 4. Quality experiencing by a receiver when suppliers are varied in Planet-Lab test-bed. (a) suppliers are chosen arbitrarily and the quality is highly unpredictable. (b) known and good peers are chosen as suppliers to provide high quality and predictable service.

We use the same setup in PlanetLab test bed as described in the previous section. Figure 4 shows that arbitrary peer can provide as high as $Q = 1$, however, the minimum value goes to less than 0.2 for each experiment. We vary the number of suppliers on the X-axis. We observe that higher number of suppliers might provide even worse quality because the receiver has to maintain more connections and some of the suppliers might share congestion in their paths to the receiver. Figure 4(b) shows that quality-aware peer selection really pays off. Not only the quality is high, but also the quality is
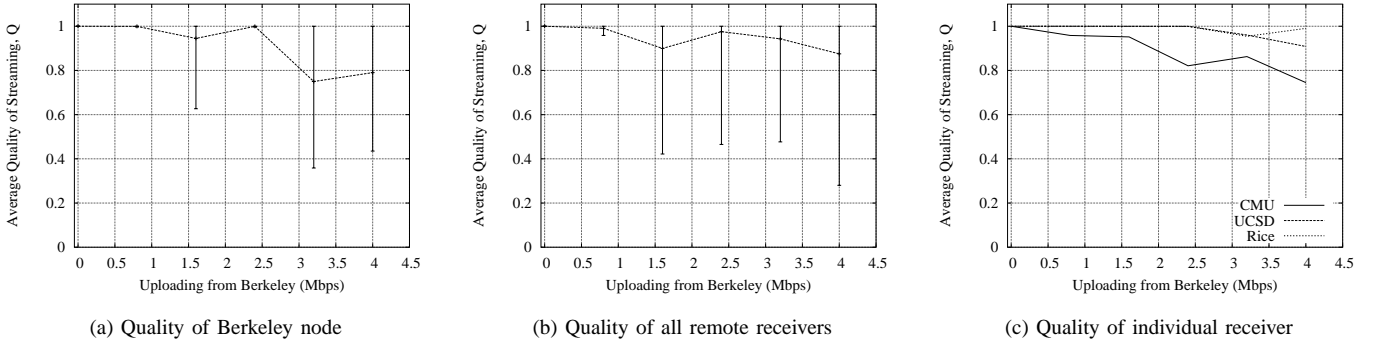
Fig. 3. Simultaneous uploading from a peer affects the quality of both local and remote peers in Planet-Lab setup. Downloading quality Q of the Berkeley node is poor when it supplies at a rate ≥ 3 Mbps. Average Q of the remote peers as well as the quality experienced by individual peers is poor when Berkeley supplies at a high rate. The figure shows average, minimum, and maximum values.

more stable in this case. Thus, a motivation to the users for sharing is that they will receive good and predictable quality of service in return.

III. PROPOSED INCENTIVE MECHANISM

P2P systems in general are characterized by large populations and asymmetries of interest, resulting in few, if any, repeat transactions between any given pair of nodes [5]. Therefore, a score-based incentive mechanism that encourages cooperation through indirect reciprocity [11] is suitable for a P2P media streaming system. We propose a service differentiated-based peer selection as an incentive mechanism for P2P media streaming.
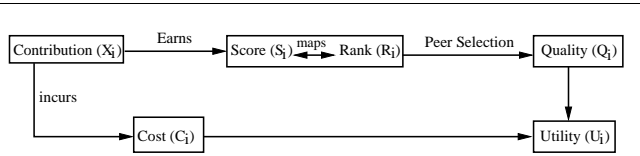

Fig. 5. Converting the contribution of a user into a score, which is mapped into a percentile rank among all users. The rank is used to select suppliers of a streaming session, which eventually determines the perceived streaming quality of a user. The utility of a user depends on the quality of service it receives and the cost of its contribution.

We consider a P2P media streaming system consisting of rational users who choose their contribution level in order to maximize their individual utility (Figure 5). The contribution level of a user is converted into a score, which in turn is mapped into a percentile rank, determining 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 selection scheme may allow a user to select peers with equal or lower rank to serve as suppliers. Thus, the peer selection takes into account the relative contributions by the users instead of their actual contributions. The outcome of the peer selection process is the realized quality of the streaming sessions.

The user contribution function X depends on its action a and a random output distortion θ. The output distortions are due to the approximation of contributions by sampling,

and are independently and identically distributed. Utility U is a function of the streaming session quality Q and the contribution cost C. We write the contribution function and the utility function for user i as follows:

$$X(a_i, \theta_i) = a_i + \theta_i, \qquad (2)$$

$$U(a_i) = \alpha Q(a_i) - \beta C(a_i), \qquad (3)$$

where α and β define the values of streaming quality and contribution cost to user i. The behavior of the overall system is defined by a social welfare function:

$$SWF = \sum_{i=1}^{N} U_i. \qquad (4)$$

In equation 3, quality can be expressed as a function of contribution, score, or rank. The quality function is system dependent, but should exhibit the following properties: (i) it is monotonically non-decreasing in user score, (ii) the quality asymptotically reaches to the value Q_{MAX}, which represents the highest possible quality provided by the system, (iii) the function has a non-negative initial value, i.e., Q_{BE} $Q(S_i=0)$ ≥ 0. We will consider quality functions later in this section.

To provide an analytical model of the service differentiated incentive mechanism, we first consider only two peers i and j and then generalize the model for N peers. Let p_i be the probability of getting the maximum quality Q_{MAX} by user i. Then user i's expected utility is

$$U_i = \alpha\{Q_{BE} + p_i(Q_{MAX} - Q_{BE})\} - \beta C(a_i). \qquad (5)$$

The probability that i receives better service than j is as follows:

$$\begin{aligned} p_i = Pr(X_i > X_j) &= Pr(a_i + \theta_i > a_j + \theta_j) \\ &= Pr(\zeta < a_i - a_j) = G(a_i - a_j), \end{aligned} \qquad (6)$$

where $\zeta = \theta_j - \theta_i$ and $G(\zeta)$ is the cumulative distribution function (cdf) of ζ. Thus, the probability is expressed in terms of the relative contributions by the users. The rank R_i ($0 \leq$

$R_i \leq 1$) can replace the probability in equation 5 to calculate the expected utility of user i because both probability and rank can be expressed using the cdf of user contributions. The rank calculation is shown in Section III-B.

To maximize the utility, we can take first order derivative of equation 5 and equate to zero i.e. $\partial U/\partial a_i = 0$. Thus, i's reaction function is

$$\alpha g(a_i - a_j)\{Q_{MAX} - Q_{BE}\} - \beta C_i' = 0, \qquad (7)$$

where $g(a_i - a_j) = \partial G(a_i - a_j)/\partial a_i$, $g(\zeta)$ is the probability density function (pdf) of ζ. Peer j acts similarly as Peer i. The equilibrium contribution levels (a_i^*, a_j^*) satisfy the best-reply requirements (for $i, j = 1, 2, i \neq j$)

$$a_i^* = \arg\max_{a_i} \alpha\{Q_{BE} + G(a_i - a_j^*)(Q_{MAX} - Q_{BE})\} - \beta C(a_i).$$
$$(8)$$

This incentive scheme has the same characteristics as rank-order tournaments. Lazear and Rosen [8] proposed the rank order tournaments as a competition among several workers who perform similar tasks for optimum labor contracts. O'Keeffee et al. [12] present conditions that must be met for fair contests in a multi-player format. Bognanno [13] examines the aspect of pay and promotion in corporate hierarchy using tournament theory. With empirical evidence the author shows that the winner's prize in CEO tournaments increases with the number of competitors for the position. The rank-order tournament theory is a viable choice for P2P media streaming because of the inherent nature of P2P media streaming application. The peer selection determines the quality of a streaming session, and it depends on relative contribution among the peers.

In rank-order tournaments, there are two levels of prizes; the top prize for the winner(s) and the other prize for the loser(s). In our mechanism, the utility of a peer varies from Q_{BE} to Q_{MAX}. The newcomers and free-riders receive Q_{BE} and the high rank users receive Q_{MAX}. A better than best effort service with quality Q, $Q_{BE} \leq Q \leq Q_{MAX}$, is provided to the users based on their level of contribution. To generalize equation 8 for N players, we extend the probability p_i of equation 5 from two users to N users, which is $p(a_1, a_2, \ldots, a_N)$.

When a peer first joins the system, it begins with a score of zero and receives best-effort service, i.e., $Q_{BE} = Q(S_i{=}0)$. The quality of this service may vary from system to system, and vary as a function of system load. For example, a supplier node may choose to serve a node with a lower score only when it is idle. Hence, best-effort service quality can be highly unpredictable. If a user wishes to receive better-than-best-effort streaming, it must earn a positive score by contributing to the system. A rational user will determine its optimal contribution level a_i^* to maximize its utility.

In our incentive mechanism, newcomers and free-riders are treated identically. This is to prevent *whitewashing* behavior [5], where free-riders repeatedly masquerade as newcomers by obtaining new identities at low or no cost prior to each request. This punishment imposed on the newcomers

is exactly the social cost of cheap pseudonyms as quantified in [14].

If the peer selection scheme allowing a user to select peers with equal or lower rank is used, and a large number of users join the system at the same time (with the same initial score of zero), it may be possible for a free-rider to obtain high quality service by selecting these fellow newcomers. However, this exploitation can be bounded in duration, beyond which the free-rider will have to contribute to continue receiving high quality streams. This is because if a peer does not contribute for a while, its score will fall behind others, and thus it will not receive any quality service from others.

We next discuss the various component stages of the mechanism. These include score and cost functions, which map contributions to scores and costs; percentile rank computation, which maps scores to ranks; and quality functions, which map rank to expected quality.

A. Scoring function and contribution cost

There is tremendous flexibility in the choice of a scoring function. A scoring function could consider only the contribution by a user, or both the contribution and consumption by a user. For example, if the amount of uploaded and downloaded bytes are B_{out} and B_{in} respectively, possible scoring functions might include: $S = B_{out}$, $S = B_{out}$ - B_{in}, $S = \frac{B_{out}}{B_{in}}$. KaZaA uses the third function in scoring their users. The scores might be subject to an aging factor to encourage contributions on a continual basis.

The scoring function could also take into account the difference in demand for different resources in the network. For example, if a user i supplies L_i blocks of a file at a data rate T_i for duration D_i, the user may receive a score of $L_i^\alpha T_i^\beta D_i^\gamma$, where α, β, and γ serve as market pricing signals to reflect the demand of the system.

Where appropriate, the scoring function could additionally take into account the amount of defection vis-a-vis the amount of cooperation by a user. Specifically, in a *Reward-Penalty* scheme, users are penalized for refusing to serve a request just as they are rewarded for serving one. This is in contrast to the baseline *Reward* scheme, where defections are not penalized. In general, defections are usually difficult or impossible to detect in P2P systems [5], so that the effectiveness of this type of scoring function may be limited.

Contribution cost incurred by node i for serving a streaming session can be expressed as a function of bandwidth and storage usage: $C_i = c_L L_i M + c_T T_i D_i$ where c_L is the unit storage cost, L_i is the number of blocks served by node i, M is the average block size in bytes, c_T is the unit transmission cost, T_i is the transmission rate of node i, and D_i is the transmission duration of node i. Hence, the total number of bytes uploaded is $B_{out,i} = L_i M = T_i D_i$.

For the evaluation section, we consider $a = B_{out}$, $S = a = B_{out}$, and $C = c_T a = c_T B_{out}$.

A common problem with any score, payment or reputation based scheme is that a malicious user can tamper the software and increase its own score. A user can collude with others to increase their scores without sharing anything. One solution

of the collusion problem is to compute the score subjectively, where one user weighs the opinion of another user based on how much they trust each other [5]. Another solution is to use a security infrastructure such as EigenTrust [15], where the global reputation of each peer i is given by the local trust values assigned to peer i by other peers, weighted by the global reputations of the assigning peers. This framework can be integrated to compute and maintain scores in our streaming system.

B. Percentile rank computation

A user's ability to select peers as its suppliers is determined by its score. For example, when a user with score S_i issues a request for a particular file, only nodes with scores less than or equal to S_i will respond to the request. We note that a peer with a score higher than S_i can still respond and be a supplier if it is selected, however, they are not bound to do it. Knowledge of one's score alone is not sufficient for predicting the expected quality to be received by the user. The score must be mapped into a percentile rank, based on the global distribution of scores, so that a user can determine whether the current score is sufficient for the user to obtain streaming service of an acceptance quality level. The rank represents the probability (equation 5) that a user's contribution is higher than others, and thus it should receive high quality service from the system.

To compute the percentile rank, we calculate the cdf of the scores. The score is a discrete variable and thus the pdf is defined only at the values where the score has a meaningful value. The cdf is defined as:

$$F(S) = \sum_{i=S_{low}}^{S_{high}} f(i), \qquad (9)$$

where f is the pdf of the scores. The cdf provides a relationship between the percentile rank and the score. The percentile is obtained by dividing the cdf with the total number of peers.

In the absence of a centralized entity that maintains global state (i.e., updated scores of all users), individual nodes can locally estimate their rank based on a sample of user scores, possibly via the passive monitoring of query and/or response messages containing score information exchanged over the network. This is an acceptable solution since percentile ranks are only used for prediction purposes, not for actual peer selection. Proposition 1 shows that the required sample size does not depend on the size of the network, but only on the variance of the population. For example, Figure 6 shows that a sample size of 50 or 100 produces a reasonable approximation of percentile rank in a network with population of 1200 users. If the approximation results in over-estimating or under-estimating the quality, the user needs to increase the sample size.

Proposition 1: Each peer requires $O(\sigma^2)$ samples to approximate its percentile rank, where σ is the standard deviation of the original population of scores.

Proof: Suppose the original population has mean μ and standard deviation σ. Let N be the number of samples taken from this population randomly, which is enough to
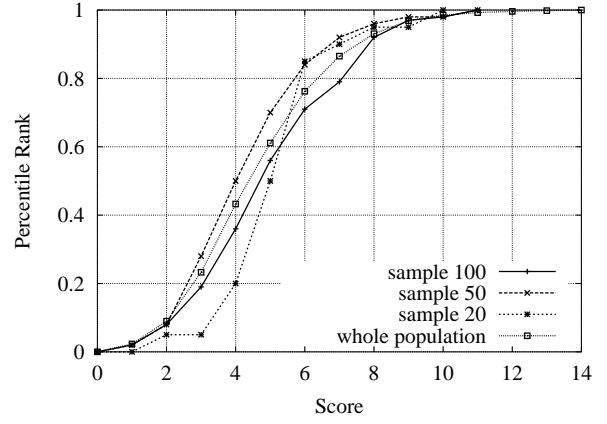

Fig. 6. Using random sampled data to estimate the percentile rank from a population of 1200 users. A sample size of 50 or 100 provides good approximation of the percentile rank, but not a sample size of 20.

approximate rank curve so that the new mean is close to μ within a small constant value a. Thus, $\left| \frac{\sum_{i=1}^{N} X_i}{N} - \mu \right| < a$

$\Rightarrow \left| \frac{\sum_{i=1}^{N} X_i - N\mu}{\sqrt{N}\sigma} \right| < \frac{a\sqrt{N}}{\sigma}$, for $\sigma > 0$

Let, $Z = \left| \frac{\sum_{i=1}^{N} X_i - N\mu}{\sqrt{N}\sigma} \right|$. For large N, Z has standard normal distribution (Central Limit Theorem). Let $b = \frac{a\sqrt{N}}{\sigma}$. If b is known, the $Pr[|Z| < b]$ can be obtained from the table of standard normal distribution. On the other hand, we can obtain the value of b for a specified probability. Using the table of standard normal distribution function, $Pr[|Z| < b] = 0.95$ if $b = 2$.

The approximated mean will be close to the real mean by a with probability greater than or equal to 0.95 if $2 \leq \frac{a\sqrt{N}}{\sigma} \Rightarrow N \geq \frac{4}{a^2}\sigma^2$.

Thus, the number of required samples to approximate the percentile rank is $O(\sigma^2)$. $\qquad \square$

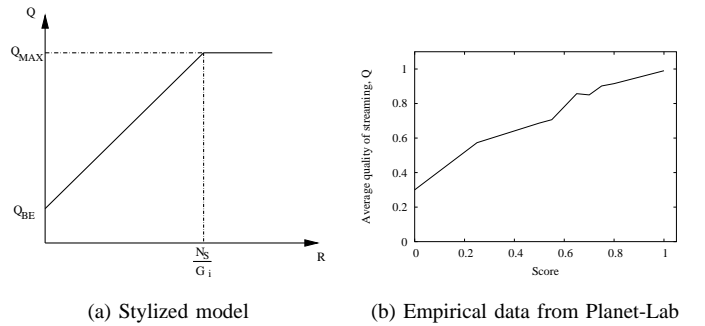(a) Stylized model (b) Empirical data from Planet-Lab

Fig. 7. Quality curves from the stylized system and the incentivized PROMISE implementation deployed on Planet-Lab. The wide-area measurements provide a good fit with (10) when $Q_{BE} = 0.3, Q_{MAX} = 1$, and $N_S = G_i$.

C. Quality function

As indicated before, the quality function, that is the mapping between rank and quality, depends on the design of the system, the characteristics of the nodes, the underlying topology, as well as the prevailing load conditions.

Nonetheless, the quality function should be well behaved (e.g., monotonicity, well defined bounds) so that users can predict the expected quality of streaming sessions based on their rank, and determine their optimal contribution level at which utility is maximized.

As an illustration, consider a stylized system with a peer selection scheme where session quality increases linearly with the number of *good* suppliers, such that a session with N_s good suppliers realize a quality of Q_{MAX}, and a session with zero good suppliers realize a quality of Q_{BE}. A node that possess characteristics such as high availability, topological proximity to the receiver i, etc., is a candidate to become a *good* supplier for i. Assume there are G_i such candidates for node i. However, node i can only select those nodes with a rank equal to or lower than R_i. Assuming independence between candidate *goodness* and rank, node i can only select up to $R_i G_i$ to be its *good* suppliers. Therefore, the quality function can be expressed as:

$$Q_i(R_i) = \begin{cases} Q_{MAX}, & R_i \geq \frac{N_s}{G_i} \\ \frac{R_i G_i}{N_s} Q_{MAX} + \left(1 - \frac{R_i G_i}{N_s}\right) Q_{BE}, & \text{otherwise.} \end{cases} \tag{10}$$

We plot in Figure 7 the quality functions using (10) as well as actual streaming quality measurements collected from the incentivized version of the PROMISE implementation deployed on the Planet-Lab test-bed. The wide-area measurements provide a good fit with (10) when $Q_{BE} = 0.3, Q_{MAX} = 1$, and $N_S = G_i$. The quality function for the PROMISE system exhibits a positive best-effort quality level for users with zero score, and quality increases monotonically with user score.

In summary, a rational user computes its score due to contribution and the cost of contribution. The score is mapped to a percentile rank that brings the quality of service for the user through peer selection. Thus, a utility maximizing user cooperates as long as the contribution brings positive utility (equation 3) for its streaming session, otherwise it defects.

IV. EVALUATION

We use simulation and wide area experiments over the Planet-Lab test-bed to evaluate our incentive model. First, we describe the setup of the experiments, and then we show peer selection and quality of service improvements that can be obtained with the proposed incentive mechanism. We note that with the incentive mechanism, the peer selection is done based on the scheme described in Section III, and without the incentive mechanism, the altruistic users act as suppliers for all streaming sessions. Furthermore, we show that without the incentive mechanism a streaming session is required to send more redundant data (to tolerate packet loss) to achieve the same QoS that the incentive mechanism can provide.

A. Setup

Simulation. We simulate the incentive mechanism using ns-2 [16]. The PROMISE [9] simulation module is extended to incorporate our incentive mechanism. The peers are connected to a network with a topology mimicking the Internet.

We use a hierarchical topology, where the highest level is composed of transit domains representing large Internet Service Providers (ISPs). These links have delays on the order of 100 ms, and the link capacity is varied from 1.5 Mbps to 5 Mbps using a uniform distribution to simulate the available bandwidth on these links. Higher bandwidth will require higher number concurrent sessions to introduce congestion in the network. On the second level stub domains (small ISPs, campus networks, and moderate-size enterprise networks) are attached to the transit domains. These links have delays on the order of 10 ms, and the link capacity has the same distribution as the transit domain. These two levels are generated using the Georgia Tech Internet Topology Mapping tool. At the lowest level, we probabilistically add the end hosts (peers) to stub routers. The topologies used in the experiments have, on average, 600 routers and 1200 peers. Each peer is connected to the stub routers with a link of 1.2 Mbps capacity and 10 ms delay. Only the end hosts participate in a streaming session either as a sender or receiver.

Peers' parameters (offered rate, availability) are chosen to reflect the heterogeneity in the P2P community. The offered rate is expressed as a fraction of the streaming rate. In our experiments, the target streaming rate is $T_0 = 1$ Mbps. The offered rate is varied from $\frac{T_0}{8}$ to $\frac{T_0}{2}$. No single peer is selected to supply at a full rate of a session. However, a user can contribute to multiple sessions simultaneously if it has enough resources to do so. The availability is varied from the range [0.1 - 0.9] using a uniform distribution. A peer with low availability has a higher chance to go offline during a streaming session. By cooperation, we mean a peer wants to share its resources with others. 50% cooperation means 50% of the total nodes have the willingness to share with others.

In each experiment, we run the model for $1000 - 2000$ rounds. In each round, we randomly pick a set of K peers as receivers for K concurrent sessions. For each session, we then look for potential suppliers as a candidate set, who have rank lower than or equal to the rank of the receivers and satisfy the requirements (rate, availability) mentioned by the receiver. The best-effort service is given to the new users by providing an arbitrary number of peers based on the availability of the peers who want to supply to a low ranked peer. A user with a higher rank obtains better candidate set that provides better quality streaming. In PROMISE, 10-20 peers are selected as a candidate set for each session. A utility maximizing peer may not choose to be a supplier for any session even if the request comes from a user with a higher rank. Upon successful streaming, the suppliers receive appropriate scores.

Planet-Lab. We use PROMISE implementation as an underlying streaming system in the Planet-Lab test-bed to conduct wide area experiments. We use eighteen nodes in the US, Europe, and Asia as senders and receivers. We build the underlying network topology and annotate the topology with link delays with traceroute, and measure available bandwidths on the paths with Pathload [17]. We use the proposed incentive mechanism by conducting quality-aware peer selection for the cooperative users to provide best possible quality, and

random peer selection to provide best-effort service.

B. User contribution & cost

We conduct experiments with the proposed incentive mechanism applied to the PROMISE streaming system to study the user behavior. The simulation is run for a long time (5000 streaming sessions) to ensure that each user gets several chances to be a supplier and receiver. On the average, a peer supplies in 20 sessions and receives 5 times in this experiment. All users start with the same score and the pdf of the contribution is a single vertical line. When the users value their costs and benefits similarly i.e., the coefficients of equation 5 are same for all users, the shape of the distribution curve is similar to the normal distribution. If we categorize the users in two groups in terms of their communications resources such as cable modem and DSL users, the user contribution is a bi-modal distribution. In both cases, the users contribute close to the average contribution of their groups.

We evaluate the incentive mechanism for a heterogeneous peer community that values the cost and benefits differently. We use three different groups (DSL, Cable modem, and Ethernet) and set different values of β in equation 5 for each group. For example, when $\beta_{DSL} = 8\beta_{Ethernet}$ the optimal level of contribution by DSL users is lower than the optimal level of contribution by the Ethernet users. When $\beta_{DSL} = 4\beta_{Ethernet}$, the difference of contribution among DSL and Ethernet users is small, assuming that the Ethernet users always contribute at a higher rate than the DSL users. Thus, users with a high cost function receive relatively less benefits from the system. To increase the utility, the users have to contribute more to the system.

C. Peer selection

There are several factors that takes into account during peer selection such as the availability of the peer, the offered rate, and the goodness of the path from the suppliers to the receiver. Altogether, the factors determine the goodness of a peer. We conduct experiments how these factors affect in selecting peers during streaming sessions. If a peer is selected more often, it will earn high scores and thus will receive high quality of service. On the other hand, if a peer is not selected at all, it will not be able to obtain a high score as well as high quality service from the system. Figure 8 shows that higher availability increases the likelihood of a peer to be selected. Low availability does not guarantee to earn a high score, i.e., a peer with low availability can obtain a high score but the variance is very high.

Path goodness—ranging in [0-1]—represents how good a path is in terms of available bandwidth and loss rate of the path. The path goodness also depends whether the path is shared among multiple suppliers. Good path is likely to be selected as it is shown in Figure 9. The path selection frequency increases with the path goodness until path goodness approaches to 0.70. After that the path selection frequency decreases with path goodness. This is because a few paths exist with this high path goodness. The path

goodness depends on the position of the receiver, and as we select the receivers randomly, the paths with high goodness exist only in a few sessions. This experiment shows that a good path is likely to be selected in a streaming session. However, if the peers on the path have low availability or offered rate, the path might not be selected.
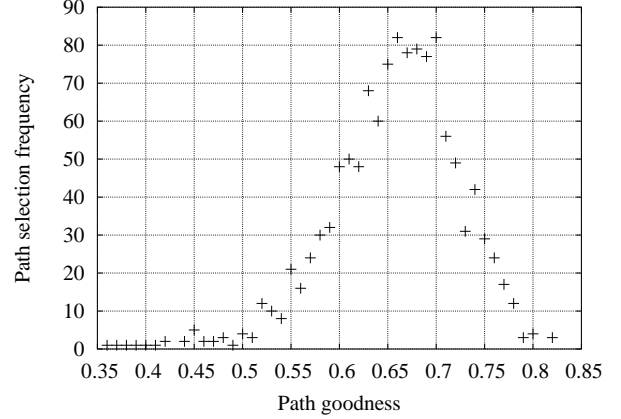
Fig. 9. Goodness of path determines the peer selection frequency. A good path is likely to be selected in a streaming session. Paths with goodness greater than 0.70 have low selection frequency because the high quality paths exist in a few sessions.

Figure 10 shows that if the peer goodness is high, it is highly likely that the peer will be selected, and thus the score of the peer will be high. However, it is not necessary to be the best peer to obtain a high score. A peer with goodness 0.3 can get score as high as the score obtained by a peer with goodness 0.7. If the goodness of the peer is very low, the score has a high variance. Thus, a peer with a low goodness does not always obtain a high score as well as high quality of service from the system. This experiment encourages the peers to contribute more to the system to obtain high quality of service from the system.
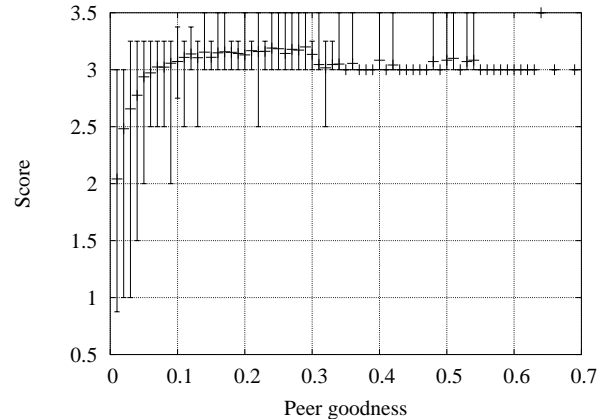
Fig. 10. The score of a peer depends how good the peer is based on its resources and the goodness of the path. A peer is not required to have best resources to obtain a high score.

The peer goodness—ranging in [0-1]—depends on peer resources and path goodness.

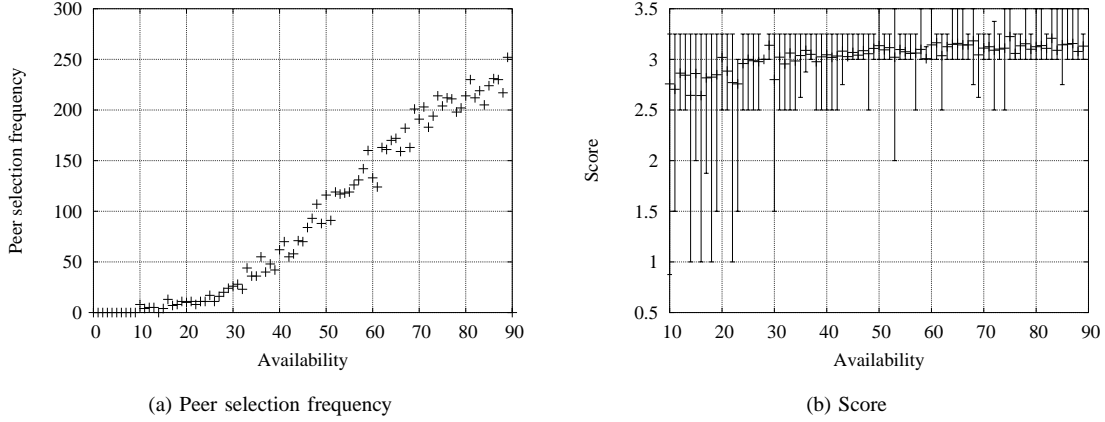(a) Peer selection frequency

(b) Score

Fig. 8. User selection frequency and score depend on the availability of the peers. Higher availability leads to higher number of peer selection, which helps in getting higher score whereas low availability does not guarantee to earn a high score.

D. Quality of service

The incentive mechanism provides flexibility to select suppliers to the cooperative users to improve the streaming quality. We design experiments to evaluate the quality a user can experience with or without the incentive mechanism. With the incentive mechanism, the system selects best suppliers for each session and ensures that each supplier has high availability so that the suppliers do not fail often. Thus, these sessions experience fewer switching (replacing the failed suppliers) and the video session is smooth. We report the social welfare function (equation 4) in terms of rate, loss, and quality of service for these experiments.

First, we measure the expected rate at the receiver. This rate is the total bytes coming from all senders. Figure 11a shows that the expected rate is low when there is no incentive mechanism and not all peers are altruistic (100% cooperation). With the incentive mechanism, the expected rate is very close to the maximum achievable rate with full cooperation among the users. Without the incentive mechanism, the percentage of altruistic users is set to 1% and 10% because it is observed that less than 7% users supply high volume of data (more than 50% data) in popular P2P system [2], and our incentive mechanism requires around 10% cooperation from the peers. If there is no cooperation, the expected rate is zero.

The network load has a big impact on the quality of service with or without the incentive mechanism. For example, when the network is idle, the rate is almost the same with or without the incentive mechanism. Thus, the gain from an incentive mechanism is not significant. The difference between the rates increases when the number of sessions increases. For K=20 sessions, the rate is 1.1 Mbps with the incentive mechanism and 0.80 Mbps without the incentive mechanism (10% cooperation). However, after $K > 20$, the difference of the rates decreases because the bottleneck is shifted from the hosts to the network.

Figure 11b shows that the network experiences as high as 33% packet loss for $K = 20$ without the incentive mechanism (10% cooperation). With the incentive mechanism, the loss

ratio is only 10% for the same number of sessions. The loss ratio curve follows the same pattern to the rate curve, i.e., the incentive mechanism is more useful when the network is not idle or heavily congested.
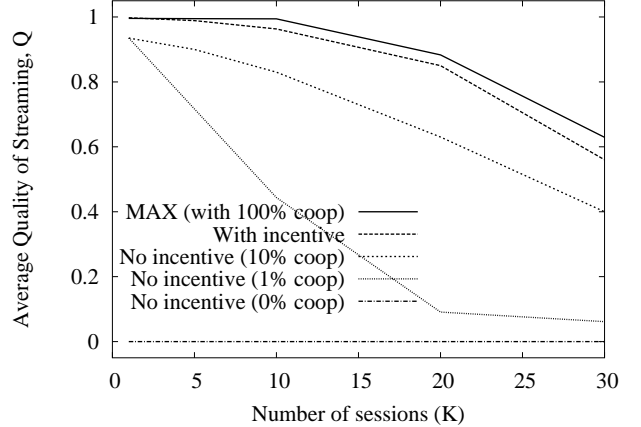

Fig. 12. Quality of service with or without incentive mechanism in ns-2 simulation. Without incentive mechanism, a few users cooperate and the stream quality is poor.

Now, we analyze the perceived quality by the receiver using the metric defined in (1). Here, packets that miss their play-out deadlines are considered as lost. Figure 12 shows that the system quality Q is close to 1 for less than 10 concurrent sessions if the incentive mechanism is used. If there is no incentive, the altruistic users cannot maintain the the system quality to 1 for the same number of concurrent sessions. The quality drops to 0.83 for 10% cooperation and 0.44 for 1% cooperation when $K = 10$. For $K = 20$, the incentive mechanism provides $Q = .85$ whereas without the incentive mechanism the quality goes down to 0.63 when 10% users are altruistic and 0.09 when 1% users are altruistic. The quality deteriorates drastically for 1% altruistic users because a fraction of packets still reach the receiver but the packets are not enough to decode a whole block. The figure also shows that with the incentive mechanism the quality is
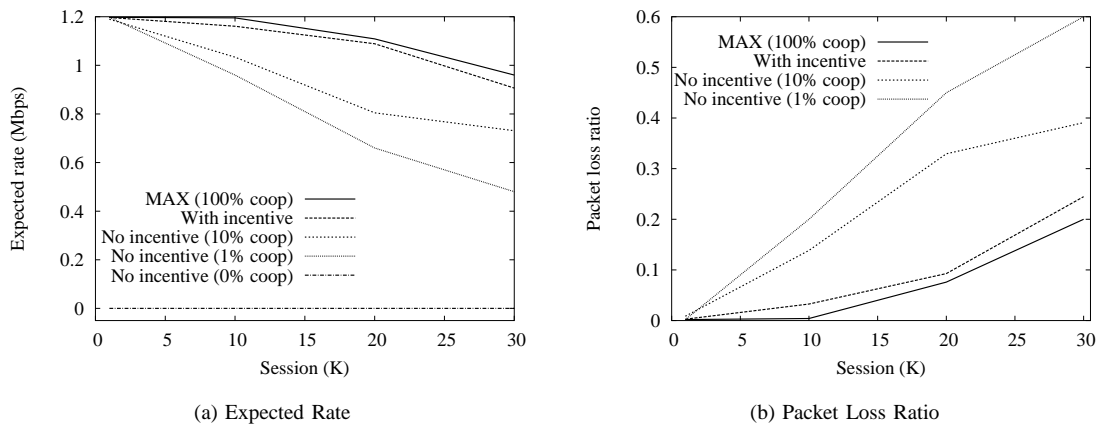
(a) Expected Rate

(b) Packet Loss Ratio

Fig. 11. Expected rate and packet loss ratio to the receiver in ns-2 simulation. The rate is the total bytes received irrespective of their usefulness. The rate is high and loss ratio is low if the proposed incentive mechanism is used to provide service differentiation.

very close to the maximum quality achievable with 100% cooperation. Moreover, with the incentive mechanism the load is distributed among all users because everybody acts as a supplier when necessary, however, this cannot be achieved without the incentive mechanism.

We conduct experiments for both the *Reward* and *Reward-Penalty* schemes. In both cases, the quality of streaming does not differ significantly. There is a cost associated with penalizing users for defection because it is not trivial to identify the defectors and penalizing users requires more accountability. Thus, we prefer the *Reward* scheme over *Reward-Penalty*.

If the network load is increased, the streaming quality deteriorates with or without the incentive mechanism. For higher number of sessions, the quality is low for both cases because the network is extremely congested. The cooperation by users does not improve the system performance. Thus, the incentive mechanism is not necessary when the network is idle and not effective when the network is heavily congested.

In our Planet-Lab experiments, we observe that the *quality-aware* peer selection provided by the incentive mechanism ensures high quality streaming. The value of Q is close to 1 with a very low variance (Table I). Whereas, in absence of the incentive mechanism, the system selects suppliers randomly and the average quality varies from 0.1 to 1 with an average value of 0.72. Therefore, suppliers selection for a streaming session really pays off, and it motivates the users to share so that they can receive high quality streaming.

E. Incentives and FEC overhead

In traditional streaming systems, data redundancy is used to tolerate packet loss and provide good quality streaming. We show that the proposed incentive mechanism can even reduce data redundancy in a streaming session. A streaming system usually uses forward error correction (FEC) coding, where n packets are sent instead of $k, k < n$ per block, where a block is defined by the streaming application as a unit of data transmission among the senders to a receiver. Any k out of n packets can reconstruct the block. Thus, the streaming

session can tolerate up to $(n - k)$ packets loss. The FEC overhead is defined as:

$$\text{FEC overhead} = \frac{n - k}{k} \qquad (11)$$

We use video traces of two movies (*Star Wars IV* and *From Dusk Till Dawn*) encoded using MPEG-4. The video traces have the information of frame number, frame type (I, P, or B), frame play-out time, and frame length in bytes. We stream the first 5 minutes of each movie, and both movies have a frame rate of 25 frames per second. For each streaming session, we record the arrival time of each single packet. Then, we determine the number of frames that would have missed their deadlines. We calculate the number of blocks that can not be decoded because fewer than k packets arrived. In this experiment, we vary the overhead due to FEC to tolerate packet loss.

Figure 13 shows the number of undecoded blocks for *Star Wars IV*. It shows that 32 blocks of this movie cannot be decoded if FEC or incentive mechanism is not used. Each block has approximately one second play-out time data. Thus, 32 seconds out of 5 minutes will be wasted. However, with the incentive mechanism only four blocks are undecodable because of the quality-aware supplier selection.

In the absence of FEC, introduction of the incentive mechanism reduces the number of undecoded blocks from 32 to 4, which is a significant reduction. Without the incentive mechanism, a FEC overhead of 35% would be necessary to achieve the same improvement of quality that the incentive mechanism can provide. With the incentive mechanism, only 10% FEC overhead is necessary to have zero undecoded block at the receiver, whereas 40% FEC overhead is necessary to achieve the same goal without the incentive mechanism. Experiments using the movie *From Dusk Till Dawn* produce similar results. These experiments show that if no incentive mechanism is used, a streaming session has to send more redundant data to achieve the same quality that the proposed incentive mechanism provides.

TABLE I

STREAMING QUALITY (Q) IN WIDE AREA EXPERIMENTS OVER PLANET-LAB TEST-BED.

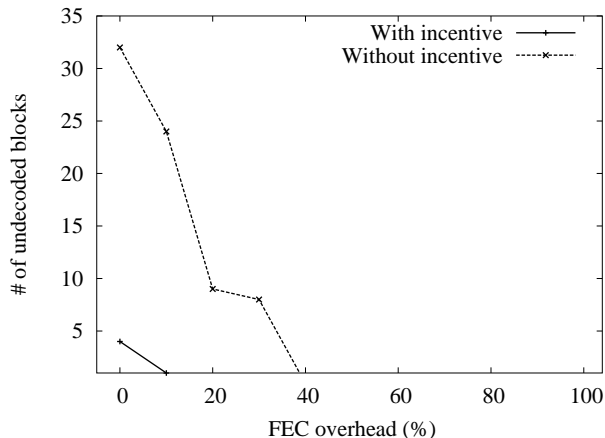|	Minimum	Average	Maximum	Variance	Standard Deviation
Incentive Mechanism	0.87	0.988	1	0.00044	0.0211
No Incentive Mechanism	0.101	0.718	1	0.10002	0.3162

Fig. 13. Number of undecoded blocks comparing to the level of redundant data due to FEC. The result is shown for the movie *Star Wars IV* in Planet-Lab setup. In the absence of FEC, introduction of the incentive mechanism reduces the number of undecoded blocks from 32 to 4.

V. RELATED WORK

There are several studies in the literature to investigate incentive mechanisms for the Internet applications and P2P systems. We study incentive mechanisms for P2P file sharing and show why a new mechanism is necessary for media streaming.

File Sharing (Payment, Reputation, and Score). Golle et al. [3] propose a micro-payment mechanism, where each user can earn rewards if they upload to other users. The rewards can be used for future download. Using a game theoretic model, the authors analyze the equilibria of user strategies under several payment mechanisms and conclude that there exists equilibrium for the micro-payment based system. The objective of this system is to achieve maximum cooperation from the users. Mojo [18], KARMA [19] and the light weight currency paradigm [20] are examples of payment-based (token-based) mechanisms. KARMA uses a single currency as a way of secure trading, and the light weight currency paradigm allows the users to trade any resource with their own currencies. Any entity can introduce its own currency as long as it is acceptable to other users in the system. In a reputation based system [21], the users earn reputation by sharing and the reputation determines peer quality. Downloading from a user with a high reputation has a higher probability to obtain better service. A score-based system [11] may allow a user to download multiple times from other users having lower scores than its own. KaZaA—a score-based P2P system—provides downloading priority to the users with high scores over the users with low scores. Our proposed incentive mechanism relies on scores to determine the user contribution, however, we map the score

to a percentile rank, which is used in determining the quality of service a user can obtain.

Probabilistic differential service. Buragohain et al. [7] propose a game theoretic framework to provide incentives in a P2P system. In this model, the peer contribution is expressed in terms of disk space shared per unit time. This contribution allows a peer to obtain differential service, i.e., more contribution to the system will earn a higher probability with which its request will be served by others. If the contribution is small, its request is more likely to be rejected. The incentive mechanism eliminates the free riders and increase the overall availability of the system. The authors show that the system reaches Nash equilibrium even with peers having different contribution and benefit parameters. The incentive framework increases the availability of the resources of the system, however, this does not provide service differentiation for media streaming, where quality of the peers is more important than having files available in streaming applications. We provide a framework where peer contribution provides higher flexibility to select good quality suppliers for better quality of service.

Synchronous vs. asynchronous. BitTorrent [22] provides incentives to the users to download files if they allow simultaneous upload by other users. This way the server redistributes the uploading cost to the downloaders, and a file can be served concurrently to a large number of users. BitTorrent does not need any score, token, or reputation computation, and therefore has the advantage of simplicity. However, the synchronous upload and download makes the model less suitable for media streaming because high quality peer selection for all receivers simultaneously is extremely difficult. For example, each downloading peer i has to look for peers who can be a potential uploader. At the same time i has to meet all the quality requirements (availability, offered rate, and the characteristics of the network paths) of peer selection for the uploaders. There might be users in the system who want to download, however, i might not be a good supplier for their streaming sessions.

Instead of using BitTorrent like *bit-for-bit* model, Chu et al. [23] propose a taxation model is proposed for multicast streaming, where peers with more resources contribute more bandwidth to the system. The peer with limited resources are subsidized by the system. The model is applicable in the streaming context because the publisher of the video stream has the means to enforce taxation on peers and the will to maximize their collective social welfare. SplitStream [24] and CoopNet [25] utilize the cooperation among the users to distribute high bandwidth content in a P2P system. In SplitStream, each user joins several multicast trees so that it can receive and forward content from/to other users. This way the forwarding load is distributed among all participating users.

The cooperative mechanism or the taxation model is useful in a multicast environment where a large number of users is interested in a streaming session, and the users want to cooperate synchronously with each other to receive the content. The synchronous model is not suitable for video-on-demand type of applications or streaming rare objects or less popular data. In this paper, our goal is to design an incentive mechanism that is applicable in an asynchronous manner, where the contribution and reward do not need to happen at the same time.

VI. Conclusion

Our study shows that there are multiple motivating factors for having an incentive mechanism in a P2P media streaming system. First, the streaming quality is poor if the level of cooperation is low even when the network is not heavily congested. Second, unlike traditional file sharing, cooperation from a few altruistic users cannot provide high quality streaming to its users in a large system.

We show that a rank order-based incentive mechanism achieves cooperation through service differentiation. In this framework, the contribution of a user is converted into a score, then the score is mapped into a rank, and the rank provides flexibility in peer selection that determines the quality of a streaming session. Cooperative users earn higher rank by contributing their resources to others, and eventually receive high quality streaming. Free riders have limited choice in peer selection, hence receive low quality streaming. The rank is estimated in a scalable way without involving all users in the system. Our experimental evaluation shows that the incentive mechanism provides near optimal quality ($Q \approx 1$) to the cooperative users until the bottleneck shifts from the hosts to the network. The incentive mechanism reduces the data redundancy required during a streaming session to tolerate packet loss. Without the incentive mechanism, it is required to send more redundant data to achieve the same QoS that the incentive mechanism can provide.

In our analysis, we assume that the users of a P2P system behave like the users of a file sharing system. There is no empirical data about the users behavior in a streaming environment. A future direction of this research is to deploy a streaming system to obtain empirical data, which can be used to validate and refine incentive mechanisms for a streaming system.

Acknowledgment

References

[1] E. Adar and B. Huberman, "Free riding on gnutella," *First Monday*, vol. 5, no. 10, Oct. 2000.
[2] S. Saroiu, K. P. Gummadi, and S. D. Gribble, "Measuring and analyzing the characteristics of napster and gnutella hosts," *Multimedia System*, vol. 9, no. 2, pp. 170–184, 2003.
[3] P. Golle, K. Leyton-Brown, and I. Mironov, "Incentives for sharing in peer-to-peer networks," in *proceedings ACM Electronic Commerce (EC '01)*, Tampa, Florida, Oct. 2001.
[4] R. T. B. Ma, S. C. M. Lee, J. C. S. Lui, and D. K. Y. Yau, "A game theoretic approach to provide incentive service differentiation in P2P networks," in *Proceedings ACM SIGMETRICS*, New York, New York, June 2004.
[5] M. Feldman, K. Lai, I. Stoica, and J. Chuang, "Scalable and robust incentive techniques for P2P networks," in *proceedings ACM Conference on Electronic Commerce (EC '04)*, New York, New York, May 2004.
[6] K. Ranganathan, M. Ripeanu, A. Sarin, and I. Foster, "To share or not to share: An analysis of incentives to contribute in collaborative file sharing environments," in *Workshop on Economics of Peer-to-Peer Systems*, Berkeley, California, June 2003.
[7] C. Buragohain, D. Agrawal, and S. Suri, "A game theoretic framework for incentives in P2P systems," in *proceedings P2P*, Sweden, Sept. 2003.
[8] E. P. Lazear and S. Rosen, "Rank-order tournaments as optimum labor contracts," *Journal of Political Economy*, vol. 89, no. 51, pp. 841–864, 1981.
[9] M. Hefeeda, A. Habib, B. Botev, D. Xu, and B. Bhargava, "PROMISE: Peer-to-peer media streaming using CollectCast," in *Proceedings ACM Multimedia '03*, Berkeley, California, Nov. 2003.
[10] "Planet-lab test-bed," http://www.planet-lab.org/, 2004.
[11] M. Nowak and K. Sigmund, "Evolution of indirect reciprocity by image scoring," *Nature*, vol. 393, pp. 573–577, 1998.
[12] M. O'Keefee, W. K. Viscusi, and R. J. Zeckhauser, "Economic contests: Comparative reward schemes," *Journal of Labor Economics*, vol. 2, no. 1, pp. 27–56, Jan., 1984.
[13] M. L. Bognanno, "Corporate tournaments," *Journal of Labor Economics*, vol. 19, no. 2, pp. 290–315, April, 2001.
[14] E. Friedman and P. Resnick, "The social cost of cheap pseudonyms," *Journal of Economics and Management Strategy*, vol. 10, no. 2, pp. 173–199, 2001.
[15] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina, "The eigentrust algorithm for reputation management in p2p networks," in *proceedings International World Wide Web Conference*, Budapest, Hungary, May. 2003.
[16] S. McCanne and S. Floyd, "Network simulator ns-2," http://www.isi.edu/nsnam/ns/, 1997.
[17] M. Jain and C. Dovrolis, "End-to-end available bandwidth: measurement methodology, dynamics, and relation with TCP throughput," *IEEE/ACM Transaction on Networking*, vol. 11, no. 4, pp. 537–549, 2003.
[18] B. Wilcox-O'Hearn, "Experiences deploying a large-scale emergent network," in *Proceedings 1st International Workshop on Peer-to-Peer Systems*, March March 2002.
[19] V. Vishnumurthy, S. Chandrakumar, and E. Gun Sirer, "KARMA: A secure economic framework for P2P resource sharing," in *Workshop on Economics of Peer-to-Peer Systems*, Berkeley, California, June 2003.
[20] D. Turner and K. Ross, "The lightweight currency protocol," Internet Draft, draft-turner-lcp-00.txt, Sept. 2003.
[21] M. Gupta, P. Judge, and M. Ammar, "A reputation system for peer-to-peer networks," in *proceedings ACM NOSSDAV*, California, June 2003.
[22] B. Cohen, "Incentives build robustness in BitTorrent," in *Workshop on Economics of P2P Systems*, Berkeley, California, June 2003.
[23] Y.-H. Chu, J. Chuang, and H. Zhang, "A case for taxation in peer-to-peer streaming broadcast," in *Proceedings ACM SIGCOMM'04 Workshop on Practice and Theory of Incentives in Networked Systems (PINS)*, Portland, Oregon, Aug. 2004.
[24] M. Castro, P. Druschel, A.-M. Kermarrec, A. Nandi, A. Rowstron, and A. Singh, "Splitstream: High-bandwidth multicast in a cooperative environment," in *ACM Symposium on Operating Systems Principles (SOSP '03)*, Bolton Landing, New York, Oct. 2003.
[25] V. Padmanabhan, H. J. Wang, and P. Chou, "Resilient peer-to-peer streaming," in *Proceedings IEEE International Conference on Network Protocols (ICNP '03)*, Nov. 2003.