Optimal Peer Selection in a Free-Market Peer-Resource Economy

Micah Adler[*], Rakesh Kumar[†], Keith Ross[‡], Dan Rubenstein[§], David Turner[¶] and David D. Yao[||]
[*]Dept. of Computer Science University of Massachusetts Amherst, MA; Email: micah@cs.umass.edu
[†]Dept. of Electrical Engineering, Polytechnic University, NY; Email: rkumar04@utopia.poly.edu
[‡]Dept. of Computer and Information Science, Polytechnic University, NY; Email: ross@poly.edu
[§]Dept. Of Electrical Engineering, Columbia University, NY; Email: danr@ee.columbia.edu
[¶]Dept. Of Computer Science, California State University San Bernardino, CA; Email: dturner@csusb.edu
[||]IEOR Dept., Columbia University, NY; Email: yao@columbia.edu

***Abstract*— In a P2P free-market resource economy a client peer may want to select multiple server peers for either downloading a file or streaming a stored audio or video object. In general, multiple server peers will make the object available, with each peer offering a different price. The optimal peer selection problem is to select from a subset of the peers those peers that can provide the service at lowest cost. In this paper we formulate and solve the problem of optimally selecting a subset of peers for parallel downloading and for parallel streaming.**

I. Introduction

Today many peer computers participate in peer-to-peer file sharing applications in which the computers contribute storage and bandwidth resources. File sharing, however, is not the only application that can exploit resources made available by vast numbers of peer computers. Other important applications that could exploit a vast pool of peer resources include peer-driven content distribution networks [1], globally distributed archival storage [5], [6], [12], and massively parallel computation [13].

Of course, applications can only harness the resource pool if peers make available their surplus resources to them. It is widely documented, however, that the P2P systems are havens for "free riders": a significant fraction of users do not contribute any resources, and a minute fraction of users contribute the majority of the resources [3], [4]. Clearly, to improve the performance of existing P2P file sharing systems, and to enable new classes of P2P applications, a compelling incentive system needs to be put in place.

Now suppose the existence of an online market place where entities - such as peers, companies, users and so on - buy and sell surplus resources. In this market place, a peer might purchase storage and bandwidth from a dozen other peers for the purpose of remotely backing up its files; a content publisher might purchase storage and bandwidth from thousands of peers to create a peer-driven content distribution network; a biotechnology company might purchase CPU cycles from thousands of peers for distributed computation. If such a flourishing resource market existed, individual peers would be incited to contribute their resources to the market place, thereby unleashing the untapped resource pool.

We envision a free market in which peers buy and sell resources directly from each other [1]. In this free market, selling peers are free to set the prices of their resources as they please. A client peer, interested in purchasing a specific resource, is permitted to "shop" the different server peers and choose the peers that best satisfy its needs at the best prices. The "money" paid by the client peers and earned by the server peers may be real money, such as US dollars, or may be some pseudo-money akin to frequent flyer miles. After a seller earns money, it can later spend the money in the resource market, obtaining resources from other seller peers. For example, a peer Alice may earn money by transferring a portion of a video file to some peer Bob. Peer Alice may then use this earned money to purchase a different video file from a different peer Claire. Or peer Bob may use the money to purchase a different type of resource, such as backup storage or CPU cycles, from peer Claire.

In the context of this free-market P2P resource economy, this paper considers the problem of optimal server peer selection. Specifically, we consider the following problem. A client peer wants to obtain an object, such as a specific video file, directly from other peers attached to the Internet. The client may obtain different portions of the object in parallel from different server peers (as is currently the case with KaZaA and other file sharing systems). In this resource economy, the server peers charge client peers when they send files (or portions of files) to the client peers. We suppose a free market economy, that is, server peers may price their services as they wish.

More specifically, when a client peer wants to obtain a specific object, the following steps are taken:

1) The client first uses a look-up service to discover server peers that have a copy of the object. KaZaA is an example of such a lookup service; structured DHTs could also be used to create a lookup service.
2) The client then queries the server peers for their prices.
3) The client may also use a reputation service to determine the reliability of each of the server peers. (Reputation services are beyond the scope of this paper; see [9], [10].)
4) From the subset of reputable server peers offering the object, the client peer selects the server peers from which it will obtain the object. The client obtains different portions of the object from each of the selected server peers. The client peer will naturally choose the server peers to minimize cost.
5) Money is transferred from the client peer to the server

peers. A protocol for transferring money in a P2P resource market is described in [1].

In this paper we explore the optimal peer selection problem for two delivery schemes: (i) **streaming**, in which case the portions of the object must arrive in a timely manner, so that the client peer can render the object without glitches during downloading; (ii) **downloading**, in which case the client wants to receive the file as quickly and inexpensively as possible, but does not render the file while downloading. We will see that this two delivery schemes lead to different formulations of the problem.

For both schemes, a content publisher may also be an active component of the system. For example, CNN.com may contract with a large number of peers to store chunks of video files. When another peer Alice asks CNN to see a video, CNN may select the peers on Alice's behalf. The selected peers would then either stream or download the video, depending on the delivery scheme.

II. Pricing Model

In this section we describe our pricing model. As mentioned in the introduction, each server is free to set its own prices. Consider a server peer i. As part of a delivery session, the server peer i will transfer a portion of the bytes of some object o to a client peer. For such a delivery, the server peer will fix an appropriate price. This price should naturally depend on:

- **The object itself:** For example, for video content, recent videos may be more expensive than older videos.
- **The number of bytes transferred:** The amount of bandwidth resources consumed at the server is proportional to the number of bytes transferred.
- **Rate of transfer:** The server may want to charge more for transferring the bytes at a higher rate.

Let Ω_i be the set of rates at which the server peer can transfer the object portion; Ω_i could consist of discrete values or could be a continuous interval of values. For a particular object o, we consider pricing functions of the form

$$\text{price} = C_i(b, x)$$

where $b \in \Omega_i$ is the rate of the transfer and x is the number of bytes transferred. Note that if a client obtains x bytes at rate b bytes/sec, then the transfer time is x/b seconds. Also note that we are taking the natural assumption that all bits have the same price.

Before proceeding, let us examine more carefully what it means for a server to be able to transfer bytes at a specific rate b. A server i will have Internet access with some upstream rate u_i. At any given time, the server peer i could be transferring files to multiple peers, with each file transfer taking place at its negotiated rate. In order to meet its commitment, server i, of course, must ensure that the sum of all the ongoing transfer rates does not exceed its upstream access rate u_i. In today's Internet (and in the foreseeable future), the bottleneck is typically in the access and not in the Internet core. In most broadband residential connections today (including cable modem and ADSL), the upstream rate is significantly less than the downstream rate. Thus, it is not unreasonable to assume that bandwidth bottleneck between server and client is the server's upload rate. Thus, it is reasonable to assume that a server can provide an offered rate b, as long as the sum of server's committed ongoing rates is less than u_i.

There will be situations, however, when the server will not be able to honor its commitment due to unusual congestion or service failures in the core. In this case, the client peer may want some form of a refund. Furthermore, either the server or the client may be dishonest and claim that service was not received/rendered when indeed it was. Thus, some form of arbitration - preferably lightweight - may be needed in a P2P resource market [1]. In Section IV, we will describe a client strategy that allows any one of the the contracted peers to fail, either because of unforeseen problems beyond the server's control or because of dishonesty.

III. Optimal Peer Selection for Downloading

In this paper we explore the optimal peer selection problem for two delivery schemes: (i) streaming, in which case the portions of the object must arrive in a timely manner, so that the client peer can render the object without glitches during downloading; (ii) downloading, in which case the client wants to receive the file as quickly and inexpensively as possible, but does not render the file while downloading. In this section we consider the downloading problem.

For the download problem, we introduce a simplification in our pricing model. Specifically, we assume that $\Omega_i = \{b_i\}$, that is, each server i advertises a single download rate b_i. We also make the natural assumption that the server's price is proportional to the number of bytes transferred, that is, we consider pricing functions of the form

$$\text{price} = c_i x_i \tag{1}$$

Thus, under this pricing model, a server i advertises a specific transfer rate b_i and a specific cost per byte c_i.

Naturally, a client desiring a specific object o would like to obtain the object as quickly as possible and at least possible cost. These two objectives will typically be conflicting, as servers that provide high transfer rates will likely also demand high per byte transfer costs. There are many ways to formulate an optimization problem that takes into account these conflicting goals. In this section, we consider one such natural formulation: the client wants to select the peers in order to minimize the total object download time subject to a budget constraint for a download.

We can now describe the optimal peer selection problem. Consider a client peer that wants to download a file o. Let F be the size of the file. As described in the Introduction, the client peer uses a location service to find a set of peers, denoted $N = \{1, \cdots, I\}$, that have a copy of the file. Each server peer $i \in N$ offers the client peer a transfer rate b_i and cost per byte c_i. We suppose that the client peer has a budget K for this particular download, that is, the client peer is prepared to spend up to K units on this download. Let x_i be the number of bytes that the client transfers from server peer i. Because the client wants to obtain the entire file, we have $x_1 + \cdots + x_I = F$.

Our optimal peer selection problem is to determine x_i, $i \in N$ that minimizes the total download time subject to the budget constraint. Because the client is downloading from multiple server peers in parallel, the total download time is the maximum download time from each of the selected peers. Because the download time from peer i is x_i/b_i, the total download time is the maximum of x_i/b_i, $i = 1, \ldots, I$.
Thus, our goal is to determine optimum values of x_i, $i = 1, \ldots, I$ for

$$min\ max\{\frac{x_1}{b_1}, \frac{x_2}{b_2}, ...\frac{x_I}{b_I}\} \tag{2}$$

subject to

$$c_1 x_1 + c_2 x_2 + ... + c_I x_I \leq K \tag{3}$$

$$x_1 + x_2 + ... + x_I = F \tag{4}$$

$$x_i \geq 0 \quad i = \{1, \cdots, I\} \tag{5}$$

We now proceed to solve this optimization problem. First, we re-order the server peers so that

$$0 < c_1 < \cdots < c_I \tag{6}$$

Note that all of the parameters b_i, c_i, F and K are positive constants. Write the above optimization problem as a linear program (LP):

$$\begin{aligned} & \min y && (7)\\ \text{s.t.} \quad & \sum_i c_i x_i \leq K, && (8)\\ & \sum_i x_i \geq F, && (9)\\ & 0 \leq x_i \leq b_i y, \quad \forall i. && (10) \end{aligned}$$

The dual of the above is as follows, with the dual variables v and w corresponding, respectively, to the two constraints in (8) and (9), and z_i corresponding to $x_i \leq b_i y$ in (10):

$$\begin{aligned} & \min Fw - Kv && (11)\\ \text{s.t.} \quad & w - z_i - c_i v \leq 0, && (12)\\ & \sum_i b_i z_i \leq 1, && (13)\\ & v \geq 0,\ w \geq 0,\ z_i \geq 0,\ \forall i. && \end{aligned}$$

Below, we start with deriving a dual feasible solution, which leads to a primal feasible solution via complementary slackness. Once these are verified — dual and primal feasibility and complementary slackness, the problem is completely solved. To simplify notation, write

$$B_j := \sum_{i=1}^{j} b_i, \quad \beta_j := \sum_{i=1}^{j} b_i c_i; \tag{14}$$

and write B_I and β_I simply as B and β. It is easy to verify that β_j/B_j is increasing in j, since

$$\frac{\beta_j}{B_j} \leq \frac{\beta_{j+1}}{B_{j+1}} \quad \text{iff} \quad \beta_j \leq B_j c_{j+1},$$

and the last inequality follows from (6).
Letting the constraints in (12) and (13) be binding, we get:

$$\begin{aligned} z_i &= w - c_i v, && (15)\\ w &= \frac{1 + v\beta}{B}. && (16) \end{aligned}$$

Then, the dual objective becomes

$$Fw - Kv = \frac{F}{B} + \Big(\frac{F\beta}{B} - K\Big)v.$$

Since we want to maximize the above, w.r.t. v, there are two cases:

(i) If $\frac{\beta}{B} \leq \frac{K}{F}$, then $v = 0$.
(ii) Otherwise, consider, for the time being, $\frac{\beta}{B} > \frac{K}{F} \geq \frac{\beta_{I-1}}{B_{I-1}}$. Then $v = v_I := 1/(c_I B - \beta)$.

Note in Case (ii), $c_I B > \beta$ follows from (6); and for all i,

$$z_i = \frac{1}{B} + \Big(\frac{\beta}{B} - c_i\Big) v_I \geq 0$$

follows from

$$v_I \leq \frac{1}{c_i B - \beta}, \quad \forall i:\ c_i B > \beta,$$

since $c_n \geq c_i$. Also note that $z_n = 0$.
In case (i), the dual feasible solution results in a dual objective value F/B. The corresponding primal feasible solution is:

$$x_i = \frac{F b_i}{B}, \quad \forall i. \tag{17}$$

It is easy to verify that complementary slackness is satisfied: all primal variables are positive, and all dual constraints are binding; all dual variables except v are positive, and all primal constraints except (8) are binding. Furthermore, note that the primal objective value is also F/B.
For case (ii), the dual feasible solution results in a dual objective value as follows:

$$\frac{F}{B} + \Big(\frac{F\beta}{B} - K\Big) \cdot \frac{1}{c_I B - \beta} = \frac{F c_I - K}{c_I B - \beta}. \tag{18}$$

For the corresponding primal solution, consider the following:

$$x_i = b_i y, \quad \forall i \neq I; \qquad x_I = F - y B_{I-1}; \tag{19}$$

where y is the primal objective value, obtained via substituting the above solution into (8) and making the latter an equality:

$$y\beta_{I-1} + c_I\Big(F - yB_{I-1}\Big) = K,$$

from which we can obtain

$$y = \frac{Fc_I - K}{c_I B_{I-1} - \beta_{I-1}} = \frac{Fc_I - K}{c_I B - \beta}, \tag{20}$$

i.e., the primal objective value is equal to the dual objective value in (18).

We still need to verify primal feasibility and complementary slackness. First note that $y \geq 0$ follows from (18): both terms on its LHS are positive. That $x_n \geq 0$ is equivalent to

$$\frac{F}{B_{I-1}} \geq \frac{F}{B} + \Big(\frac{F\beta}{B} - K\Big) v_I,$$

which simplifies (with some algebra) to $\frac{K}{F} \geq \frac{\beta_{I-1}}{B_{I-1}}$, the assumed condition in

Case (ii). Other aspects of primal feasibility hold trivially. Complementary slackness is readily verified: all primal variables are positive, and all dual constraints are binding; all dual variables except z_n are positive, and all primal constraints except $x_I \leq b_I y$, the I-th in (10), are binding.

Next, in Case (ii), suppose K/F falls into the following range:

$$\frac{\beta_{I-1}}{B_{I-1}} > \frac{K}{F} \geq \frac{\beta_{I-2}}{B_{I-2}}.$$

Then, the dual solution is:

$$v = v_{I-1} := \frac{1}{(c_{I-1}B_{I-1} - \beta_{I-1})}, \qquad w = \frac{1 + v_{I-1}\beta_{I-1}}{B_{I-1}};$$

and

$$z_i = w - c_i v, \quad i \leq I-1; \qquad z_I = 0.$$

The primal solution is:

$$x = b_i y, \quad i \leq I-2; \qquad x_{I-1} = F - yB_{I-2}, \qquad x_I = 0;$$

and

$$y = \frac{Fc_{I-1} - K}{c_{I-1}B_{I-2} - \beta_{I-2}} = \frac{Fc_{I-1} - K}{c_{I-1}B_{I-1} - \beta_{I-1}}.$$

Feasibility (primal and dual) and complementary slackness can be verified as earlier.

To summarize, we have

Proposition 1: The solution to the optimization problem in (2) takes the following form (with the costs ordered in (6) and the notation B_j, β_j, B, β defined in (14)).

If $K/F \geq \beta/B$, then $x_i = b_i y$ for all i, where $y = F/B$. Otherwise, suppose for some $j \leq n$,

$$\frac{\beta_j}{B_j} > \frac{K}{F} \geq \frac{\beta_{j-1}}{B_{j-1}}.$$

(If $\beta_1/B_1 > K/F$, i.e., $K < Fc_1$, then there is no feasible solution to the optimization problem in (2).)

Then,

$$x_i = b_i y, \quad i \leq j-1; \qquad x_j = F - yB_{j-1}$$

$$x_{j+1} = \cdots = x_n = 0;$$

where

$$y = \frac{Fc_j - K}{c_j B_j - \beta_j}.$$

In both cases, y is the optimal objective value.

IV. Optimal Selection for Streaming

In this section we consider streaming encoded (compressed) audio or video. The delivery constraints are more stringent than for downloading: in order to prevent glitches in playback, the servers must continuously deliver segments of the object on or before scheduled playout times.

An important parameter for the streaming delivery is the object's playback rate, denoted by r. For an object of size F with playback rate r, the viewing time is $T = F/r$ seconds. Suppose the user at the client begins to view the video at time 0. A fundamental constraint in the streaming problem is that for all times t with $0 \leq t \leq T$, the client must receive the first $r \cdot t$ bytes of the object. We refer to this constraint as the "no-glitch" constraint. Thus, when selecting the server peers and the object portions to be obtained from each server peer, the client must ensure that this no-glitch constraint is satisfied. A natural optimization problem is, therefore, to select the peers in order to minimize the total streaming cost subject the no-glitch constraint.

For the streaming problem, it is critical that the service continue even in one of the server peers fails to provide its service. For this reason, we consider the problem of contracting an additional peer in such a manner that if any one server peer fails, the client can continue to render the object without glitches. In future we will examine multiple peer failures.

In this section, we suppose that each server peer advertises a set of rates Ω_i, as described in Section I. To simplify the analysis, we suppose that $\Omega_i = [0, \infty]$ for all $i \in N$. However, more general rate sets can be handled as well. For each advertised rate $b \in \Omega_i$, the server advertises an associated cost rate of $c_i(b)$. Thus, server i charges $c_i(b) \cdot t$ to transfer bytes at rate b for t seconds.

The client must not only select a subset of peer servers, but it must also determine and schedule the specific portions of the file it is to download. There are two broad approaches that can be taken to solving this problem: **time segmentation** and **rate segmentation**. In time segmentation, the video is partitioned along the time axis in distinct segments, and each server is responsible for streaming only one of the segments in the partition. This approach requires client buffering. Furthermore, in the optimal solution, the client will typically receive the video from all the selected servers at the beginning of the video and from only one of the selected servers at the end of the video. This means that the client must be able to download (at the beginning of the video) at a rate that is equal to the sum of the server download rates, which will exceed the playback rate. In rate segmentation approach, each of the selected servers contributes bytes for each of the frames in the video, and at any instant of time the client downloads at the playback rate. In this paper we consider only rate segmentation.

A. Problem Formulation

For non-decreasing cost functions $c_i(\cdot)$, $i = 1, \ldots, I$, the rate segmentation problem is

$$\begin{aligned} \min \quad & c_1(b_1) + \cdots + c_I(b_I) \qquad (21) \\ \text{s.t.} \quad & \sum_{j \neq i} b_j \geq r, \quad i = 1, \ldots, I. \\ & b_i \geq 0, \quad i = 1, \ldots, I. \end{aligned}$$

Note that the rate constraint ensures that client continues to receive at rate at least r even when one of the servers becomes unavailable. The above problem can be solved by first solving the following problem: for any given y: $0 \leq y \leq r$,

$$\begin{aligned} \min \quad & c_1(b_1) + \cdots + c_I(b_I) \qquad (22) \\ \text{s.t.} \quad & \sum_{j} b_j \geq r + y, \\ & 0 \leq b_i \leq y, \quad i = 1, \ldots, I. \end{aligned}$$

Denote $C(y)$ as the corresponding optimal value. Then, solve the problem $\min_{y \leq r} C(y)$.

To justify that the problem outlined in (22) is equivalent to the problem outlined in (21), let Φ_1 be the set of all feasible solutions $(b_1, b_2, \cdots, b_I)$ to (21) and $\Phi_2(y)$ be the set of all feasible solutions $(b_1, b_2, \cdots, b_I)$ to (22) for parameter y. Also let

$$\Phi_2 = \bigcup_{0 \leq y \leq r} \Phi_2(y)$$

It is straightforward to show that $\Phi_1 = \Phi_2$. Thus optimizing $c_1(b_1) + \cdots + c_I(b_I)$ over Φ_1 is equivalent to optimizing the same function over Φ_2. In particular, an optimal solution for (22) (optimized for all $0 \leq y \leq r$) is also optimal for (21).

B. Convex Costs

Suppose $c_i(.)$ is a convex function, for all $i = 1, ..., n$. Then, given y, (22) is greedily solvable via the following algorithm:

Marginal Allocation:

- Start with $\mathcal{S} := \{1, ..., I\}$ and $b_i = 0$ for all $i \in \mathcal{S}$.
- Each step identify
$$i^* = \arg\min_{i \in \mathcal{S}} \{c_i(b_i + \Delta) - c_i(b_i)\},$$
where $\Delta > 0$ is a pre-specified small increment (depending on required precision), and reset $b_{i^*} \leftarrow b_{i^*} + \Delta$. Whenever $b_i > y - \Delta$, reset $\mathcal{S} \leftarrow \mathcal{S} - \{i\}$.
- Continue until the constraint $\sum_j b_j \geq r + y$ is satisfied.

Note that the complexity of this algorithm is proportional to $n(r+y)/\Delta$. To determine the best y, we can do a line search on $C'(y) = 0$, for $y \in [\frac{r}{I-1}, r]$. (If $y < \frac{r}{I-1}$, then (22) is infeasible.)

If $C(y)$ is convex in y, then $\min_y C(y)$ is itself greedily solvable: we can start with $y = \frac{r}{n-1}$; increase y by a small increment each time, and solve the problem in (22); stop when $C(y)$ ceases to decrease or $y = r$ is reached.

In this algorithm, when we go from one y value to the next, say, $y + \delta$, we do not have to do the marginal allocation that generates $C(y+\delta)$ from scratch (i.e., starting from all x_j values being zero and $\mathcal{S} := \{1, ..., I\}$). We can start from where the previous round of marginal allocation — the one that generates $C(y)$ — first hits a boundary, i.e., $b_j = y$ for some j, and continue from there. Or, if no b_j has hit the boundary in the previous round, then simply start from where the previous round ends (i.e., continue with the solution generated by the previous round). [Recall, $y \in [\frac{r}{I-1}, r]$. As y increases, the number of b_j values that can hit the boundary in the marginal allocation will decrease. Specifically, when $y \in [\frac{r}{k-1}, \frac{r}{k-2}]$, for $k = 3, ..., I$, the number of b_j values that can hit the boundary cannot exceed k, since we have $ky \geq r + y$.]

The convexity of $C(y)$, in turn, is guaranteed when $c_i(\cdot)$'s are convex functions. To see this, let $(b_i(y))_{i=1}^I$ denote the optimal solution to the problem in (22), and consider two such problems, corresponding to $y = y_1$ and $y = y_2$, respectively. For any $\alpha \in (0, 1)$, we have

$$\begin{aligned} & \alpha C(y_1) + (1-\alpha) C(y_2) \\ = \quad & \alpha \sum_j c_j(b_j(y_1)) + (1-\alpha) \sum_j c_j(b_j(y_2)) \\ \geq \quad & \sum_j c_j(\alpha b_j(y_1) + (1-\alpha) b_j(y_2)), \end{aligned}$$

where the inequality follows from the convexity of the c_j's. Next, consider a third version of (22), with $y = \alpha y_1 + (1-\alpha) y_2$. It is straightforward to verify that $\alpha b_j(y_1) + (1-\alpha) b_j(y_2)$, $j = 1, ..., I$, is a feasible solution to this problem. Therefore, we have

$$\sum_j c_j(\alpha b_j(y_1) + (1-\alpha) b_j(y_2)) \geq C(y),$$

and hence

$$\alpha C(y_1) + (1-\alpha) C(y_2) \geq C(y) = C(\alpha y_1 + (1-\alpha) y_2).$$

That is, $C(y)$ is a convex function. To summarize, we have

Proposition 2: Suppose for each $i = 1, ..., I$, $c_i(\cdot)$ is a convex function. Then, the optimal value in (22), $C(y)$, is convex in y. In this case, the streaming problem in (21) is greedily solvable: each step increase y by a small increment (starting from $y = \frac{r}{I-1}$), and apply the marginal allocation algorithm to generate $C(y)$; stop when $C(y)$ ceases to decrease or $y = r$ is reached.

V. Conclusion

We have formulated and solved several peer selection problems that arise in a free-market peer-resource economy. We have examined two delivery models, namely, downloading and streaming. For both models, the selected servers transfer the object to the client in parallel. For the download problem, the total download time is the maximum download time from each of the selected servers. This gives rise to a min-max optimization problem, for which we obtain an explicit solution. For the streaming problem, we examine optimal rate partitioning for which continuous playback is ensured even if one server fails.

References

[1] D.A. Turner and K.W. Ross, *"A LightWeight Currency Paradigm for the P2P Resource Market"* submitted.

[2] D.A. Turner and K.W. Ross, *"The Lightweight Currency Protocol,"* Internet Draft, September 2003.

[3] E. Adar and B.A. Huberman, *"Free Riding on Gnutella,"* First Monday, 2000.

[4] S. Saroiu, P. K. Gummadi, and S. D. Gribble, *"A measurement study of peer-to-peer file sharing systems,"* Proceedings of Multimedia Computing and Networking, Jan 2002.

[5] J. Kubiatowicz, D. Bindel, Y. Chen, S. Czerwinski, P. Eaton, D. Geels, R. Gummadi, S. Rhea, H. Weatherspoon, W. Weimer, C. Wells, and B. Zhao, *"Oceanstore: An architecture for global-scale persistent storage,"* ASPLOS, 2000.

[6] F. Dabek, M. F. Kaashoek, D. Karger, R. Morris, and I. Stoica, *"Wide-area cooperative storage with CFS,"* SOSP '01.

[7] R. Krishnan, M.D. Smith, and R. Telang, *"The Economics of Peer-to-Peer Networks,"* Working Paper, Carnegie Mellon University.

[8] P. Golle, K.Leyton-Brown, and I.Mironov, *"Incentives for Sharing in Peer-to-Peer Networks,"* ACM Conference on Electronic Commerce, 2002.

[9] S. D. Kamvar, M. T. Schlosser, H. Garcia-Molina, *"The EigenTrust Algorithm for Reputation Management in P2P Networks,"* In Proceedings of the Twelfth International World Wide Web Conference, 2003.

[10] R. Dingledine, N. Mathewson, P. Syverson, *"Reputation in P2P Anonymity Systems,"* Workshop on Economics of Peer-to-Peer Systems, 2003.

[11] K. Ranganathan, M. Ripeanu, A. Sarin, I. Foster, *"To Share or not to Share' An Analysis of Incentives to Contribute in File Sharing Environments,"* International Workshop on Economics of Peer to Peer Systems, 2003.

[12] A. Rowstron and P. Druschel, *"Storage management and caching in past, a large-scale, persistent peer-to-peer storage utility,"* Proceedings of SOSP 2001.

[13] *"The Grid: Blueprint for a Future Computing Infrastructure,"* Morgan Kaufmann Publishers, USA, 1999.

[14] S. G. M. Koo, C. Rosenberg, D. Xu, *"Analysis of Parallel Downloading for Large File Distribution,"* FTDCS 2003.