

A Resource-trading Mechanism for Efficient Distribution of Large-volume Contents on Peer-to-Peer Networks

Simon G. M. Koo[†], C. S. George Lee[‡], Karthik Kannan^{*}

[†]Department of Mathematics and Computer Science, University of San Diego, CA 92110

[‡]School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907

^{*}Krannet School of Management, Purdue University, West Lafayette, IN 47907

email: simonkoo@ieee.org, csgee@purdue.edu, kkarthik@mgmt.purdue.edu

Abstract

In recent years, the rapid growth of peer-to-peer (P2P) networks has provided a new paradigm for content distribution. To improve the efficiency of a P2P system, it is important to provide incentives for the peers to participate and contribute their resources. Various attempts have been made to reward/penalize peers by providing service differentiation based on a requesting peer's history or reputation. However, in a truly distributed, non-cooperative environment, maintaining and preventing the untruthful revealing of such information within the community impose larger computation and communication overheads to the system. These problems are further magnified when large-volume contents are being distributed because of the length distribution processes and the update of history or reputation has to keep up with the distribution process. In this paper, we address the incentive provisioning problem for distribution of large-volume content in P2P networks, and present a "seeing-is-believing" incentive-compatible mechanism (protocol) in which a peer will decide how much resources will be assigned to which neighbors based on what it has experienced. The protocol applies a utility-based resource-trading concept where peers will maximize their contributions for a fair or better return, and we show that by adopting this protocol, the system will achieve Cournot Equilibrium. Furthermore, our protocol is light-weight, completely decentralized, and cheat-proof. Experimental results illustrate significant improvements on the distribution efficiency of our protocol over other adopted alternatives.

I. INTRODUCTION

The foundation of peer-to-peer (P2P) systems is based on the contribution of resources from peers and such contribution plays a significant role in the efficiency and performance of the system. A lot of efforts have been made to avoid *free-riding* and *tragedy of the commons* (Hardin, 1968), two of the major problems in P2P networks. Various designs have been proposed to provide incentives for peers so that they are more willing to share their own resources with the community, and resources of interest may include processing power, disk space, files, and bandwidth.

Peer-to-peer content distribution systems (PCDS) such as BitTorrent (Cohen, 2003) and eDonkey (2000) are among those whose performance depends highly on peers' bandwidth contributions. Recent survey shows that PCDS are taking over and gaining popularity over file-sharing systems like Kazaa (Borland, 2004). While both systems share and distribute contents in a P2P fashion, file-sharing systems emphasize more on effective content searching and locating, and PCDS focus more on distribution efficiency. This separation makes PCDS a better candidate for sharing/distributing large-volume contents when the overhead of locating content is negligible relative to the distribution process. PCDS also provide a low-cost alternative to Content Distribution Networks (CDN) (Dilley et al., 2002) for content providers since CDN could incur a high cost of operation when many customers want the content. Currently, Linux distributors like Debian and Mandrake use PCDS to distribute their version of Linux.

However, the problem of providing incentives for peers to contribute is not an easy one because it is very difficult to monitor the activities of each and every peer in the community and account for their contributions to the community. Attempts have been made to reward/penalize peers in P2P. Nevertheless, in a truly distributed, non-cooperative environment, maintaining and preventing the untruthful revealing of such information within the community have resulted in a costly computation and communication overhead. These problems are further magnified when the contents being distributed are large in volume because the distribution process lasts relatively longer and the update of history/reputation has to keep up with the distribution process. Furthermore, with distribution efficiency being the primary objective, it is not sufficient just to

measure a peer's contribution by how much data one has sent but also the rate one transmits data. Distribution efficiency is also affected by network condition and peers' connectivities so both should be considered when evaluating contributions. Most existing mechanisms are designed only to handle the file-sharing scenarios and do not take network traffic and data rate into consideration.

In this paper, we address the incentive provisioning problem for distribution of large-volume content in PCDS, and present a "seeing is believing" mechanism (protocol) in which a peer will decide how much capacity will be assigned to which neighbors based on what it has experienced. Our protocol applies a utility-based Cournot adjustment process where peers will maximize their contributions for a fair or better return, and we show that by adopting this utility-based protocol, the system will achieve Cournot Equilibrium (Luenberger, 1997). Nonetheless, our protocol enhances distribution efficiency, and because it does not rely on any reputation or history exchange, the overhead it imposes on the system is minimal. Experimental results confirm that our protocol performs better than widely-adopt counterparts in terms of user satisfaction and distribution efficiency,

The rest of the paper is organized as follows: We will review some of the related works in Section II, then we will present our resource-trading protocol in Section III. Incentive-compatible consideration on the protocol is discussed in Section IV. Experiments to evaluate the performance of our proposed protocol is presented in Section V. A discussion on the strength and limitation of our protocol is presented in Section VI. We summarize the conclusions in Section VII.

II. RELATED WORKS

Several works have been dedicated to the impact of peers behavior on the network performance, in particular, freeloaders and flash crowds. Ge *et al.* (2003) performed in-depth modeling and analysis and investigate the effect of freeloaders on peer-to-peer file-sharing systems. They showed that peer-to-peer file-sharing systems can tolerate a significant number of freeloaders without suffering much performance degradation. In many cases, freeloaders can benefit from the available spare capacity of peer-peer systems and virtually increase overall system throughput.

Stavrou *et al.* (2002) considered the problem of “flash crowd”, and they have proposed a lightweight protocol to handle these abnormal, and yet frequently occurred, client behaviors. Both works considered the *sharing* scenario, where contents are provided by the peers. They have not addressed the situation where contents are being *distributed* from a content provider instead of sharing from peers.

The design of an incentive-compatible mechanism is an important factor that determines the distribution efficiency. The “tit-for-tat” mechanism presented in the design of BitTorrent (Cohen, 2003) provides an incentive-provisioning scheme to dissuade freeloaders. However, the problem of providing incentives for peers to contribute is not an easy one because it is very difficult to monitor the activities of each and every peer in the community and account for their contributions to the community. Many attempts have been made to reward/penalize peers in peer-to-peer networks by providing service differentiation based on a requesting peer’s contribution history or reputation (Buehgger and le Boudec, 2004; Kamvar *et al.*, 2003; Kung and Wu, 2003; Ma *et al.*, 2004; Moreton and Twigg, 2003; Tamilmani *et al.*, 2004). In a distributed and non-cooperative environment, however, maintaining and preventing the untruthful revealing of such information within the community have resulted in a costly computation and communication overhead.

Another approach to provide incentive is through micro-payment. Once again, many new problems including account management, transaction security, and user credibility, etc. are associated with the mechanism, and Odlyzko (2003) provided a strong argument against the use of micropayments. The solution we propose is a better one in the way that peers make their own decisions based on what they experienced. As a result, all the problems that are associated with reputation exchange and security are removed.

III. RESOURCE-TRADING PROTOCOL

In this section we present our resource-trading protocol in PDCS. We consider the situation where a large file that is being distributed is cut into smaller pieces and peers who are downloading these pieces are also concurrently uploading to the community with the pieces

they already possess. This scenario has been widely adopted by many PDCS and is proven to be beneficial to the distribution efficiency (Qiu and Srikant, 2004; Yang and de Veciana, 2004). We assume that peers have gone through the neighbor-selection process (Koo et al., 2004) and have already identified a set of neighbors who are downloading the same content. The mechanism to exchange information about which content pieces each peer has is also assumed (see Cohen (2003) for an example of such mechanism). The problem is to determine how much capacity a peer is going to assign to each of its neighbors if the peer has something to contribute. For peers who are new to the system (have nothing to contribute) or unable to contribute because of content availability (their neighbors already have what they have), there is no decision to be made and they will just wait until they receive something from a neighbor that is interested by the others. Furthermore, we assume all peers are rational and selfish, and their primary objective is to receive the complete content in the shortest period, so they are willing to contribute up to their limit for maximum downloading capacity.

A. Notations and Definitions

We consider the following: for a peer s , it has a set of N neighbors. We denote the capacity that s receives from neighbor $i = \{1, 2, \dots, N\}$ at time t to be $\lambda_i(t)$ and the capacity that s assigns to neighbor i at time t to be $\mu_i(t)$. Peer s has a maximum in-bound and out-bound capacity of λ_{\max} and μ_{\max} , respectively. These can also be the declared limits of a peer instead of the physical limits, but we will show later that there are incentives for peers to reveal this truthfully. For ease of discussion, we will assume the uplink and downlink are separate, which means that peer s can receive up to λ_{\max} and send up to μ_{\max} at the same time.

For elastic applications like file transfer in PDCS, the general form of utility function of capacity is *strictly concave* because of a diminishing marginal rate of performance enhancement (Skenker, 1995). We define for peer s the utility function of capacity received from neighbor i as:

$$U_s(\lambda_i(t)) = \lambda_i(t)^{(1-\omega_s)} \frac{\mu_{\max}}{\lambda_{\max}^{(1-\omega_s)}} \quad (1)$$

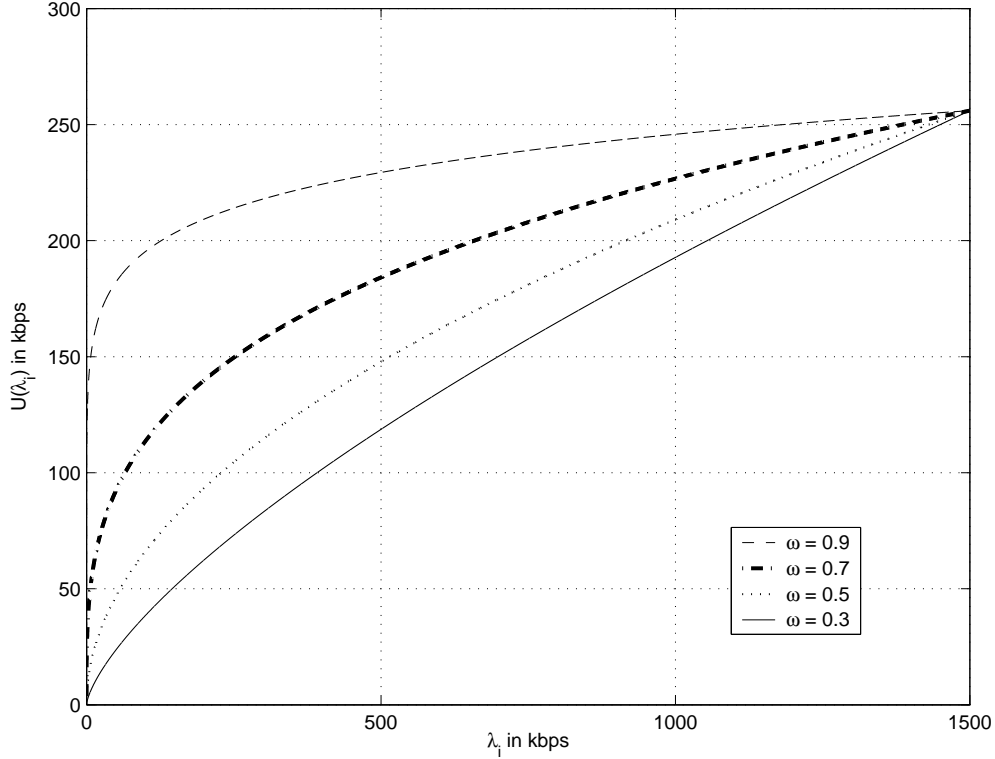


Fig. 1. Effect of ω on $U(\lambda_i)$. In this example $\lambda_{\max} = 1.5Mbps$ and $\mu_{\max} = 256kbps$

where ω_s is the *willingness-to-pay* of peer s and $0 < \omega_s < 1$. A larger ω means that a peer is willing to contribute more for the *same amount of capacity it received* from its neighbors. Figure 1 shows how the function changes with different ω 's. The unit of the utility function is defined to be in contribution capacity. Therefore, from the perspective of peer s , a desirable trade requires

$$\mu_i(t) \leq U_s(\lambda_i(t)) \quad (2)$$

We take this definition of utility function for the following reasons: (a) it is strictly concave, which agrees with Skenker (1995); (b) it can accommodate different levels of willingness-to-pay, which is a user-dependent parameter; (c) it eliminates the need to use different utility functions for received and contributed capacity; and (d) it is upper bounded by μ_{\max} when $\lambda_i(t) = \lambda_{\max}$,

which also concurs with the primary goal, where a peer is willing to contribute up to its limit to maximize the downloading rate.

It should be noted that our protocol does not require the utility function to be defined as Eq. (1) and it will work for all strictly concave utility functions that is bounded between 0 and λ_{\max} . We will use Eq. (1) as the utility function in the remaining of the paper for illustrative and analytical purposes.

B. Decision Mechanism

Initially, $\lambda_i(0)$ is set to zero for all i . For ease of discussion, we will drop the time-dependent notation on the transfer rate because we are interested in the capacity assignment based on the *instantaneous* transfer rate observed. This allows us to write λ_i and μ_i instead of $\lambda_i(t)$ and $\mu_i(t)$.

If a peer has nothing to contribute to its neighbor for any reason, there is no decision to be made and it will just wait until it receives something from a neighbor that is also interested by the others. If a peer has already finished downloading and stays in the system to be a “seeder”, the decision strategy is simply picking a random subset of neighbors and to serve them for a period of time, and then pick another random subset and repeat the process.

If a peer s has something to contribute to its neighbors, and it has not yet downloaded the complete content, it will run the Capacity Assignment Update (CAU) algorithm and decide the amount of capacity dedicated to its neighbors. The algorithm first computes the desired contribution to each neighbor using the observed instantaneous capacities λ_i 's and Eq. (1). If λ_i is smaller than some threshold $\Delta\lambda$, which means neighbor i is either having nothing to contribute or contributing slowly, the algorithm will assign $\Delta\mu = U_s(\Delta\lambda)$ to it if there is capacity available.

The algorithm then sorts all the λ_i 's in descending order and starts providing $\mu_i = U_s(\lambda_i)$ to the neighbors that are contributing with the highest instantaneous capacities. As a boundary case, one neighbor may be assigned less than $U_s(\lambda_i)$ because of resource limitation. The rest of the neighbors (if there are any) will be assigned zero capacity.

The introduction of the threshold value $\Delta\lambda$ and correspondingly $\Delta\mu$ in CAU algorithm has two purposes. First, it serves as the *seeding capacity* for newcomers. For peers that are not able

Algorithm 1 Capacity Assignment Update (CAU)

```
1: for  $i = 1$  to  $N$  do
2:   if  $\lambda_i < \Delta\lambda$  then
3:      $x(i) \leftarrow \Delta\mu \{= U(\Delta\lambda)\}$ 
4:   else
5:      $x(i) \leftarrow U(\lambda_i)$ 
6:   end if
7: end for
8:  $y_k \leftarrow$  the index  $i$  of the  $k^{th}$  largest  $\lambda_i$ 
9: {If  $\lambda_i = \lambda_j$ , the order is randomly picked.}
10:  $remain\_cap = \mu_{max}$ 
11: for  $j = 1$  to  $N$  do
12:   if  $x(y_j) \leq remain\_cap$  then
13:      $\mu_{y_j} \leftarrow x(y_j)$ 
14:      $remain\_cap \leftarrow remain\_cap - x(y_j)$ 
15:   else if  $remain\_cap > 0$  then
16:      $\mu_{y_j} \leftarrow remain\_cap$ 
17:      $remain\_cap \leftarrow 0$ 
18:   else
19:      $\mu_{y_j} \leftarrow 0$  {No more capacity available.}
20:   end if
21: end for
```

to contribute or contribute slowly for any reason (lack of content or neighbors not interested in content downloaded), the mechanism ensures that they will receive a minimum amount $\Delta\mu$ of data rate to build up their inventory and start contributing, provided that their neighbors *still have capacity remained after assigning to other contributors*.

The second purpose that $\Delta\mu$ serves is to trigger the *Cournot adjustment process* (Cournot, 1960). The original Cournot model is a generalization of the Cournot game to describe industry structure. Each of N firms will choose a quantity of output. Cournot adjustment process is a form of repeated game play where players ignore the effect of their current actions on the opponent's future actions. It is also a *myopic game*, where in which there is no communication between players, no memory of past events, or prediction of future payoffs. The adaptation is based on the current state of the game. Players take turns setting their outputs and each player's output is a best response to the output his opponent choice in the previous period.

Consider a two-firm game, where q_i^k for $i = 1, 2$ is the quantity produced by firm i in the k -th

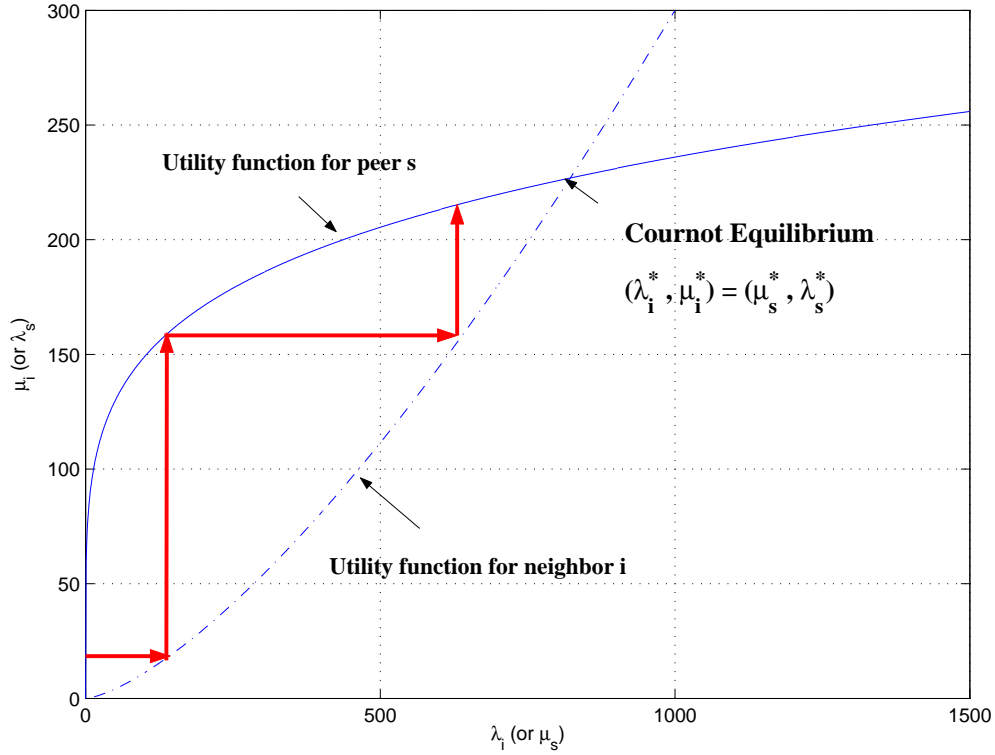


Fig. 2. Illustration of the Cournot adjustment process and the convergence to Cournot Equilibrium.

period. A Cournot reaction (utility) function r_i for firm i specifies each firm's optimal output for each fixed output level of its opponent. The Cournot adjustment process will have the sequence:

$$q_1^0 \rightarrow q_2^1 = r_2(q_1^0) \rightarrow q_1^2 = r_1(q_2^1) \rightarrow q_2^3 = r_2(q_1^2) \rightarrow \dots$$

The steady state of this adjustment process, or the Cournot equilibrium, will be:

$$q_1^* = r_1(q_2^*) \tag{3}$$

$$q_2^* = r_2(q_1^*) \tag{4}$$

It should be noted that a Cournot equilibrium is an example of a Nash equilibrium, and is sometime referred to as Cournot-Nash equilibrium. Cournot adjustment process is not limited to

two-player Cournot game, but also applicable to n -player game in general.

In the CAU algorithm, initially $\lambda_i = 0$ for all i . If two neighbors each has content pieces that are interested by the others, contributing $\Delta\mu$ by one party will jumpstart the trading process. A graphical illustration is outlined in Figure 2. Consider the case where peer s starts contributing $\Delta\mu$ to neighbor i , and neighbor i will respond by contributing $U_i(\Delta\mu)$ back to peer s in the next run of the CAU algorithm. Peer s , after receiving $U_i(\Delta\mu)$ from peer i , will respond by providing $U_s(U_i(\Delta\mu))$ to neighbor i . By repeating the CAU algorithm periodically, the process will continue until either (i) one party cannot keep up because of resource limitation (the boundary case) or (ii) Cournot Equilibrium is reached. Either way, both parties consider the trade desirable and are able to get the most out of each other. Since Cournot adjustment process is not limited to two peers and is applicable to n peers in general, the CAU algorithm will yield the Cournot Equilibrium of an n -player game.

In addition to achieving Cournot Equilibrium, periodically running the CAU algorithm will also allow the peers to adapt to the latest network condition, and the dynamic join/leave of neighbors. The computational complexity of CAU algorithm is $O(N \log N)$ (step 8 in the algorithm), where N is the number of neighbors a peer is connected to. Therefore, running CAU will impose a very low computation overhead on the peer.

IV. INCENTIVE COMPATIBILITY

In this section, we investigate the incentive-compatibility issues in our mechanism. We consider the incentives to follow the CAU algorithm, the truthful revealing of the utility function parameters, and the problem of free-riding.

A. Following the CAU algorithm

As we have seen in Section III, trading according to the utility function defined in Eq. (1) is considered to be a desirable trade by a peer. However, $U(\lambda)$ is the *maximum* amount of capacity a peer is willing to pay for λ , so the question becomes what the gain is (if any) if a peer contribute a value less than $U(\lambda)$ back to a neighbor that provides λ (except the boundary case where there is not enough resources to do so).

CAU algorithm ranks the instantaneous capacity provided by each neighbor, and those that share more will be given a priority to receive contributions in return. If peer m (for malicious) returns $(U_m(\lambda_i) - c)$ to neighbor i instead of $U_m(\lambda_i)$ according to CAU algorithm, it is more likely to be ranked lower than it should be. If the new ranking is still high enough and c is not too large, peer m may still receive λ_i from neighbor i . However, peer m has no information about the utility function of neighbor i , which has a direct effect on how large c can be; nor does it know about the condition of neighbor i 's neighbors. Therefore, by returning $(U_m(\lambda_i) - c)$, peer m faces the possibility of receiving a smaller λ_i , if not zero, than it should. As the primary objective of the peers is to completely download the content in the shortest time, any action that is not going to increase neighbors' contribution is not a desirable action for the peer.

Besides, peer m has no information about where the Cournot equilibrium between neighbor i and itself lies (since it does not know the value of λ_{\max} , μ_{\max} , and ω_i of peer i), so by truthfully returning $U_m(\lambda_i)$, it is more likely to get a larger λ_i in the next period because of the Cournot adjustment process.

B. Truthful revealing of utility parameters

Another incentive issue that is interesting to know is that if there is any incentive for the peer to truthfully reveal its λ_{\max} , μ_{\max} , and ω , which determine its utility function.

We consider again with peer m and its neighbor i . If truthfully revealed, the utility function for peer m is:

$$\mu_i = U_m(\lambda_i) = \lambda_i^{(1-\omega_m)} \frac{\mu_{\max,m}}{\lambda_{\max,m}^{(1-\omega_m)}} \quad (5)$$

and that for neighbor i is:

$$\mu_m = U_i(\lambda_m) = \lambda_m^{(1-\omega_i)} \frac{\mu_{\max,i}}{\lambda_{\max,i}^{(1-\omega_i)}} \quad (6)$$

At equilibrium, $\lambda_i^* = \mu_m^*$ and $\lambda_m^* = \mu_i^*$, Eq. (6) becomes

$$\lambda_i^* = \mu_i^{*(1-\omega_i)} \frac{\mu_{\max,i}}{\lambda_{\max,i}^{(1-\omega_i)}} \quad (7)$$

$$\text{or } \mu_i^* = \left(\lambda_i^* \frac{\lambda_{\max,i}^{(1-\omega_i)}}{\mu_{\max,i}} \right)^{\frac{1}{1-\omega_i}} \quad (8)$$

Setting $\lambda_i = \lambda_i^*$ and $\mu_i = \mu_i^*$ in Eq. (5) and combining it with Eq. (8), we have

$$\lambda_i^{*(1-\omega_m)} \frac{\mu_{\max,m}}{\lambda_{\max,m}^{(1-\omega_m)}} = \left(\lambda_i^* \frac{\lambda_{\max,i}^{(1-\omega_i)}}{\mu_{\max,i}} \right)^{\frac{1}{1-\omega_i}} \quad (9)$$

Taking logarithm on both sides,

$$\begin{aligned} & (1 - \omega_m) \log \lambda_i^* + \log \left(\frac{\mu_{\max,m}}{\lambda_{\max,m}^{(1-\omega_m)}} \right) \\ &= \frac{1}{1 - \omega_i} \log \lambda_i^* + \frac{1}{1 - \omega_i} \log \left(\frac{\lambda_{\max,i}^{(1-\omega_i)}}{\mu_{\max,i}} \right) \end{aligned} \quad (10)$$

After simplification, we have,

$$\lambda_i^* = \exp \left(\frac{k_1 - k_2}{\frac{1}{1-\omega_i} - (1 - \omega_m)} \right) \quad (11)$$

where

$$k_1 = \log \left(\frac{\mu_{\max,m}}{\lambda_{\max,m}^{(1-\omega_m)}} \right) \quad (12)$$

$$k_2 = \frac{1}{1 - \omega_i} \log \left(\frac{\lambda_{\max,i}^{(1-\omega_i)}}{\mu_{\max,i}} \right) \quad (13)$$

Since two peers will not trade beyond the Cournot Equilibrium level, it is obvious that peer m wants to maximize λ_i^* , the equilibrium (and maximum) capacity neighbor i would contribute. From Eq. (11) we can see that λ_i^* is an increasing function of k_1 , and from Eq. (12), k_1 is a decreasing function of $\lambda_{\max,m}$ and an increasing function of $\mu_{\max,m}$. Therefore, in order to maximize λ_i^* , peer m should set $\lambda_{\max,m}$ to the smallest desirable value and $\mu_{\max,m}$ to the largest

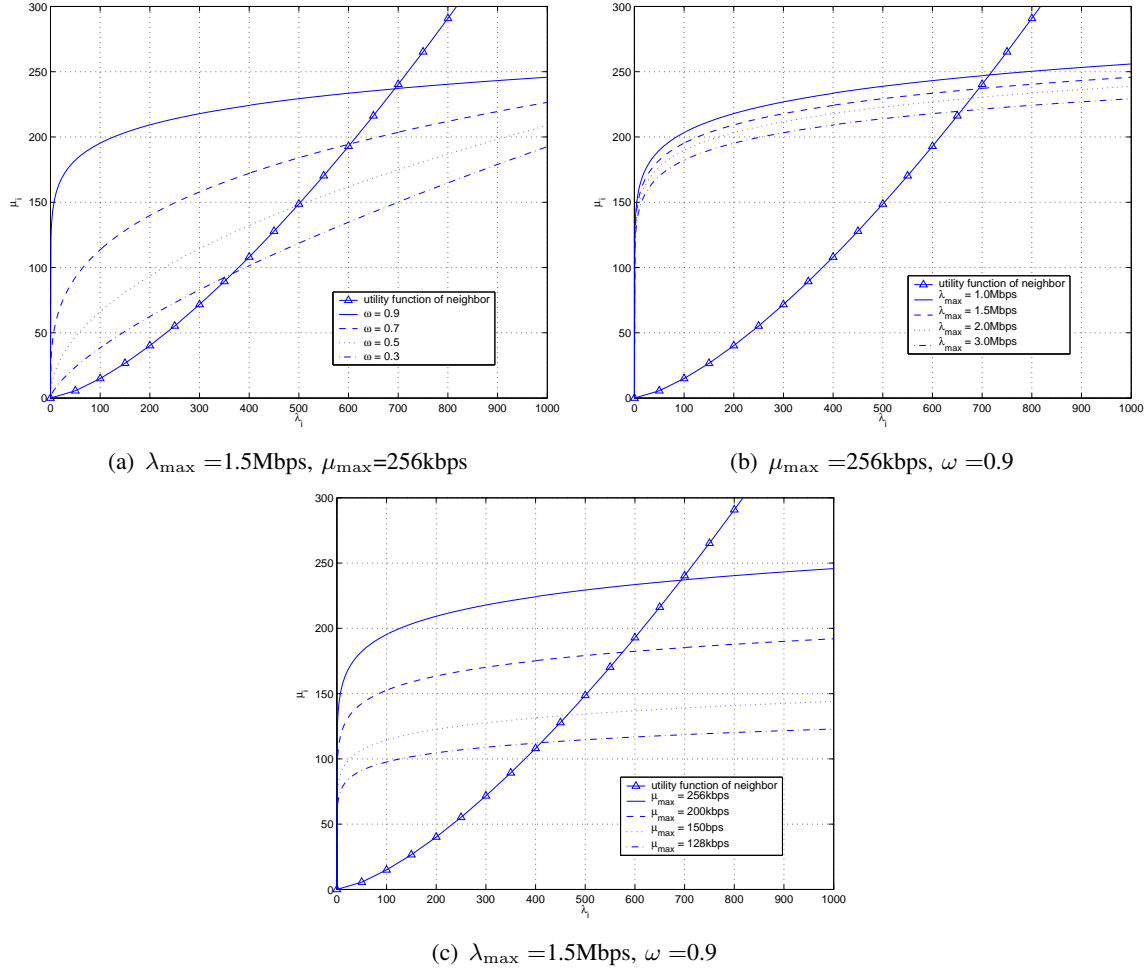


Fig. 3. Effect of different parameters on λ^* : (a) ω ; (b) λ_{\max} ; and (c) μ_{\max}

desirable value. There is absolutely no incentive for any peer to declare a λ_{\max} that is smaller than its physical limit because that will restrict the amount of incoming capacity. There is also no incentive to declare a μ_{\max} to be larger than a peer can physically supply, because for the same λ , $U(\lambda, \mu_{\max,i}) < U(\lambda, \mu_{\max,j})$ for all $\mu_{\max,i} > \mu_{\max,j}$, and Cournot Equilibrium also can never be reached if μ_{\max} is set to a value greater than one's physical limitation. As a result, revealing the true $(\lambda_{\max}, \mu_{\max})$ is the best strategy for any peer to achieve maximum λ^* .

The relationship between λ^* and ω and can be obtained by differentiating Eq. (11) with respect to ω . Once again, the parameter is positively related to λ^* , and over-declaring is clearly undesirable, so the best strategy for a peer is revealing its true ω . Figure 3 illustrates the

relationship between λ^* and the user-declared parameters.

C. Threshold values $\Delta\lambda$ and $\Delta\mu$

The CAU algorithm requires the peers to contribute $\Delta\mu$ to one's neighbors that are observed to be providing $\Delta\lambda$ or less capacity, if there is capacity remaining. The effect of non-compliance to these values by some malicious peers has only minor setback to the system performance. Peers with little or no content to contribute will have to wait for a longer period to get their "seed" content from complying peers. From a peer's perspective, setting a low $\Delta\mu$ will slow down the Cournot adjustment process, which results in an undesirably longer downloading time for itself. Furthermore, a low $\Delta\mu$ will also receive a lower ranking in CAU algorithm on the neighbors' end, which leads to low service priority from the neighbors.

D. Handling free-riders

Free-riders are not strictly prohibited under our proposed mechanism. However, a free-rider will only be served when either its neighbors have excess capacity or when it is connected to a "seeder" and randomly selected to be served. Their non-contribution will not degrade the performance of the system, as the resources they obtained is mostly surplus in the community that are not used otherwise.

V. COMPUTER SIMULATIONS

In this section we implemented the CAU algorithm and compared its performance against widely-adopted alternatives. We selected BitTorrent (BT) as the benchmark for comparison because of its distribution efficiency as a PCDS and its popularity (Borland, 2004). Please refer to Cohen (2003) for the implementation details of BT.

A. Simulation Setup

We used the experimental setup described in Koo et al. (2004) to evaluate the performance of the CAU algorithm using *ns-2*. In the setup, peers could be one of the following three classes:

- **Class 1:** 10 Mbps bi-directional link, representing campus LAN hosts.
- **Class 2:** 128 kbps uplink and 1.5 Mbps downlink, representing ADSL hosts.
- **Class 3:** 56 kbps bi-directional link, representing (roughly) dial-up hosts.

The number of peers in each class was exogenously set. Results reported in this paper were obtained assuming that 30% of peers were Class 1, 50% were Class 2, and 20% were Class 3. We have also simulated other peer compositions, and the simulation results shown in this section demonstrate the typical trend. The arrival pattern of these peers was taken from real traffic traces from a tracker running on the Internet, and the departure of peers after completion of download followed an exponential distribution. In the simulation, we considered a Class-1 peer to be the seeder, which has a file size of 700 MB (volume of a CD-ROM), divided into 1400 pieces (512 kB each) to distribute. The simulation is repeated multiple times for 150 simulation-hours each to obtain statistical significant results. More than 2000 nodes are generated in each of the experiments, and there are about 400 to 500 nodes concurrently downloading the content at any time after the initial seeding phrase.

For the model of intermediate network, we adopted the one used in Koo et al. (2003) where it is assumed that the capacity bottleneck was always at the last hop, that is, either the outgoing capacity of the sender or the incoming capacity of the receiver was the cause of end-to-end congestion. Each peer can be connected to at most 20 neighbors. We also assumed the presence of a generic BT tracker (essential for BT to function, but only needed by CAU for selecting neighbor peers), and the bandwidth of the tracker was assumed abundant. We implemented the CAU algorithm and the generic BT “tit-for-tat” mechanism in the peers, and they followed the selected exchange mechanism once they are assigned to their neighbors.

The *willingness-to-pay* for each peer i (ω_i) in the experiment is uniformly distributed between 0.1 and 0.9, and the value of λ_{max} and μ_{max} are set to the respective maximum. (These values are only used by CAU. BT does not need them.)

TABLE I
AVERAGE DOWNLOADING TIME IN MINUTES. VARIANCES ARE SHOWN IN PARENTHESES.

	BT	CAU
Class 1	37.55 (0.14)	32.51(0.12)
Class 2	131.86 (0.31)	119.73(0.35)
Class 3	3234.77 (60.12)	3089.18 (55.84)

B. Simulation Results

Two metrics were utilized for comparison: (i) percentage gain in utility P_{gain} and (ii) uplink and downlink utilization. The percentage gain in utility is defined as the ratio of the total *net* utility received to the total contribution. Equivalently,

$$P_{gain} = \frac{\int \sum_i (U(\lambda_i(t)) - \mu_i(t)) dt}{\int (\sum_i \mu_i(t)) dt} \times 100\% \quad (14)$$

This metric measures the satisfaction of a peer throughout the whole process. Figure 4 shows the distribution of P_{gain} measured from the experiment. Since in the CAU algorithm peers only trade when it is desirable, except for seeding the newcomers, it is not surprising that the majority of peer experience a positive P_{gain} . On the other hand, even though BT provides incentives for peers to contribute by serving the highest contributors first, it does not consider the *absolute* amount of contribution. For example, if 10 neighbors are connected to a peer, and 9 of them serving 1kbps, the tenth one will be given the highest priority if it is serving 2kbps. As a result, a peer may *over-contribute* and receive negative utility. The average value of P_{gain} for CAU clients is 30% and that for BT clients is -1280%! This result also explains why in reality many BT clients prefer to limit their upload capacity to a value lower than μ_{max} , as they are getting negative utility from the community.

The average downloading time required by each class of peers to download the complete content is shown in Table I. Simulation results show that for the same class of nodes, those that use CAU have a lower average than their counterparts. Such gain can be explained as a result of better link utilization.

We measured the average link utilization for each class of peers under BT and CAU. Both uplink and downlink are measured, and the results are shown in Figure 5. As mentioned above, BT protocol does not limit uplink contribution so a higher uplink utilization compare to CAU clients is expected. The experimental results for class-2 and class-3 peers concur with this expectation. Intriguingly, the same phenomenon does not happen to class-1 peers. The reason for this counterintuitive result can be explained by the following: In BT, a client chooses the top K neighbors to serve. However, the aggregated available downlink capacity of all the selected neighbors may be much lower than the uplink capacity available from the contributing peer. This rarely happens in CAU, where peers carefully budget their capacity assignment and utilize any available uplink capacity to try to get more contributions from the community. This leads to a significant result: All three classes of peers are benefited and have a higher downlink utilization because of the better utilization of capacity from class-1 peers, and high downlink utilization translate to high distribution efficiency, which is the primary objective of PCDS.

VI. DISCUSSIONS

The strength of the CAU algorithm is that it does not require any information exchange among peers (except those on content availability, which is required by almost all P2P networks). The information one relies on can be obtained locally by measuring or estimating the incoming capacities received from one's neighbors. This is particularly useful in a fully-distributed environment, where cooperation among peers is not guaranteed.

However, how to measure or estimate instantaneous capacity is a problem that has long been considered and no vastly acceptable model is available. In our experiments, we adopt a simplistic approach used by many other researchers, and that is to use the amount of data received from a neighbor, take the average over time, and use the value as the instantaneous capacity. This time-average estimation is not the optimal measure, but it provides important insights to the study.

It is also interesting to know, BitTorrent, one of the most popular P2P applications nowadays, takes on a special case of the CAU algorithm, by setting $\Delta\lambda$ to be λ_{\max} and $\Delta\mu$ to be $K \times \mu_{\max}$,

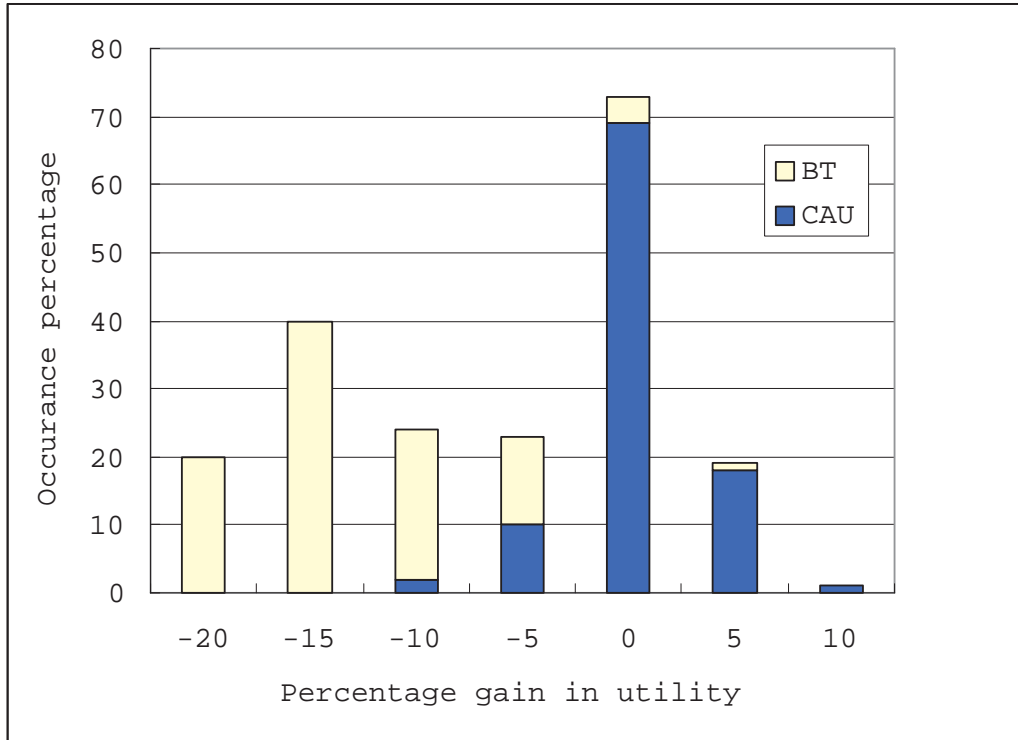


Fig. 4. Distribution of percentage gain in utility in BT and CAU.

where K is the number of neighbors that the peer is going to send data to. BitTorrent assigns all of a peers outbound capacity to the K highest contributing neighbors (plus another peer under “optimistic unchoking” (Cohen, 2003), which is randomly picked by the peer to serve), so it should have a high distribution efficiency under a homogeneous, high-speed network. But as Section V suggests, in a heterogeneous environment where some peers have faster links than the others, BitTorrent protocol becomes insufficient while CAU will better utilize the peers capacity to achieve a higher efficiency. Also, BitTorrent does not provide incentive for peers to reveal their true connectivities.

Throughout the experiment we assumed there is a data integrity checking mechanism to avoid data pollution (Liang et al., 2005). The design of such mechanism is out of the scope of this paper, but the mechanism is an important guard to malicious users who want to defeat the system. For example, if a new peer who doesn’t have any content pieces wants to get the file faster, it

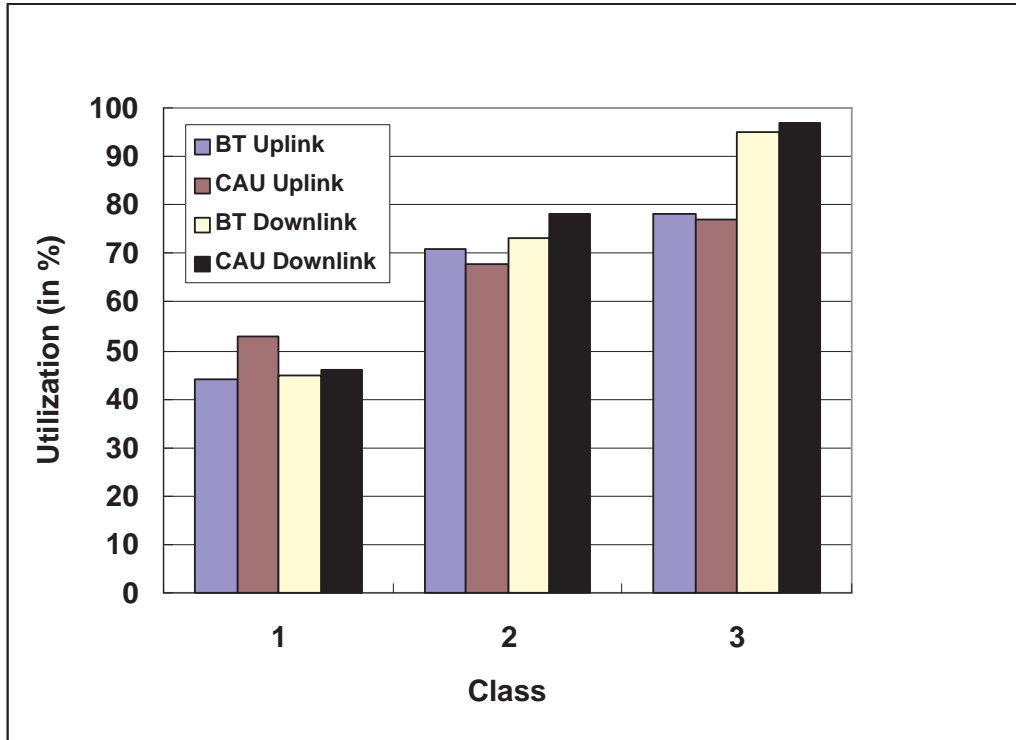


Fig. 5. Link utilization for different classes using BT and CAU.

may send junk data to its neighbors at a high rate, and get a much better service rate in return. With a simple checking mechanism (see Cohen (2003) for the one used in BitTorrent), peers will be able to verify the integrity of the content pieces received from their neighbors, ignore the polluted data, and nullify the false contributions.

VII. CONCLUSIONS

In this paper, we addressed the incentive provisioning problem for distribution of large-volume contents in PCDS, and presented a “seeing-is-believing” protocol CAU in which a peer will decide how much capacity will be assigned to which neighbors based on what it has experienced. We applied the Cournot adjustment process where peers will try to maximize their contributions for a fair or better return, and we showed that by adopting CAU, the system will achieve the Cournot Equilibrium. We also showed through analysis that there are incentives to comply

with the CAU algorithm, and CAU encourages peers to reveal their true willingness-to-pay and uplink/downlink capacities.

Because the CAU algorithm does not rely on any reputation/history exchange, the overhead it imposes on the system is minimal, and the algorithm itself runs in $O(N \log N)$ time, where N is the number of neighbors a peer connects to. Simulation results confirmed that CAU performs better than widely-adopt counterparts like BitTorrent in terms of both user satisfaction and distribution efficiency, the primary objective of a PCDS.

REFERENCES

- Aberer, K. and Despotovis, Z. (2001). Managing Trust in a Peer-2-Peer Information System. In *Proc. of the Ninth International Conference on Information and Knowledge Management (CIKM 2001)*, pages 310–317, Atlanta, GA.
- Anagnostakis, K. G. and Greenwald, M. B. (2004). Exchange-based Incentive Mechanisms for Peer-to-Peer File Sharing. In *Proc. of the 24th Intl. Conf. on Distributed Computing Systems (ICDCS '04)*, pages 524–533, Tokyo, Japan.
- Antoniadis, P., Courcoubetis, C., and Mason, R. (2004). Comparing economic incentives in peer-to-peer networks. *Computer Networks*, 46:133–146.
- Borland, J. (2004). Survey: Movie-swapping up; Kazaa down. CNET News.com. [Online] <http://news.com.com/2100-1025-5267992.html>.
- Buchegger, S. and le Boudec, J.-Y. (2004). A Robust Reputation System for P2P and Mobile Ad-hoc Networks. In *Proc. of the Second Workshop on Economics of Peer-to-Peer Systems*, Cambridge, MA.
- Buragohain, C., Agrawal, D., and Suri, S. (2003). A Game Theoretic Framework for Incentives in P2P Systems. In *Proc. of the Third Intl. Conf. on Peer-to-Peer Computing (P2P'03)*, pages 48–56.
- Cohen, B. (2003). Incentives Build Robustness in BitTorrent. In *Proc. of the First Workshop on Economics of Peer-to-Peer Systems*, Berkeley, CA.

- Cournot, A. (1960). *Research into the Mathematical Principles of the Theory of Wealth*. Wiley, New York.
- Dilley, J., Maggs, B., Parikh, J., Prokop, H., Sitaraman, R., and Weihl, B. (2002). Globally Distributed Content Delivery. *IEEE Internet Computing*, pages 50–58.
- eDonkey (2000). [Online] <http://www.edonkey2000.com/>.
- Ge, Z., Figueiredo, D. R., Jaiswal, S., Kurose, J., and Towsley, D. (2003). Modeling Peer-Peer File Sharing System. In *Proc. of INFOCOM 2003*, San Francisco, CA.
- Ghosal, D., Poon, B. K., and Kong, K. (2005). P2P contracts: a framework for resource and service exchange. *Future Generation Computer Systems*, 21(3):333–347.
- Halberstadt, A., Mui, L., and Mohtashemi, M. (2002). A Computational Model of Trust and Reputation. In *Proc. of the 35th Hawaii International Conference on System Science (HICSS)*, pages 2431–2439.
- Hardin, G. (1968). The Tragedy of the Commons. *Science*, 162:1243–1248.
- Jurca, R. and Faltings, B. (2003). An Incentive Compatible Reputation Mechanism. In *Proc. of the IEEE Conference on e-Commerce*, pages 285–292, Newport Beach, CA.
- Kamvar, S. D., Schlosser, M. T., and Garcia-Molina, H. (2003). The Eigentrust Algorithm for Reputation Management in P2P Networks. In *Proc of the Twelfth International World Wide Web Conference*, pages 640–651.
- Koo, S. G. M., Kannan, K., and Lee, C. S. G. (2004). A Genetic-Algorithm-Based Neighbor-Selection Strategy for Hybrid Peer-to-Peer Networks. In *Proc. of the 13th IEEE International Conference on Computer Communications and Networks (ICCCN'04)*, pages 469–474, Chicago, IL.
- Koo, S. G. M., Rosenberg, C., and Xu, D. (2003). Analysis of Parallel Downloading for Large File Distribution. In *Proc. of the 9th IEEE International Workshop on Future Trends of Distributed Computing Systems (FTDCS 2003)*, pages 128–135, San Juan, Puerto Rico.
- Kung, H. T. and Wu, C. (2003). Differentiated Admission for Peer-to-Peer Systems: Incentivizing Peers to Contribute Their Resources. In *Proc. of the 2nd International Workshop on Peer-to-Peer Systems (IPTPS '03)*, Berkeley, CA.

- Lai, K., Feldman, M., Stoica, I., and Chuang, J. (2003). Incentives for Cooperation in Peer-to-Peer Networks. In *Proc. of the First Workshop on Economics of Peer-to-Peer Systems*, Berkeley, CA.
- Liang, J., Kumar, R., Xi, Y., and Ross, K. (2005). Pollution in P2P File Sharing Systems. In *Proc. of INFOCOM*, Miami, FL.
- Luenberger, D. G. (1997). *Microeconomic Theory*. McGraw-Hill.
- Ma, R. T. B., Lee, S. C. M., Lui, J. C. S., and Yau, D. K. Y. (2004). An Incentive Mechanism for P2P Networks. In *Proc. of the 24th Intl. Conf. on Distributed Computing Systems (ICDCS '04)*, pages 516–523, Tokyo, Japan.
- Moreton, T. and Twigg, A. (2003). Trading in Trust, Tokens, and Stamps. In *Proc. of the First Workshop on Economics of Peer-to-Peer Systems*, Berkeley, CA.
- Ngan, T.-W., Wallach, D., and Druschel, P. (2003). Enforcing Fair Sharing of Peer-to-Peer Resources. In *Proc. of the 2nd International Workshop on Peer-to-Peer Systems (IPTPS '03)*, Berkeley, CA.
- Odlyzko, A. (2003). The Case Against Micropayments. In Camp, J. and Wright, R., editors, *Financial Cryptography 2003*. Springer.
- Qiu, D. and Srikant, R. (2004). Modeling and Performance Analysis of BitTorrent-Like Peer-to-Peer Networks. In *Proc. ACM SIGCOMM*, pages 367–378, Portland, OR.
- Resnick, P., Zeckhauser, R., Friedman, E., and Kuwabara, K. (2000). Reputation Systems. *Communications of the ACM*, 43(12):45–48.
- Skenker, S. (1995). Fundamental Design Issues for the Future Internet. *IEEE Journal on Selected Areas in Communications*, 13(7):1176–1188.
- Stavrou, A., Rubenstein, D., and Sahu, S. (2002). A Lightweight, Robust P2P System To Handle Flash Crowds. In *Proceedings of IEEE ICNP 2002*, Paris, France.
- Tamilmani, K., Pai, V., and Mohr, A. (2004). SWIFT: A System With Incentives For Trading. In *Proc. of the Second Workshop on Economics of Peer-to-Peer Systems*, Cambridge, MA.
- Yang, X. and de Veciana, G. (2004). Service Capacity of Peer to Peer Networks. In *Proc. of INFOCOM 2004*, Hong Kong.