A Reciprocation-Based Economy for Multiple Services in Peer-to-Peer Grids

Miranda Mowbray†, Francisco Brasileiro‡, Nazareno Andrade‡, Jaindson Santana‡, Walfredo Cirne‡

†Hewlett-Packard Laboratories Bristol
Filton Rd, Stoke Gifford, Bristol BS34 8QZ, UK

‡Universidade Federal de Campina Grande
Av. Aprígio Veloso, s/n - 58.109-970, Campina Grande, Brazil

miranda.mowbray@hp.com {fubica,nazareno,jaindson,walfredo}@dsc.ufcg.edu.br

Abstract

Peer-to-peer grids aim to build computational grids encompassing thousands of sites. To achieve this scale, such systems cannot rely on trust or off-line negotiations among participants. Moreover, without incentives for donation, there is a danger that free riding will prevail, leading the grid to collapse. Reciprocation-based incentive mechanisms that use no information provided by untrustworthy peers have been proposed to deal with this problem, and marginalize free riders. However, they have only been studied for the case in which a single service is shared. In this paper we study reciprocation-based incentive mechanism for peer-to-peer grids in which multiple services, such as processing power and data transfers, are shared explicitly. We have modeled such a system and established how peers should assess whether it is profitable to exchange services with another peer (under the assumption that the other peer is not a free rider), an issue that is not present in the single service case. Unfortunately, to apply such a function a peer must rely on information provided by untrustworthy peers on how they value services. As an alternative, we have extended, to the case of multiple services, a reciprocation-based mechanism which uses only reliable information gathered locally. We have assessed this mechanism by simulating scenarios in which services are exchanged that are combinations of two different basic services. In our scenarios the mechanism performs very well, and can marginalize free riders even when the cost to peers of donating a service is nearly as large as the utility gained by receiving it.

1 Introduction

A computational grid is a federation of sites across different administrative domains which shares computational services. A peer-to-peer grid is a large-scale, free-to-join computational grid in which participants do not necessarily trust each other. In a peer-to-peer grid, each peer represents a site; whenever the site has idle capacity it provides services to other peers, and whenever the site needs more computational power, it requests services from all other peers. Examples of projects building such grids are OurGrid [9], Cluster Computing On the Fly [15] and the Self-organizing Flock of Condors proposed by Butt et al. [7].

Peers gain utility from using the services of other peers, and there is a cost for providing a service to the grid. This cost is incurred to maintain the hardware and software needed by a site to provide services to the grid, and through the security risk posed to this site when providing such services to non-trusted parties. These costs must be low in comparison with the benefit obtained by joining the grid, and they can be lowered by reducing the effort needed to install and maintain the grid middleware, as well as by improvements to security mechanisms. However, we argue it is naive to assume that all costs, and in particular the security risks, can be eliminated altogether.

In contrast to most grids currently in production, a peer-to-peer grid cannot rely on off-line negotiations or trust chains comprising all participants in the grid to enforce cooperative behavior. In this setting, an incentive mechanism plays a key role in promoting service provision to the grid. If there is a non-zero cost for donating services and a peer can obtain the same amount of service no matter how much it serves other peers, then peers have an economic incentive to contribute nothing and free ride. This behavior is indeed found in peer-to-peer file sharing systems [2, 13]. Free riding reduces the amount of service available in the grid, and diminishes the utility of the system for users of resource-intensive applications; most users of computational grids run such applications. This observation motivates the use of an incentive mechanism to promote collaboration in peer-to-peer grids.

Andrade et al. have proposed the Network of Favors, a reciprocation-based mechanism for peer-to-peer grids in which a single service is shared [3, 4]. In the Network of Favors, peers exchange donations of services. Peers always donate their spare service, and decide whom to serve based solely on the record of their past bilateral interactions with peers requesting the service.

It has been shown that in a peer-to-peer grid sharing a single service – access to processing power – this autonomous behavior provides an incentive for peers to contribute as much as possible [3, 4]. This happens because peers who contribute more get more in return when they make requests. Since the balance of past interactions is very little information, a peer can keep track of its interactions with a very large number of other peers. The Network of Favors is currently implemented in a peer-to-peer grid named OurGrid [16], which is in production since December 2004 (see status.ourgrid.org).

The Network of Favors is particularly suited for peer-to-peer grids because, by keeping the behavior of each peer completely autonomous, it can be implemented without depending on any centralized mechanism or trust infrastructure. However, the mechanism is not suitable for scenarios where the value of a service changes over time or where peers would like to express the priority of the work they need done.

In this paper we consider how to create a reciprocation-based economy grounded on the Network of Favors in which peers provide multiple services to the grid. Our main motivation comes from practical experience with the deployment of a peer-to-peer grid. Users of this grid have manifested the need for the system to consider incentives not only for the provision of processing power, but also for data transfers and storage. This happens because it is not possible to abstract all services as a single one, as different users value these services differently. Some users run data-intensive applications and compete for the capacity of other peers to receive data, while other users run applications with only light data requirements.

Introducing multiple services in an economy complicates matters because in such a setting, it is in the interest of peers to choose trading partners not only based on their behavior (ie. the likelihood of reciprocation), but also on the way these peers value the services they provide and consume.

We study the problem posed by this requirement to identify what is necessary to extend the prioritization model of the Network of Favors to a system sharing an arbitrary number of services, so that we can implement such an extension in our peer-to-peer grid. We believe our results might also be useful when extending other systems that use similar exchange-based mechanisms, such as BitTorrent [10].

The structure of this paper is as follows. We situate our work among related efforts in Section 2. In Section 3 we discuss how the Network of Favors should be extended to deal with multiple services. In particular, we discuss the issue that it may be unprofitable for two peers who are not free riders to exchange services, an issue that is not present in the single service case. Then, in Section 4, we model the problem and discuss how peers can implement a function to assess whether it is profitable or not to interact with another peer, under the assumption that the other peer is not a free rider. As we will show, it turns out that to implement such a function a peer must rely on information provided by untrustworthy peers. Since this is not desirable, we consider an alternative, in which interactions with unprofitable peers may occur, but a peer donates services preferentially to peers with whom it expects to have beneficial interactions in the future, based on its direct past experience. In Section 5 we present simulation results under several scenarios for the exchange of services that are combinations of two different basic services - for instance, the basic services might be processing power and storage. In our simulations the mechanism using only local information performs very well, and can marginalize free riders even when the costs to peers of donating services are nearly as large as the utility gained by receiving them. We present our conclusion in Section 6.

2 Related Work

There is a body of research on using market-based mechanisms to regulate service provision in resource sharing systems [1, 5, 8, 14, 19]. Markets can provide incentives for provision in a flexible and robust way. However, we argue that market-based mechanisms are not suitable for the scenario of peer-to-peer grids that we have described. Participants in peer-to-peer grids do not trust each other, and there is no centralized or widely trusted entity in the system. Market-based resource allocation mechanisms rely on contracts, auditing, banking and electronic cash payment systems, which are very difficult to deploy with such constraints on the system. Shneidman et al. [17] discuss other open issues in getting market-based resource allocation into production.

Andrade et al. [3, 4] originally proposed the Network of Favors as an alternative to market-based mechanisms. The Network of Favors is designed not to depend on the existence of banking, trust or negotiation among peers. This mechanism is similar to the tit-for-tat mechanism used in BitTorrent, where peers exchange pieces of a file based on their past bilateral interactions [10]. However, neither of these works apply to the case in which multiple services are exchanged.

It has been suggested, as part of the vision of utility computing, that all computing services (including processing power, memory, disc storage and bandwidth) should be described in terms of a single unit, the *computon*, just as electricity is sold by the kilowatt-hour. The price for a computon would vary according to supply and demand [18]. This would reduce the economic problem of the provision of multiple computing services to the provision of a single service, measured in computons. However, attempts to produce an agreement between the few largest computing suppliers of exactly how the computon should be defined (for instance, how many bytes of disc storage should be the equivalent in computons of a second of processing power on a reference machine) have not been successful [12]. Agreement between suppliers on how to define the computon is even more problematic in a peer-to-peer grid, because in a peer-to-peer grid all peers are potentially service suppliers, and services that are hard for some peers to produce will in general be easily produced by others.

3 The Network of Favors for Multiple Services

The Network of Favors has been previously explored in a system where a single service is shared [3, 4]. In this paper we discuss how it can be extended to a system in which peers may provide more than one service and may differ in how they value services.

The basic idea of the Network of Favors is that peers prioritize the requests they receive based solely on the record of their past interactions with the requesters. As a result, there is no need to trust other peers or a central entity in order to assess the global reputation of each requester.

The extension presented here is designed for a system composed of a number of peers, where each peer owns a set of resources and can provide multiple services with them. All peers alternate independently between periods where they have spare resources, and periods where they have demand for services that cannot be immediately met by their resources. We call a peer that currently has a spare resource a *provider*. As in the original Network of Favors, the work done by a provider for another peer is called a *favor*, which in the general case may be any combination of the services available in the system.

When there are multiple services, a favor of one type of service may be repaid in another type of service, and peers may value services differently. The main design challenge this imposes for the mechanism is the need for peers to choose with which other peers they should interact. Providers need to decide both with which other peers it is worthwhile to interact in the long run, and which of the current requests for their services they should prioritize. They do so using a long-term and a short-term policy, respectively.

The use of the long-term policy allows peers to protect their overall utility, by not donating to another peer if they assess that the expected effect on their overall utility resulting from a long-term interaction with this peer is unsatisfactory.

The short-term policy governs to whom a provider decides to donate a favor, when there are several other peers which are not excluded by the long-term policy and which are currently requesting favors. This decision is based on information from interactions with these peers in the past. Whenever there is no contention for the services of a provider, it donates these services to any peer requesting them that is not excluded by the long-term policy. This serves as a bootstrap for the exchange of favors.

4 Incentive Mechanisms

In this section we describe two incentive mechanisms, by giving the details for each of a long-term and a short-term policy by which peers calculate which peer to donate a favor to. As we will show, the effect of these policies is that there is an incentive for donation.

For the first incentive mechanism, which we call *PI* for *positive interactions*, the long-term policy forbids interactions between pairs of peers unless their long-term interaction should be profitable, under the assumption that neither are free riders; and the short-term policy makes a peer donate services preferentially to peers with whom it expects to have beneficial interactions in the future, based on its past interactions and on knowledge of how the other peers value services. It turns out that knowledge of how other peers value services is also necessary for the long-term policy for *PI*.

The second incentive mechanism, which we call *xNoF* for *extended Network of Favors*, does not require this knowledge; it relies only on peers' direct experiences and knowledge of how they themselves value services, and thus is closer in spirit to the Network of Favors. Its long-term policy is the trivial policy allowing interaction with all peers, so for reciprocation it relies on its short-term policy, under which a peer donate services preferentially to peers with whom it expects to have beneficial interactions in the future, based on its past interactions and its own (but not others') valuation of services.

Our experience is that in practice it can be difficult for peers in a peer-to-peer grid to predict the utility that they, or others, would gain by being donated a particular favor at a point in the future. We therefore do not assume that peers can estimate these utilities. However, costs are easier for peers to predict. We assume that peers can easily determine the costs that they would incur when donating a favor. These costs will in general vary between different peers, and for different types of favors.

4.1 Notation and assumptions

Here are some general assumptions that we make about the system.

G1. The system is a peer-to-peer system in which peers independently decide whether or not to do favors requested by other peers. Peers are content to participate in the system if their expected future net utility gain as a result of being in the system (that is, utility gain from being donated favors minus utility loss from the cost of donating favors to others) is positive.

G2. Each peer A can accurately estimate the utility cost $v_A(f)$ that it would incur if it provided favor f for another peer. Costs are additive, i.e. the cost of providing favor $f1$ and then favor $f2$ is $v_A(f1) + v_A(f2)$. For all non-zero favors f, the utility cost $v_A(f)$ is positive.

G3. The utility to a peer A of receiving a favor f, written $u_A(f)$, may vary over time, but always satisfies $u_A(f) > v_A(f)$. That is, the utility to A of receiving a favor that it requests is greater than the cost to A of donating the same favor to another peer.

G4. If a peer A tries to pay a favor to peer B, it will eventually succeed in doing so. This implies that eventually A will be able to provide a service at a time that it is requested by B, and that the granularity of requests for services by B can be made small enough that a request is never too large for A to ever satisfy.

G5. Some peers in the system are *collaborative*, and follow the algorithm specified. However, some are *non-collaborative*, and choose behavioral strategies in order to maximize their expected net utility gain. In particular, one strategy that they may consider is *free riding*, that is, requesting and consuming favors from the system but donating no favors to the system. They may spread false information about other peers, or about themselves, and may conspire with other non-collaborative peers to make their falsehoods more plausible. We assume there is a low turnover of collaborative peers in the system, but that there may be a very high turnover of non-collaborative peers.

4.2 The long-term policy for PI

Let $\overline{f}_A$ be the average favor that a peer A requests to the other peers. This represents a probability distribution of the types of favors requested by A over the long term. For instance, if in a typical time interval A requests on average one unit of basic service s_1 and three units of basic service s_2, then $\overline{f}_A$ will be $(s_1 + 3s_2)/4$.

Now, suppose peer A is deciding whether or not to interact with B. For a long-term interaction in which A donates $n.\overline{f}_B$ to B and B donates $m.\overline{f}_A$ to A for some m, n to be beneficial to both A and B, it needs to be the case that both $m \cdot u_A(\overline{f}_A) - n \cdot v_A(\overline{f}_B)$ and $n \cdot u_B(\overline{f}_B) - m \cdot v_B(\overline{f}_A)$ are positive. Such positive real numbers n, m exist if and only if

$$u_A(\overline{f}_A) \cdot u_B(\overline{f}_B) > v_A(\overline{f}_B) \cdot v_B(\overline{f}_A) \qquad (1)$$

It is difficult for A to estimate the functions u_A or u_B, but, by assumptions G2 and I2, A knows the functions v_A and v_B, and by assumption G3 $u_A(f_A) > v_A(f_A)$ and $u_B(f_B) > v_B(f_B)$. Therefore the sensible thing for A to do is to check whether the following more stringent inequality holds:

$$v_A(\overline{f}_A) \cdot v_B(\overline{f}_B) \geq v_A(\overline{f}_B) \cdot v_B(\overline{f}_A) \qquad (2)$$

Roughly speaking, this inequality means that the cost to the pair of peers A and B of producing the average favors that they request themselves is greater than that of producing the requested favors for each other. Note that if $\overline{f}_A = \overline{f}_B$ or $v_A = v_B$, the inequality is automatically satisfied. In particular, in the special case that all requested favors are multiples of a single service, then all pairs of peers can gain by interacting.

For pairs of peers A, B for which inequality (2) holds, both peers know that they can benefit from a long-term exchange of favors, and so both have an incentive to initiate such an exchange of favors. When A has spare resources it will look for peers B requesting favors for which this inequality is satisfied, and use the spare resources to grant one of these favors.

Pairs of peers for which (2) does not hold cannot tell whether or not it is possible for them both to benefit from a long-term exchange of favors. They do not interact, so as to avoid being drawn into an interaction which causes them to lose utility.

4.3 The short-term policy for PI

Now suppose that there is more than one peer B requesting favors with which the long-term decision function of a provider A does not prevent interactions. How does A decide which of these peers to donate to? The answer is that A

donates to whichever of these peers it expects to gain most from interacting with; A calculates this using its short-term policy.

The short-term policy uses the *cost balance* from the previous pair-wise interactions between peers. Each peer keeps a record, for each other peer with which it interacts, of this number, which is calculated as follows. Before the two peers have interacted the cost balance is equal to zero (although this value is not explicitly recorded). If peer A donates a favor f to peer B, then A decreases its cost balance for interactions with B by the cost $v_A(f)$ of producing this favor - or, if the original cost balance was less than $v_A(f)$, peer A sets its cost balance for B to zero. Meanwhile, B increases its cost balance for A by $v_A(f)$. Note that different peers will in general record different cost balances for the same peer A.

When choosing which peer to donate a favor to, provider A donates the favor to the candidate peer with the highest cost balance, where the candidate peers are the peers requesting the favor for whom inequality (2) holds.

4.4 The policies for $xNoF$

An obvious caveat of the way the policies for PI are calculated is that they require knowledge of the average favors requested by other peers ($\overline{f}_A$), and also the costs that other peers would incur when donating a favor ($v_A(f)$). Other peers are not necessarily trustworthy, and may be able to increase their expected utilities by giving false information about these. In particular, in the absence of reliable information on other peers' costs, it is not feasible to use a long-term policy which distinguishes profitable collaborative peers from unprofitable ones. The alternative used by the incentive mechanism $xNoF$ is to remove this check, or equivalently to use a trivial long-term decision function that never prevents two peers from interacting, and rely simply on the short-term function to marginalize both free riders and unprofitable collaborative peers.

In addition, the way the cost balance accounts for favors received should also be changed. The approach that we use for $xNoF$ is instead of a peer using the unknown cost function of a favor's provider to calculate the cost balance, it uses its own cost function. Thus, the short-term policy for $xNoF$ differs from that of PI in that when B receives favor f from A, B increases its cost balance for A by $v_B(f)$ rather than by $v_A(f)$.

4.5 Consequences of the policies

Whether the cost balance is calculated in the way specified by the short-term policy for *PI* or by the short-term policy for *xNoF*, the cost balance that A records for B decreases when A donates favors to B (provided that it was not zero to start with) and increases when B donates favors to A. It can therefore be thought of as an indication of the net benefit that A has gained so far by interacting with B. By donating the favor to the candidate peer with the highest value of the cost balance, provider A is choosing to interact with a peer with whom it expects to have beneficial interactions in the future, based on its past experience. A similar way of selecting peers to interact with is described (for a different context) in [6].

The cost balance is greater than or equal to zero for all peers. If it is zero for all candidate peers, A still donates the service to one of the candidate peers. This allows newcomers to the system to have a chance of donating and receiving favors. We do not require any other bootstrap mechanisms to deal with newcomers.

Free riders will sometimes be donated favors, and A will lose utility as a result of any donations that A makes to a free rider. However, A will not donate a favor to a free rider unless all the candidate peers have zero cost balance. On the other hand, the way that the cost balance is calculated has the effect that if collaborative peer A ever donates a favor to collaborative peer B, then throughout the subsequent history of the system either A's cost balance for B will be positive, or B's cost balance for A will be positive, or both will be positive. Thus the system promotes continuing interactions between collaborative peers.

Since donating a favor to a collaborative peer increases the expected amount of favors received from that peer in the future, over the long run the more a peer donates to the system the more it can expect to receive back.

Free riders receive services with low priority. Thus the expected long-term utility gain for free riders should be lower than the expected utility gain of peers that do not free ride, so that non-collaborative peers will choose not to free ride.

Since we do not assume that there is a limit on the number of different IDs that a peer can use in the system, non-collaborative peers can whitewash their identities by leaving the system and rejoining it under a different ID. This has the effect that non-collaborative peers become indistinguishable from collaborative newcomers. However, they cannot increase their chances of obtaining services from any peer A by doing so, because A's cost balance for their new identity will be zero, the minimum value. Peers can only increase their priority for donations from others by making donations to others themselves.

Another consequence of peers' potential ability to use multiple identities is that it is particularly difficult to design a reliable global reputation system, because a non-collaborative peer has the possibility of creating very many clones of itself which propagate false reputations. Our incentive for donation does not rely on a global reputation system.

5 Evaluating the Network of Favors for Multiple Services

In this section we evaluate the performance of the incentive mechanisms proposed in the previous section. Our evaluation is based on simulations of the two settings discussed. Before analyzing the results attained from the simulations, we present the model implemented by our simulator, the metric measured and the scenarios simulated.

5.1 System Model

We consider a grid comprised of N peers which offer and consume services whose resource requirements are combinations of two different basic services, s_1 and s_2. For instance, the two basic services might be processing power and storage. Peers can either be collaborative or non-collaborative. In our simulations, the timeline is in turns, and at each turn a peer can be either in consuming or in non-consuming state. When in non-consuming state, collaborative peers donate the use of their spare resources, while non-collaborative peers go idle. In addition to not donating any services, we assume that non-collaborative peers change their identities every time they interact with another peer. The design parameters that we consider for the system are:

- Frequency of consumption. We assume that at a given turn each peer has an independent probability ρ of being in consuming state.
- Service availability. $D = (d_1, d_2)$, where d_1 and d_2 are, respectively, the maximum amount of services s_1 and s_2 that a peer is able to donate in a given turn.
- Relative favor consumption profile. For a peer A, π_A is the proportion of service s_2, in relation to s_1, in a favor that A requests. It follows that $\overline{f_A}$ is a multiple of $s_1 + \pi_A.s_2$.
- Relative cost of donation. For a peer A, κ_A measures how costly it is for A to provide a unit of basic service s_2 relative to providing a unit of s_1. A favor f that corresponds to x units of service s_1 and y units of service s_2 costs $v_A(f) = x + y \cdot \kappa_A$.
- Percentage of free riders. ϕ is the percentage of peers that free ride and change their identities. The other peers collaborate.
- Incentive mechanism. The incentive mechanism is either PI or $xNoF$. The long-term and short-term policies used are those specified in Section 4.

We assume that peers in consuming mode are able to consume as much service as the providers are able to offer them, restricted only by the consumption profiles of the consuming peers. Our practical experience has shown us that this is generally the case in a computational grid. In each turn providers are selected randomly and donate as much as possible of their available services, until all providers have been selected. Each provider performs the following steps: (i) uses the long-term policy to select among all peers in consuming state which are the peers with whom it may interact; (ii) uses the short-term policy to select a consuming peer to donate its services; (iii) donates as much service as it can to the selected peer; and, (iv) updates the corresponding cost balance. Collaborative consuming peers also update their corresponding cost balance.

5.2 Performance metric

The performance metric we use to evaluate the system, named M (for *metric*) is designed by considering the special cases where, for some Ω, utilities are given by:

$$u_A(f) = \Omega \cdot v_A(f) \text{ for all peers } A \text{ and favors } f \qquad (3)$$

By assumption G3 in Subsection 4.1, Ω must be greater than 1.

The metric M can be used to judge how much advantage the system gives to collaborative peers. It is the minimum value of Ω such that if (3) holds, then an average free rider gains no more from being in the system than an average collaborative peer does. Since Ω is known to be > 1, it follows that if M is close to 1, then the system is very effective at marginalizing free riders.

Let $f_{rec}(A)$ and $f_{prov}(A)$ be the accumulated amount of favors that a peer A has received and provided, respectively. Also, let $\mathcal{C}$ be the set of collaborative peers and $\mathcal{F}$ the set of non-collaborative ones. Then,

$$M = \frac{\sum_{C \in \mathcal{C}} v_C(f_{prov}(C))}{\sum_{C \in \mathcal{C}} v_C(f_{rec}(C)) - (1/\phi - 1) \cdot \sum_{F \in \mathcal{F}} v_F(f_{rec}(F))}.$$

5.3 Scenarios

We chose scenarios in which the parameter values satisfy $D = (10, 10)$; $\rho \in \{0.1, 0.5, 0.9\}$; and, $\phi \in \{0.25, 0.5, 0.75\}$. We chose these parameter values to cover a variety of scenarios, to include both low and high realistic values. Moreover, we chose the values κ_A and π_A (with A ranging over all the peers) in three different ways, so as to create three different sets of pairs of mutually profitable collaborative peers: (i) the interactions between any two collaborative peers are mutually profitable (single-set); (ii) there are two disjoint sets of collaborative peers with the same cardinality, such that interactions between collaborative peers of the same set are mutually profitable, while

Table 1. Summary of simulation results

Scenario	ρ	ϕ	M for PI	M for $xNoF$
single-set	0.1	0.25	1.02	1.02
mutex	0.1	0.25	1.04	1.03
three-set	0.1	0.25	1.03	1.02
single-set	0.1	0.50	1.04	1.04
mutex	**0.1**	**0.50**	**1.07**	**1.04**
three-set	0.1	0.50	1.04	1.04
single-set	0.1	0.75	1.15	1.16
mutex	**0.1**	**0.75**	**1.22**	**1.09**
three-set	0.1	0.75	1.08	1.07
single-set	0.5	0.25	1.01	1.01
mutex	0.5	0.25	1.02	1.01
three-set	0.5	0.25	1.01	1.01
single-set	0.5	0.50	1.01	1.01
mutex	0.5	0.50	1.02	1.01
three-set	0.5	0.50	1.01	1.01
single-set	0.5	0.75	1.02	1.02
mutex	0.5	0.75	1.04	1.03
three-set	0.5	0.75	1.02	1.02
single-set	0.9	0.25	1.02	1.02
mutex	0.9	0.25	1.04	1.02
three-set	0.9	0.25	1.02	1.02
single-set	0.9	0.50	1.02	1.03
mutex	0.9	0.50	1.05	1.03
three-set	0.9	0.50	1.03	1.03
single-set	0.9	0.75	1.04	1.05
mutex	0.9	0.75	1.08	1.06
three-set	0.9	0.75	1.06	1.06

interactions of collaborative peers in different sets are non-profitable (mutex); and, (iii) there are three sets of peers, such that within each set collaborative peers have mutually profitable interactions and collaborative peers in the one of the sets have mutually profitable interactions with collaborative peers in all sets; all other interactions are non-profitable (three-set). For each of these scenarios, the non-collaborative peers are uniformly distributed over all the sets.

We simulated all these scenarios both for the incentive mechanism PI and for the incentive mechanism $xNoF$.

5.4 Analysis

We have run enough simulations to reduce the error to 1% with a 95% confidence interval. Table 5.4 summarizes the results attained for *Adv* in all settings and scenarios simulated.

As can be seen, for most cases the performance of PI and $xNoF$ is similar, and there are no cases in which the performance of PI is better than the performance of $xNoF$ by more than the 1% margin of error. However, for two cases $xNoF$ is better than PI by 3% or more (see the entries in bold in Table 5.4). In these cases there are always very few peers in consuming mode, and many peers are free riders. Thus, in many turns all the profitable peers in consuming mode will be free riders. The incentive mechanism $xNoF$ works better than PI in this situation because on such turns a provider using $xNoF$ may donate to an unprofitable collaborative peer rather than to a free rider. In some cases it is less costly provide to service to an unprofitable collaborative peer than to a free rider, because an unprofitable collaborative peer may provide *some* service in return, whereas a free rider will not.

More importantly, in most cases the value of M for both IP and $xNoF$ is very close to 1, and in all cases it is less than 1.25. If utilities are given by (3), this implies that there is an incentive for non-collaborative peers to change their strategy from free riding to providing services, provided that the cost to a peer of donating a favor is less than four-fifths of the utility gained by the peer if it receives the same favor. In practice, the cost of providing a service in a peer-to-peer grid is typically small compared to the utility gained. The simulation scenarios in [4], which were chosen to be realistic based on practical experience with a system running the Network of Favors, satisfy (3) and have the cost of donation equal to at most two-fifths of the utility gained.

In our design of $xNoF$, we dealt with the problem of unreliable second-hand information by avoiding the use of such information altogether. A different approach to unreliable information in peer-to-peer systems is to use majority voting by peers to ascertain which information is correct (see e.g. [6, 11]). However, this will fail if less than half the peers are collaborative, because the non-collaborative peers can collude to outvote peers that provide truthful information. In contrast, note that the incentive mechanisms presented here perform well even in the cases for which $\phi = 0.75$: in these cases only a quarter of the peers are collaborative.

6 Conclusions

In this paper we have designed and evaluated mechanisms for promoting service provision in a peer-to-peer grid in which several services are available. We have extended the Network of Favors, which is a mechanism to provide incentives using a simple autonomous behavior of the peers in the system. Simulation results show that the extended Network of Favors has an excellent performance, even when the costs of providing services are high and fewer than half the peers are collaborative.

References

[1] D. Abramson, R. Buyya, and J. Giddy. A Computational Economy for Grid Computing and its Implementation in the Nimrod-G Resource Broker. *Future Generation Computer Systems J.*, 18(8):1061–1074, 2002.

[2] E. Adar and B. A. Huberman. Free Riding on Gnutella. *First Monday*, 5(10), Oct. 2000.

[3] N. Andrade, F. V. Brasileiro, W. Cirne, and M. Mowbray. Discouraging Free Riding in a Peer-to-Peer CPU-sharing Grid. In *Proc. 13th IEEE Symposium on High Performance Distributed Computing, Honolulu, Hawaii*, pages 129–137, June 2004.

[4] N. Andrade, M. Mowbray, W. Cirne, and F. V. Brasileiro. When Can an Autonomous Reputation Scheme Discourage Free-riding in a Peer-to-Peer System? In *Proc. 4th Workshop on Global and Peer-to-Peer Computing (GP2PC 04), Chicago, USA*, Apr. 2004.

[5] A. AuYoung, B. N. Chun, A. C. Snoeren, and A. Vahda. Resource allocation in federated distributed computing infrastructures. In *1st Workshop on Operating System and Architectural Support for the On-demand IT InfraStructure (OASIS'04), Boston, MA*, Oct. 2004.

[6] D. Banerjee, S. Saha, S. Sen, and P. Dasgupta. Reciprocal resource sharing in P2P environments. In *4th International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS'05), Utrecht, The Netherlands*, pages 853–869, 2005.

[7] A. Butt, R. Zhang, and Y. Hu. A Self-Organizing Flock of Condors. In *Proc. Supercomputing 2003, Phoneix, Arizona, USA*, Nov. 2003.

[8] R. Buyya and S. Vazhkudai. Compute Power Market: towards a market-oriented grid. In *First IEEE/ACM Int'l Symp. Cluster Computing and the Grid, Brisbane, Australia*, pages 574–581. IEEE Computer Society Press, May 2001.

[9] W. Cirne, F. V. Brasileiro, J. Sauvé, N. Andrade, D. Paranhos, E. L. Santos-Neto, R. Medeiros, and F. Silva. Grid computing for Bag-of-Tasks applications. In *Proc. 3rd IFIP Conference on E-Commerce, E-Business and E-Covernment*, Sept. 2003.

[10] B. Cohen. Incentives Build Robustness in BitTorrent. In *Proc. Workshop on Economics of Peer-to-Peer Systems*, Berkeley, CA, USA, June 2003.

[11] E. Damiani, S. D. C. di Vimercati, S. Parabosci, and P. Samaranti. Managing and Sharing Servents' Reputations in P2P Systems. *IEEE Trans. Data and Knowledge Engineering*, 15(4):840–854, July/Aug. 2003.

[12] M. Hall. Utility pricing. *Computerworld, 21 November*, 2005.

[13] D. Hughes, G. Coulson, and J. Walkerdine. Freeriding on gnutella Revisited: the bell tolls. *IEEE Distributed Systems Online*, 6(6), Feb. 2005.

[14] K. Lai, B. A. Huberman, and L. Fine. Tycoon: A distributed market-based resource allocation system. Technical Report cs.DC/0404013, HP Labs, Apr. 2004. http://arxiv.org/abs/cs.DC/0404013.

[15] V. M. Lo, D. Zappala, D. Zhou, Y. Liu, and S. Zhao. Cluster Computing on the Fly: P2P Scheduling of Idle Cycles in the Internet. In *Peer-to-Peer Systems III: Third International Workshop, IPTPS 2004*, volume 3279 of *Lecture Notes in Computer Science*, pages 227–236, La Jolla, CA, USA, Feb. 2004. Springer Berlin / Heidelberg.

[16] OurGrid Project. OurGrid site. http://www.ourgrid.org/, 2003–2006.

[17] J. Shneidman, C. Ng, D. Parker, A. AuYoung, A. C. Snoeren, and A. Vahda. Why markets could (but don't currently) solve resource allocation problems in systems. In *10th USENIX Workshop on Hot Topics in Operating System*, June 2005.

[18] The Economist. Who wants to buy a computon? 12 march, 2005.

[19] R. Wolski, J. Plank, and J. Brevik. G-commerce: Market formulations controlling resource allocation on the computational grid. Technical report ut-cs-00-450, University of Tennessee at Knoxville, Oct. 2000.