Reciprocal Resource Sharing in P2P Environments

Dipyaman Banerjee, Sabyasachi Saha, Sandip Sen
Department of Math & Computer Science
The University of Tulsa
{dipu,saby,sandip}@utulsa.edu

Prithviraj Dasgupta
Computer Science Department
University of Nebraska, Omaha
pdasgupta@mail.unomaha.edu

ABSTRACT

Peer-to-peer (P2P) systems enable users to share resources in a networked environment without worrying about issues such as scalability and load balancing. Unlike exchange of goods in a traditional market, resource exchange in P2P networks does not involve monetary transactions. This makes P2P systems vulnerable to problems including the free-rider problem that enables users to acquire resources without contributing anything, collusion between groups of users to incorrectly promote or malign other users, and zero-cost identity that enables nodes to obliterate unfavorable history without incurring any expenditure. Previous research addresses these issues using user-reputation, referrals, and shared history based techniques. Here, we describe a multi-agent based reciprocity mechanism where each user's agent makes the decision to share a resource with a requesting user based on the amount of resources previously provided by the requesting user to the providing user and globally in the system. A robust reputation mechanism is proposed to avoid the differential exploitations by the free-riders and to prevent collusion. Experimental results on a simulated P2P network addresses the problems identified above and shows that users adopting the reciprocative mechanism outperform users that do not share resources in the P2P network. Hence, our proposed reciprocative mechanism effectively suppresses free-riding.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intel ligence

General Terms

Experimentation, Performance

Keywords

Peer-to-peer, Cooperation, reciprocity, agents, free-riding

1. INTRODUCTION

The number of users accessing the Internet over the last few years has multiplied and this has encouraged online interaction between users using different communication models. The recent popularity of file sharing systems such as Napster, Gnutella and SETI@home indicate the P2P paradigm as a major medium for users to interact with each other for real-time collaboration and information sharing in a large-scale, distributed environment. The P2P model offers several design and implementation challenges. A node in a P2P network needs appropriate techniques for determining peers that possesses useful resources, algorithms for trading those resources between the peers and a trust model for sharing resources. In this paper, we concentrate on the latter problem and describe a multi-agent based reciprocative mechanism to address this problem.

A P2P system consists of nodes in a distributed network environment that are capable of sharing resources. The distributed nature of P2P systems enables thousands of nodes to interact with each other without problems of scalability or load balancing. However, in the absence of a central server node that maintains information about the nodes participating in the P2P network, users can obtain free entry to the system and also maintain no-cost identities in the system. Another challenge in P2P systems is the bottleneck provided by the *free rider* problem. Most users in the systems want to obtain resources from other nodes in the network without themselves ever contributing (sharing) any resources to other nodes. A number of researchers [4, 6] have worked on incentive schemes for P2P systems which try to ensure consistent collaboration between peers while restricting the free rider problem. Yet another problem in P2P networks is collusion among a group of nodes to incorrectly malign or promote another user to exploit the system. Previous research in this area [5, 9] address these problems using reputation and referral based mechanisms that identify and reward nodes contributing significantly in the system. However, most of these mechanisms are not efficient and do not adequately address all the problems introduced by zero cost identity and group collusion in the system. These solutions typically assume that nodes share a common history, which is a major limitation. Therefore, developing techniques that prevent malicious exploitation of nodes in a P2P network remain an open and challenging research problem. Here we consider a P2P network, where, users are located on nodes and each node is provided with a self-interested autonomous agent that determines the resource sharing decision for that node. Henceforth, we use the term node and agent interchangeably.

To address the free rider problem in a P2P setting, we propose that resources should be shared with a requesting node in proportion to the resources shared by that node in the past. Reciprocity mechanisms in multi-agent systems provide a suitable paradigm for implementing such a resource sharing strategy. Reciprocity based techniques have already been applied to evolve cooperation and trust in multi-agent societies [12, 11]. Researchers have shown that the development of mutual cooperative relationships leading

AAMAS'05, July 25-29, 2005, Utrecht, Netherlands.

to exchanges of help can improve both agent and system-level performances. But these research assume that an agent cannot change its identity. If exploitative agents can change their identity without any cost, they can constantly exploit reciprocative agents who help newcomers with the hope of bootstrapping cooperative relationships. In this paper, we propose an expected utility based decision mechanism to determine the sharing decision when another agent requests a resource. The proposed decision mechanism of an agent considers its past interaction history with the asking agent, asking agent's reputation and the interaction possibility with the asking agent in future. When a resource is requested from an agent, it uses this decision mechanism to evaluating its chance of obtaining help from the asking agent in future. The requested resource is provided if there is a net expected benefit for interacting with the requester in the future.

Our mechanism rewards an agent by enhancing its probability to receive resources it requested only when the node itself shares its own resources with other nodes in the system. This ensures that users are motivated to contribute resources and increase their chances of obtaining "rewards" instead of free-riding and switching identities. In our proposed decision mechanism, an agent considers, along with past interaction history and future interaction possibilities, the help giving reputation of the asking agent from the feedback of the other agents about the asking agent. So, it is not necessary for each agent to interact with all of the agents individually which could have been a limitation in a large P2P network. Using an weight update mechanism of the opinions from the other agents, we have made our decision mechanism robust against collusion by a group of selfish, free-riding agents. Experimental results of our mechanism on a simulated P2P network illustrates that the reciprocative technique can mitigate the issues of zero cost identity and collusion among P2P nodes and discourages free-riding in a P2P network.

2. CHALLENGES IN P2P SHARING

A P2P network is set up using the P2P node discovery protocol, and, resources are located by nodes using the P2P resource discovery protocol. In this paper, we assume an existing P2P network where these protocols are already implemented and concentrate on the algorithm for determining the sharing decision made by a node in response to requests for downloading resources it receives from other nodes.

We consider a Gnutella-like pure P2P system that does not contain any central server location that registers information including identity, network usage and amount of contribution or sharing about the participating nodes in the P2P network. Ensuring fairness from every node in sharing resources becomes a challenging problem in such a decentralized and unsupervised environment. The principal problems arising from the open environment provided by pure P2P systems are the following:

- **Zero-cost Identity.** A new node can enter a pure P2P system by sending a node-discovery request to an existing P2P node. This process only involves a nominal cost corresponding to the network usage to send the node discovery request and does not involve any costs for participation in the P2P network. Therefore, it is virtually cost free for a node to enter the network with multiple and continuously changing identities. This problem allows malicious nodes to perform harmful activities in the P2P network, leave the network and reenter with a new identity to continue its malicious activities. Simultaneously, a stringent strategy that deters nodes with new identities from receiving resources from other nodes would dissuade participation in the system. A suitable mechanism for addressing the zero-cost identity problem should moderately provide resources to nodes with new identities to encourage participation, while, rewarding nodes that have a history of active participation in the network over a long time period.

- **Free-Riding.** The free-riding problem in P2P networks in P2P networks involves selfish nodes that obtain resources from other nodes in the network without themselves sharing any resources with other nodes. In the presence of such selfish nodes, not sharing resources becomes a dominant strategy among all nodes in the network and ultimately leads to a passive network without any resource exchange among nodes [1]. Free-riding can be addressed by a mechanism that shares resources a node in proportion to the contribution of the node to the other nodes in the network.

- **Collusion among Nodes.** Because there is no centralized authority in a pure P2P system, it is relatively easy for a group of nodes to collude together to promote one or more nodes in the group, or, malign other "good" nodes in the network. A suitable mechanism for preventing collusion among nodes should consider the contribution of a node to all nodes in the network instead of considering contributions reported by a possibly colluding clique of nodes.

In the following section, we describe our expected utility based reciprocity mechanism for addressing these issues in a P2P network.

3. EXPECTED UTILITY BASED HELPING DECISIONS

We assume a set of $\mathcal{A}$ agents interacting in a P2P environment. The set $\mathcal{A} = \mathcal{A}_r \cup \mathcal{A}_s$. where A_r and and A_s denote the sets of reciprocative and selfish agents respectively. Every agent has expertise in resource type $T \in \Upsilon$ where Υ is the set of all such resource types. Agents request resources of types in which they are not experts from other agents. The probability that an agent has a particular resource of a given type is much higher if an agent is an expert in that resource type than when it is not. In this paper, the corresponding probability values we have used are 1 and 0 respectively. An agent helps another agent if it provides a resource that is requested from it. Reciprocative agents return help, selfish agents do not.

Let $\mathcal{H}$ denote the interaction histories of the agents. $\mathcal{H}$ is an ordered list of tuples where each tuple is of the form $\langle\, i,\, j,\, x,\, t,\, c_i,\, c_j, help\rangle$ where the components are respectively the agent requesting a request, the agent being requested, the type of resource requested, the time of request, the cost of requesting agent to procure the resource by itself, the cost of the helping agent to satisfy the request for help, and whether or not j helped i. Let $\mathcal{H}_{i,j} \subseteq \mathcal{H}$ be the part of the history that contains interactions between agents i and j only. Let H denote the space of all possible histories. Our goal is to derive a decision procedure $\mathcal{F} : \mathcal{A}_r \times \mathcal{A} \times \Upsilon \times H \rightarrow Yes/No$ that maps a request from an agent to another agent to a boolean decision based on the resource type involved and the interaction history of these two agents.

We introduce an expected utility based decision mechanism used by the reciprocative agents to decide whether or not to honor a request for help from another agent. When requested for help, an agent, using this decision mechanism, estimates the utility of agreeing to the request by evaluating its chance of obtaining help from the asking agent in future. An agent, being self-interested, has the objective of earning more savings by receiving help than cost incurred by helping others in the long run. When an agent using this strategy decides whether or not to provide help, it uses a statistical summary of its past interactions with the requesting agent as a metric for evaluating its expected interaction pattern with the latter in future. Using this information, it evaluates the difference between the expected benefit and the expected cost it might incur for that agent by helping it in the future. In the following, we present the expected utility based decision mechanism that agent m uses to evaluate a help request by another agent o for sharing file type τ. The expected utility of agent m for interacting with agent o at time T and future time steps, $E_T(m, o, \tau)$, is defined as:

$$\begin{aligned} E_T(m, o, \tau) = & \\ & \sum_{t=T}^{\infty} \gamma^{t-T} [\sum_{x \in \Upsilon} (D_m^t(x) \mathrm{Pr}_{m,o}^t(x) cost_m(x)) - \\ & \sum_{x \in \Upsilon} (D_{o,m}^t(x) \mathrm{Pr}_{o,m}^t(x) cost_m(x))], \end{aligned} \quad (1)$$

where $cost_i(x)$ is the expected cost that i incurs to procure a resource of type x by itself, γ is the time discount, and Υ is the set of different area of expertise. The evaluation of the expected utility of agent m helping agent o considers all possible interactions in future and for all types of resources. In equation 1, $D_m^t(x)$ is the expected future distribution of resource types that agent m may require at time instance t, and $D_{o,m}^t(x)$ is the expected future distribution of resource types that agent o may ask from m at time instance t. We define $\mathrm{Pr}_{i,j}^t(x)$ as the probability that agent j will share a resource of type x, when requested by agent i at time t.

We observe that $\sum_{t=T}^{\infty} \gamma^{t-T} \sum_{x\in\Upsilon} D_m^t(x) \, \mathrm{Pr}_{m,o}^t(x) cost_m(x)$ is the time discounted (with discount factor γ) expected savings of m by receiving help from o in future. Hence, when an agent m is helped with resource type x, its savings is $cost_m(x)$, the cost it would have incurred to get the file on its own. We use an infinite time horizon and increasingly discount the future estimates by the factor γ^{t-T}, where $0 < \gamma < 1$, and t refers to the time period. The term, $\sum_{t=T}^{\infty} \gamma^{t-T} \sum_{x\in\Upsilon} D_{o,m}^t(x) \, \mathrm{Pr}_{o,m}^t(x) cost_m(x)$ is the net expected cost that can be incurred by m for (a) helping in the current time instance and (b) incurring helping cost for o in the future. Thus, $E_T(m, o, \tau)$ gives the net time-discounted future expected benefit that agent m has for interacting with agent o.

We note that the distribution $D_m^t(x)$, $D_{o,m}^t(x)$ in the future and the help giving probabilities $\mathrm{Pr}_{i,j}^t(x)$ are unknown to an agent. As an approximation, we estimate these values by the corresponding observed values until the current time T. Correspondingly, the time superscripts, t, are replaced with T in Equation 1.

The formulas to calculate different terms in Equation 1 are given below:

$$D_m^T(x) = \frac{\text{no. of times resource of type } x \text{ was required until } T}{\text{total \# of requirements for } m \text{ until } T}.$$

$$D_{o,m}^T(x) = \frac{\text{\# of times } o \text{ asked } m \text{ for file type } x \text{ until } T}{\text{total \# of times } o \text{ asked } m \text{ for help until } T}.$$

$$Pr_{i,j}^T(x) = \frac{\text{\# of times } j \text{ helped } i \text{ for file type } x \text{ until } T}{\text{\# of times } i \text{ asked for file type } x \text{ until } T}.$$

Our proposed decision mechanism $\mathcal{F}$ will evaluate a help request from agent o to agent m for a file of type τ at time T given the history of interactions, $\mathcal{H}_{m,o}^T$ and $\mathcal{H}_{o,m}^T$, between these two agents. The history is used to first calculate $\mathrm{Pr}_{m,o}(x)$ and $\mathrm{Pr}_{o,m}(x)$ values and the requirement distribution $D_m^T(x)$ and $D_{o,m}^T(x)$. It then calculates the expected utility of agent m for interacting with agent o in the current and future time steps, $E_T(m, o, \tau)$. Our prescription is for agent m to help agent o in the current time if this expected utility is positive. But initially as the probability values are all zero and so are these expected utilities, no agent will be inclined to help first. To break this deadlock, we introduce two bootstrapping mechanisms which are discussed in detail in the experimental section accompanied by a discussion of their relative advantages and disadvantages.

3.1 Reputation as a further deterrence to free-riding

The above-mentioned decision mechanism suffers from the problem of inertia. For a sufficiently large agent population, interaction between any two given agent may be infrequent, and it can take a long time to ensure enough interaction among agents to build up informative interaction histories. Consequently, reciprocative agents may require too long to recognize and benefit from other reciprocative agents, which jeopardizes the mutually beneficial relationship between these agents. To alleviate this problem, we propose to use a reputation mechanism to help identify "good guys" without having to have multiple direct interactions with them. We also want to use the same mechanism to identify free riders (those who receive help but do not help back) based on opinions of others who have been exploited by them. In this reputation framework, when an agent m is asked for help by another agent o, m requests other agents, $\mathcal{C}$, who have interacted with o before to share their experiences about o. Upon request, $\mathcal{C}$ agents send report their complete interaction history with o to m. The helping agent, m, then uses this information to compute a more accurate probability of o's help-offering behavior for different resource types by weighing its personal experience with o and the average of the probabilities reported by $\mathcal{C}$ agents. Therefore, the $\mathrm{Pr}_{m,o}^T(x)$ term in Equation 1 is replaced by the reputation of o for providing help for task type x,i.e., $\mathrm{Pr}_o^T(x)$, and which is calculated as

$$\mathrm{Pr}_o^T(x) = (1-\alpha)\mathrm{Pr}_{m,o}^T(x) + \alpha \frac{\sum_{a \in \mathcal{A} - \{m,o\}} \mathrm{Pr}_{a,o}^T(x)}{|\mathcal{A}| - 2}, \quad (2)$$

where $\mathrm{Pr}_{a,o}^T(x)$ is the opinion about o reported by a. These opinions are averaged from all agents except the interacting parties and the weight α on others opinion is an inverse function on the number of times m has asked o for help

$$\alpha = \frac{1}{1 + \text{no. of times } m \text{ asked help from } o}. \quad (3)$$

This strategy helps peer agents share their knowledge about other peers. As different agents start off interacting with different sections of the agent population, by sharing their opinions they can form expectations about a larger section of the population. Such a reputation mechanism, therefore, enables agents to make informed interaction decisions with other agents at a relatively early stage of their lifetime. Over time, however, an agent would have interacted often enough with another agent to be able to rely on its own interaction history. Equation 2 captures the trade off between using local information and others opinion. From Equation 3 we observe that alpha is inversely correlated with the number of time an agent asks help from another agent. This implies that an agent will depend more on its own experience with a longer interaction history.

3.2 The problem of collusion and its deterrence

This reputation mechanism, however, assumes that every agent is truthful in reporting their reputation for other agents. Selfish agents may attempt to disrupt this mechanism by forming colluding groups that falsely report good opinions about other colluders to third parties. If reciprocative agents are not able to discern between such collusion and truthful opinions, colluding selfish agents will be able to exploit others. Sen [12] observes a similar problem with "Believing Reciprocity Agents" and addressed it by having reciprocative agents trust the opinions of those agents with whom they have good balance of help exchange. A shortcoming of that approach is that nothing prevents a reciprocative agent also to lie about other agents with the goal of sharing a larger share of the time and resources available to another agent. We posit that help-giving and reputation-reporting behavior are orthogonal and hence it is necessary to learn them separately for any peer agent.

To address the problem of estimating a peer's reputation-reporting behavior, we propose a Bayesian update scheme to discriminate between truthful and lying agent. In this approach, reciprocative agents compute the probability that the information supplied by another agent is true. Initially, it assumes every one to be truthful and then uses a Bayesian update technique to judge the truthfulness of each agent based on its interaction experience with those about whom reputation was reported. Subsequently, the opinions reported by an agent is weighted by its estimated truthfulness. Equation 2 is then updated to include this estimated truthfulness:

$$\mathrm{Pr}_o^T(x) = (1-\alpha)\mathrm{Pr}_{m,o}^T(x) + \alpha \frac{\sum_{a\in\mathcal{A}-\{m,o\}} \mathrm{Pr}_{a,o}^T(x)\mathrm{Pr}^T(a)}{|\mathcal{A}|-2}, \tag{4}$$

where $\mathrm{Pr}^T(a)$ is the estimated probability, at time T, of peer a being truthful. Let the initial estimates for the truthfulness of all agents be some constant $\mathcal{P}$.

We now illustrate the Bayesian update mechanism. Let $H_{o,m}^{x,C}$ and $\overline{H_{o,m}^{x,C}}$ denote respectively the events of agent o helping and not helping agent m with a resource type x at the current time instant, C. Agent m then recalls all the opinions it had received from other agents about o the last time T when o had asked m for help. As stated above, agent m received opinion $\mathrm{Pr}_{a,o}^T(x)$ from another agent a at time T about the likelihood of agent o helping for resource type x. The new truthfulness estimates for the next time instance, $\mathrm{Pr}^{C+1}(a)$, is then calculated as

$$\mathrm{Pr}^{C+1}(a) = \begin{cases} \mathrm{Pr}^C(a|H_{o,m}^{x,C}) & \text{if } o \text{ helped m with } x \text{ at } C. \\ \mathrm{Pr}^C(a|\overline{H_{o,m}^{x,C}}) & \text{if } o \text{ did not help m with } x \text{ at } C. \end{cases}$$

Now,

$$\mathrm{Pr}^C(a|H_{o,m}^{x,C}) = \frac{\mathrm{Pr}^C(a)\mathrm{Pr}(H_{o,m}^{x,C}|a)}{\mathrm{Pr}(H_{o,m})},$$

where the denominator on the RHS is the true probability of agent o helping m with resource type x if asked. Since this probability is not known, it is approximated by the estimated probability in Equation 4. Also, the second term in the numerator in the RHS is the reputation that a would report about o's help-giving likelihood for resource type x at time C. Since, this information was not solicited by m, and hence not available, the most recent such information, from time T, can be substituted. This results in the updated equation to be

$$\mathrm{Pr}^C(a|H_{o,m}^{x,C}) = \frac{\mathrm{Pr}^C(a)\mathrm{Pr}_{a,o}^T(x)}{\mathrm{Pr}_o^C(x)}.$$

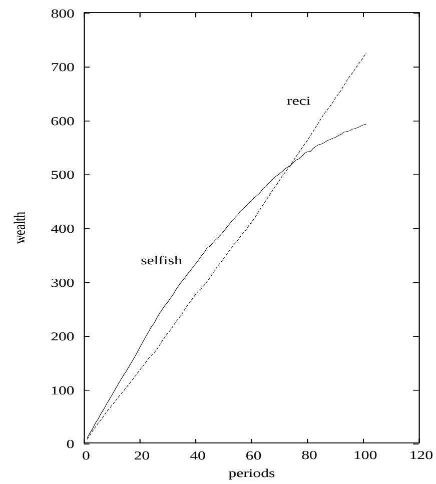

Figure 1: Performance of reciprocative and selfish agents.

Similarly,

$$\mathrm{Pr}^C(a|\overline{H_{o,m}^{x,C}}) = \frac{\mathrm{Pr}^C(a)(1-\mathrm{Pr}_{a,o}^T(x))}{1-\mathrm{Pr}_o^C(x)}.$$

3.3 Addressing the zero-cost identity problem

In P2P systems, it might be possible to regenerate new identity at near-zero cost. This might allow free-riders to bypass the reputation based deterrence discussed above unless corresponding corrective measures are adopted. We propose the following simple scheme: newcomers are not helped until their reputation is above a threshold for at least one resource type. This will require the newcomers to "invest" to enter the market by incurring upfront costs when it helps others without helping back. This is a reasonable demand and is faced by most entrants to new environments. The above measure, however, poses the following caveat: who should the newcomers help? If they help anyone requesting help, free-riders can get their way. We propose that the newcomers use the above-mentioned reputation scheme to decide who to help. Though the reputation mechanism is not as effective for newcomers, as they cannot use their experience to update the truthfulness of the other agents, it will work as long as a majority of the population are not colluding as a group. The latter is a very unlikely scenario in large-scale, open P2P communities.

In the next section we discuss the experimental results we obtained by using these strategies.

4. EXPERIMENTAL SECTION

We organized the sequence of our experiments to incrementally include the strategies used by the selfish agents and the counter strategies taken by the reciprocative agents to prevent exploitation. We start with very basic strategies used by both type of agents and gradually show the effects of using more sophisticated techniques. In all our experiments we compare the balance of different types of agents averaged over all the agents in that type. Note that higher balance implies more help received than given. We ran our system with 50 agents and 10 different area of expertise. Every agent is randomly chosen to be an expert in an area. In each iteration the agents randomly issues a request in an area of which it is not an expert and ask other agents for help. Other agents then decide to help or not depending on their strategies and their expertise. An

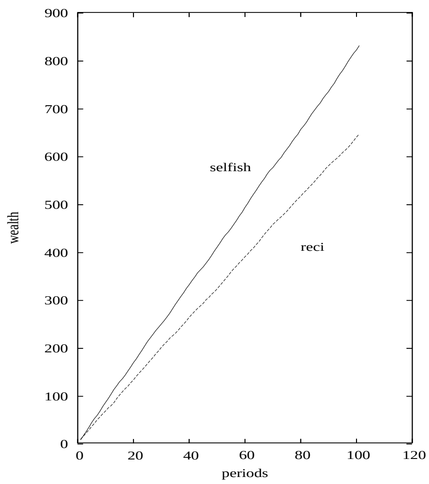

Figure 2: Effect of zero cost identity problem on performance of reciprocative and selfish agents.

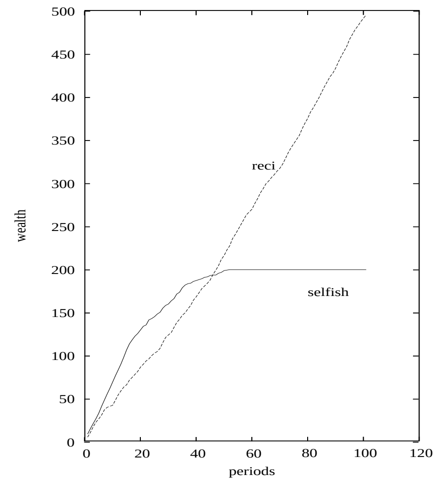

Figure 3: Performance of reciprocative and selfish agents after removing the zero cost identity problem.

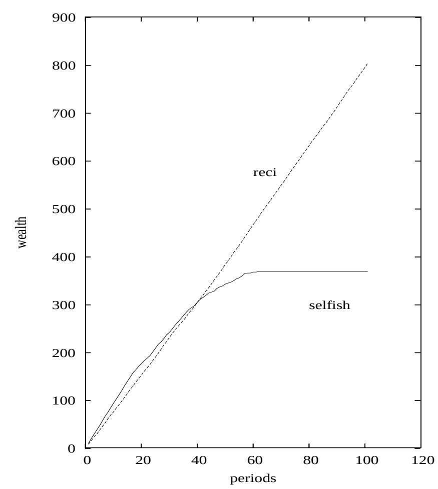

Figure 4: Performance of reciprocative and selfish agents using reputation information from other agents.

agent cannot help in an area in which it is not an expert. We also assume that an expert when helps a non-expert incurs a cost of 10 and the non-expert saves a cost of 1000. Every experiment is run for 100 iterations. The initial proportion of selfish agents in the population is taken as 0.4.

In our first experiment, the reciprocative agents who are experts in the requested area helps an agent if he had received help from him and expects to receive future help. However, if everyone uses this strategy the systems would reach a deadlock state as everyone would expect to receive help from others. So, to bootstrap the help giving behavior, reciprocative agents help any new agent asking for help for the first 5 times without considering its past history or future expectation of getting help in return. After that period it helps only if there is a non-zero probability to receive help in return. Selfish agent never helps. In this framework, we observe from Figure 1 that the reciprocative agents do much better than the selfish agents in spite of being exploited for the first 5 times. We also observe that after some time selfish agents performance cease to increase where as the reciprocative agents perform continually better. This is the point when reciprocative agents stop giving the free help and uses its decision mechanism to identify the selfish agents.

In our second experiment, the selfish agents start changing their identity at regular intervals. The reciprocative agents helps every new agent individually 5 times to initiate interaction. However, as the reciprocative agents are unaware of the change of identity by the selfish agent, they wrongly consider them as new agents. As a result they keep on helping them and were exploited miserably by the selfish agents. The result is shown in Figure 2.

To counter this strategy, reciprocative agents decide to help benevolently only 20 times after it enters the system, irrespective of the agents asked for help. Now as the reciprocative agents do not consider the identity of other agents and uses its decision mechanism after a fixed interval of time, consequently, selfish agents could not exploit them by changing their identity. The result of this experiment in Figure 3 clearly shows that this strategy is able to eradicate the effect of zero cost identity problem and the reciprocative agents outperform the selfish population. However, this strategy is too restrictive and it limits interaction between reciprocative agents also. As a result reciprocative agents treat other reciprocative agents wrongly as selfish agents based on the insufficient information collected during the bootstrapping and stop giving help. Hence, we observe the performance of reciprocative agents deteriorate as well. An easy solution would be to increase the bootstrapping period but note that it would also give the selfish agents more time to exploit.

In our next experiment we address this issue by adding reputation mechanism in our previous system. In this strategy an agent when asked for help beyond the bootstrapping period of 20 helps, it asks other agents about the asking agent to estimate the probability of receiving help in future. It then uses the average of all the reputations and its own experience to decide whether to help. This mechanism allows sharing of the limited interaction history of individuals. This provides an agent more elaborate information about other agents which it cannot perceive from its own limited experience. This helps an agent to discriminate between selfish and reciprocative agents more accurately. The results in Figure 4 shows the balance of reciprocative agents improve by a large margin than

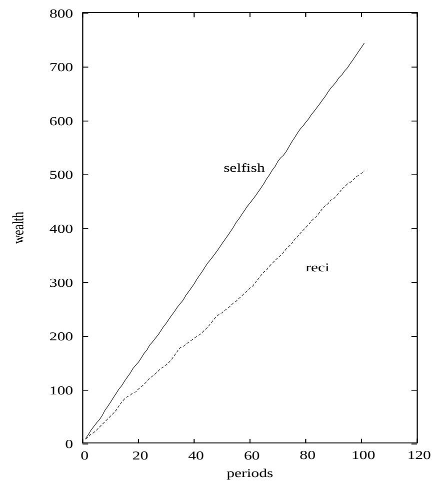

Figure 5: Performance of reciprocative and selfish agents when selfish agents are colluding.

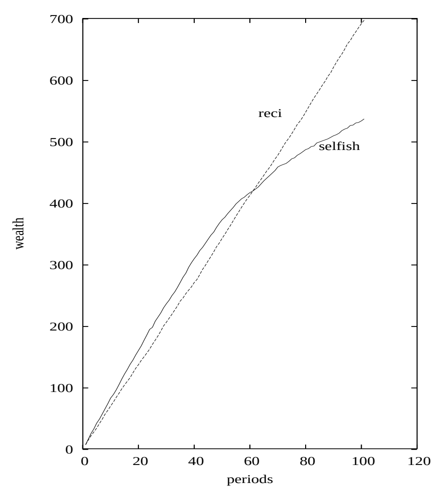

Figure 6: Performance of reciprocative and selfish agents, reciprocative agents using Bayesian update technique.

that of Figure 3 as they can now identify each other correctly. Note that selfish agents perform as poorly as before.

In the next experiment we show the effect of collusion on this reputation framework. The selfish agents tries to exploit the reputation system by colluding among themselves. In this scheme, selfish agents when asked for reputation about other selfish agents gives a very high reputation value to mislead the helping agent. Incapable of distinguishing between truthful and lying agents, reciprocative agents were easily exploited by the selfish agents. Figure 5 shows that selfish agents perform better than the reciprocative ones as they successfully mislead the reciprocative agents using collusion strategy. To alleviate this problem, reciprocative agents computes the probability of the information supplied by another agent being true. Initially It assumes everyone to be truthful and then uses a Bayesian update technique to judge the truthfulness of the agent based on the actual observation it receives from the environment. The information supplied by the agent is then weighted by their truthfulness. We observe from Figure 6 that initially selfish agents do better than the reciprocative agents using collusion. However, reciprocative agents eventually identifies the lying agents and ignore their opinions. In this final experiment we thus produced a robust strategy that maximizes the payoffs for reciprocative agents against the free rider, zero cost identity and collusion problems.

5. RELATED WORK

In this paper we have described an expected utility based reciprocative mechanism for addressing the problems of free-riding, zero-cost identity and collusion among nodes in pure P2P networks. The reciprocity based mechanism described in this paper applies to pure P2P systems that do not have a central server location containing information about the P2P nodes. In contrast, hybrid and centralized P2P systems [7, 13, 14, 10] include one or more servers that contain information about participating nodes, and, fair-sharing can be implemented using audit-based mechanisms in such systems.

Most of the related research for determining the sharing strategy for agents in a P2P network model P2P interactions as a prisoners' dilemma game and suggest mechanisms based on referrals and shared history. Open source systems like Mojo nation [8] use "tokens"(counters) to accrue reputation based on the contributions of a agent in the P2P network. Evolutionary trust based mechanisms for P2P networks have been used in [2, 3, 15] to determine suitable agents to interact with. In [16], P2P systems are modeled as social networks and referrals between agents are used to improve the reputation of a agent in the system. In [9], interactions in a P2P network are modeled as a multiple prisoners' dilemma game and solutions are proposed using reputation based mechanisms. However, the major wrinkle in purely reputation based mechanisms is that they are susceptible to collusion. In [5], P2P interactions are modeled as a prisoners' dilemma game and a maxflow based reputation mechanism is used to solve collusion among agents. Agents maintain both shared and local histories of agent contributions to address the zero cost identity problem. However, the problem with the maxflow based algorithm for collusion prevention is that it considers a subset of agents while considering the reputation of an agent. If all the member of this subset collude to promote or malign the reputation of the agent being referred to, collusion still persists in the system. In contrast, the technique proposed in our reciprocity based mechanism collects the reputation of an agent from all agents in the system and, therefore, is insusceptible to collusion.

6. CONCLUSIONS AND FUTURE WORK

In this paper we describe an expected utility based approach to promote cooperation in P2P networks. Experimental evaluation of the proposed decision mechanism on a simulated P2P environment demonstrate that reciprocal resource sharing is the dominant strategy. This implies a dis-incentive to and eradication of free-riding and corresponding overall improvement of system performance. Our mechanism also successfully handles two other major concerns in P2P systems: the zero-cost-identity problem and collusion between free-riders. In particular, we have introduced and evaluated a novel, probabilistic update-based reputation learning technique that protects against colluding free-riders.

We have restricted our simulations using agents who cannot change their attitude during the course of interactions. We plan to introduce dynamic agent behavior where an helping agent can become selfish after some period of time and vice versa. We also want to evaluate the effectiveness of gathering reputation from a limited number of agents in the environment.

Acknowledgments: This work has been supported in part by an NSF award IIS-0209208.

7. REFERENCES

[1] E. Adar and B. A. Huberman. Free riding on gnutella, September 2000.

[2] A. Birk. Boosting cooperation by evolving trust. In *Applied Artificial Intelligence*, pages Vol 14, 769–784, 2000.

[3] S. Braynov and T. Sandholm. Incentive compatible mechanism for trust revelation. In *AAMAS 2002: First International Joint Conference on Autonomous Agents and Multi Agent Syatems*, pages 310–311, 2002.

[4] C. Buragohain, D. Agrawal, and S. Suri. A game-theoretic framework for incentives in p2p systems. In *3rd IEEE International Conference on Peer-to-Peer Computing*, pages 48–56, Linkoping, Sweden, 2003.

[5] M. Feldman, K. Lai, I. Stoica, and J. Chuang. Robust incentive techniques for p2p networks. In *Proceedings of the ACM Conference on Electronic Commerce*, pages 102–111, 2004.

[6] P. Golle, K. Leyton-Brown, and I. Mironov. Incentives in peer-to-peer file sharing. In *Proceedings of the ACM Conference on Electronic Commerce*, pages 264–267, 2001.

[7] R. Mahajan, M. Castro, and A. Rowstron. Controlling the cost of reliability in peer-to-peer overlays. In *IPTPS'03*, pages 21–32, February 2003.

[8] Mojo nation. URL http://sourceforge.net/projects/mojonation/.

[9] K. Rangathan, M. Ripeanu, A. Sarin, and I. Foster. To share or not to share: An analysis of incentives to contribute in collaborative file sharing environments. In *Proceedings of the Workshop on Economics of Peer to Peer Systems,*, 2003.

[10] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker. A scalable content-addressable network. In *SIGCOMM*, pages 161–172, 2001.

[11] S. Saha, S. Sen, and P. S. Dutta. Helping based on future expectation. In *Proceedings of the Second Intenational Joint Conference on Autonomous Agents and Multiagent Systems*, pages 289–296, 2003.

[12] S. Sen. Believing others: Pros and cons. *Artificial Intelligence*, 142(2):179–203, 2002.

[13] I. Stoica, R. Morris, D. Liben-Nowell, D. Karger, M. Kaashoek, F. Dabek, and H. Balakrishnan. Chord: a scalable peer-to-peer lookup protocol for internet applications. In *EEE/ACM Transactions on Networking, vol. 11, no. 1*, pages 17–32, 2003.

[14] The fasttrack protocol. URL http://cvs.berlios.de.

[15] L. Xiong and L. Liu. A reputation-based trust model for peer-to-peer ecommerce communities. In *ACM Conference on Electronic Commerce, 2003*, pages 228–229, 2003.

[16] B. Yu and M. P. Singh. Searching social networks. In *Proceedings of Second International Joint Conference on Autonomous Agents and Multi-Agent Systems*, pages 65–72, 2003.