Autonomous reputation for equitable peer-to-peer resource sharing among eager consumers

Miranda Mowbray, Nazareno Andrade[1], Francisco Brasileiro[1], Walfredo Cirne[1]
Application Systems Department
HP Laboratories Bristol
HPL-2003-211(R.1)
February 28, 2005*

peer-to-peer, reputation, grid

We consider the problem of encouraging equitable resource sharing over a peer-to-peer network of eager consumers. An eager consumer is one that will at times benefit from having as much resources as possible, and the rest of the time will have no requirements for resources from the network. This behaviour is exhibited by consumers of several types of resources. If eager consumers share their idle resources, we show that it is possible to ensure that resources are shared fairly and also that users are encouraged to donate spare resources, by using a simple autonomous reputation scheme. We use this approach in OurGrid, a system we are currently developing. We show through simulations that our reputation scheme can successfully deal with free-riders who try to fool the system by changing identity. Our scheme is very lightweight, and does not require central coordination or a cryptographic infrastructure.

* Internal Accession Date Only
[1] Universidade Federal de Campina Grande, Brazil
Approved for External Publication

Autonomous Reputation for Equitable Peer-to-Peer Resource Sharing Among Eager Consumers

Nazareno Andrade, Francisco Brasileiro, Walfredo Cirne (UFCG)
Miranda Mowbray (HP Labs Bristol)

Abstract

We consider the problem of encouraging equitable resource sharing over a peer-to-peer network of eager consumers. An eager consumer is one that will at times benefit from having as much resources as possible, and the rest of the time will have no requirements for resources from the network. Our study is motivated by the fact that this dichotomy in behaviour is exhibited by consumers of several types of resources. For this scenario, if the eager consumers share their idle resources, we show that it is possible to ensure that resources are shared fairly and also that users are encouraged to donate spare resources by using a simple autonomous reputation scheme. We use this approach in OurGrid, a system that we are currently developing. In OurGrid, each peer A stores for each peer B a local reputation for B based on the past interactions between A and B. If B donates a resource to A, A's reputation for B increases, and if A then donates a resource to B the reputation decreases. If peer A receives conflicting requests for resources from several peers, it will prioritize requests from peers with high reputation. Using this totally decentralized mechanism, the emergent behaviour of the system is that peers who have contributed more to the network are prioritized. This is a very lightweight mechanism, and does not require central coordination or a cryptographic infrastructure.

1 Introduction

Users of a computer network will typically have some periods during which they have processors sitting idle, but at other times would benefit from using extra computing power. The demand times typically vary among different users, so that when one user has spare CPU cycles another user could benefit from using them. A community of network users could therefore gain access to more processing power at the time they could benefit from it if there is a mechanism by which users in the community can donate their spare computing power to other community members. Similarly, they might also benefit from sharing other computing resources such as storage or the use of particular software packages.

However, it may be possible for users to *free-ride*, consuming resources donated by others but not donating any of their own. Experience with deployed peer-to-peer systems shows that in the absence of incentives for donation, a large proportion of the peers only consume the resources of the system [1, 11, 13]. Free-riding decreases the utility of the resource-sharing system, potentially to the point of system collapse. To avoid this, it may be necessary to introduce incentives to make it in users' own interest to share their spare resources with the community.

Ideally, a peer-to-peer resource-sharing mechanism should be equitable, in the sense that peers donating a larger amount of resources to the community should be able to receive a large number back. In particular, an equitable system would give an incentive not to free-ride.

This paper describes an incentive scheme for the case when resource consumers are *eager*, that is, at any given time a peer either has no use for extra resources, or can benefit from as many resources as possible. This scheme was developed for using in OurGrid, a system we are currently developing [2] where peers share CPU cycles in order to run *bag-of-tasks* applications, i.e. massively parallel applications whose tasks are independent [3]. OurGrid exemplifies how, in eager consumption scenarios, the mechanism by which peers identify collaborators (that is, peers who contribute their spare resources to the community) does not have to rely on common knowledge to be effective. It turns out that it is not necessary to store global reputations: local reputations, based only on interactions directly involving the peer that stores them, can be sufficient for an effective incentive scheme.

In the scheme discussed in this paper, each peer maintains local reputations based on its past interactions with each of the other peers and uses these to prioritize incoming requests. Peer A stores, for each peer B, a number representing the current local reputation of peer B. This number increases when B donates a resource to A and decreases (unless it is already at a minimum value) when B is donated a resource from A. Through the autonomous behaviour of its components, the system prioritizes the peers who have higher reputations, motivating sharing.

The rest of this paper is structured as follows. In Section 2 we describe peer-to-peer reputation schemes. Next, in Section 3 we describe a particular resource-sharing system, OurGrid, which uses an autonomous reputation scheme to identify collaborators. We discuss elements of OurGrid's design that offer some protection against peers who try to cheat the system to gain a higher reputation than they deserve. In Section 4 we use simulations to show that the lightweight, autonomous reputation scheme used by OurGrid is enough to ensure equitable sharing. We extend this to a general analytical result for resource sharing among eager consumers, in Section 5: if there is some mechanism (not necessarily the OurGrid reputation scheme) by which collaborators can be identified accurately enough, and collaborators are given priority over free-riders for access to resources, then free-riding will die out. Finally, in Section 6 we discuss the types of resource-sharing systems for which autonomous reputation schemes can be effective.

2 Related work

A reputation scheme for a peer-to-peer system is a way of recording information about past behaviour of peers, for use as a guide to other peers. The information may be derived from objective facts, or the subjective impressions recorded by other peers, or a combination of these.

In *centralized* reputation schemes, such as the one used by eBay to rate traders [6], the reputation information is stored and updated in a central data base accessible by all peers. Central storage creates a bottleneck that may cause delays in information retrieval and adversely affect the robustness and scalability of the system. As a result, peer-to-peer systems in practice tend to use distributed reputation schemes.

A *distributed* reputation scheme needs no centralized entity, and eliminates the potential performance bottleneck. In this type of scheme, the reputation information is available to all peers, but is distributed through different parts of the system. For example, in P2PRep [5] each peer stores information about their own interactions with other peers, and in EigenRep [9] each peer stores local reputation values and in addition random peers store global values derived from multiple local values. In a distributed reputation scheme, a peer can retrieve all the information from the system concerning a given peer, using a retrieval protocol.

A challenging issue that a retrieval protocol must deal with is guaranteeing that the information gathered about peers is reliable, as malicious peers may tamper with the information they store. To assure the reliability of this information, P2PRep relies on voting for gathering opinions about a peer, heuristics to find clusters of potential malicious voters, and on an underlying cryptographic structure to verify the identities of the peers involved in a transaction. Alternatively, in EigenRep some replicated mother peers compute and store a global reputation value for a peer. The mother peers find the peers they must keep track of, and are found by peers who need information, through a distributed hash table.

In contrast, we argue that for settings where there is eager consumption, it is possible to circumvent the need to provide such guarantees by not aggregating a global reputation value for a peer. Instead, collaboration can be efficiently promoted using an *autonomous* reputation scheme. In such a scheme, peers can only access reputation information involving peer-to-peer interactions in which they themselves have participated. This information is stored locally by the peer, so is quick to retrieve. The reputation of a given peer will in general be different in the eyes of different peers, based on their own past interactions with the peer, and there is no attempt to reconcile, average, or combine these local reputations to create a global assessment. Because the system only uses local reputations, there is no potential scalability issue arising from the retrieval and/or storage of global reputations. Moreover, as there is no need for mechanisms to ensure the integrity of information received from other peers about their interactions with third parties, such as a cryptographic infrastructure or a specialized storage infrastructure, the scheme is also lightweight.

Two other peer-to-pear systems are related to our autonomous reputation proposal: KaZaA [14] and GNUnet [8] networks. In KaZaA, the participation level of a node is stored locally in the node, and is communicated as part of requests. Peers then prioritize requesters with greater participation levels. However, users have discovered how to hack the participation level storage, making it possible to set their own participation level at will [10]. Nevertheless, this approach differs from ours as in the scheme we propose the reputation of a peer is stored not in the peer itself, but in the other peers with which it has interacted. This makes it impossible for peers to alter their own reputations by tampering with the information that they store. GNUnet is an anonymous file-sharing network that uses an autonomous trust-based economy focusing on protecting the network from denial-of-service attacks. In GNUnet, a peer makes a request to its neighbors only, and these forward the request as if it was theirs. Peers therefore gain knowledge of which of their neighbors reply to requests, and gain trust in them. In contrast to the work presented in [8], we don't assume the network communication works as in GNUnet – we use a more general model.

3 OurGrid and the Network of Favors

OurGrid, a system that we are currently developing, is a short term solution to the problem of automatic grid assembling for users of bag-of-tasks applications [2]. Through OurGrid, users get access to the idle processors of the community in a peer-to-peer fashion. The assembled processing power from the community forms the grid of each user.

In OurGrid, idle processors are not explicitly advertised, but requests are propagated through the system to as many peers as possible. Messages typically have several alternative routes to reach peers, so that it is difficult for a malicious peer to block others' requests. Peers with idle processors can allocate the use of these processors to a requesting peer, sending the result of the calculation directly to the requesting peer. The peers in consuming state consume all the processing donated to them.

OurGrid uses an autonomous reputation scheme called the *Network of Favors* to help peers with idle processors determine which requesting peer to donate to. A key motivation for the design of this scheme was to make it particularly lightweight and easy to implement in real systems.

The central idea of the Network of Favors is that the users who are greater net contributors of processing power should get higher priority access to the spare processing power of the community. This principle acts as a guide to the apportioning of the available resources among the users currently requesting them and, thus, as an incentive for donation.

In the Network of Favors, allocating a processor to a peer that requests it is a favor, and the value of that favor is the value of the work done for the requesting peer. Each peer keeps a local record of the total value of favors it has given to and received from each known peer in the past. Every time it does a favor or receives one, it updates the appropriate number. The peer calculates a local reputation for each peer based on these numbers, such that a peer who has given many favors and received few will have a high reputation. The peer then uses the current reputations to decide to whom to offer a favor when it has to arbitrate between more than one requester. Thus, whenever there is resource contention, requesters with higher reputation get priority.

As an illustration, suppose that A, B and C have not interacted before. Write $r_A(B)$ for B's local reputation according to A. We have $r_A(B) = r_A(C) = 0$. If B donates some CPU cycles to A, $r_A(B)$ will increase above zero (whereas $r_A(C)$ will remain zero). If then A has to choose between a request from B and a request from C, A will choose to donate to B. A would make the same decision if C had interacted with A before but $r_A(C)$ was smaller than $r_A(B)$.

Low contribution levels can happen for many reasons, for example there may be failures of services or within the communication network; the peer may have few or no spare resources; or the peer may be a free-rider who has decided not to contribute any resources. If the peer has built up a good reputation in the past but then stops contributing, the peer's reputation - and hence the peer's ability to access the spare resources of the community - should gradually diminish.

However, peers with low reputation (including newcomers) still can benefit from membership of the community, since they can use resources that are not requested by any other peer with higher reputation. Allowing peers with low reputations to do this improves utilization of the community resources without harming peers who have earned high reputations. It also ensures that peers outside the community have a short-term incentive to join the community as a newcomer, in addition to the longer-term reputation-based incentive to donate spare resources once they have joined the community.

Also, since an autonomous reputation scheme uses no information on interactions that did not directly involve the peer assessing the reputation, this reduces the options that malicious peers have to distort the reputations. Malicious strategies based on lying about the behaviour of third parties cannot be applied. One of the remaining possibilities for a malicious peer to attack the reputation system is to change identity. We can make it harder for peers to assume an existing peer's identity by using as identity the address to which donated resources are delivered. However, the potential problem remains of a malicious peer that assumes a previously nonexistent identity. We consider this problem in detail in Subsection 3.1.

Another malicious behaviour relevant to our study but not directly related to cheating the reputation system is sabotage. Sabotaging means faking a resource usage, instead of providing the promised service. In a completely autonomous system, each peer has to detect saboteurs without the help of other peers. This potentially makes the autonomous reputation system take more time to discover the saboteurs, and will have a greater impact if the community has a very dynamic behaviour. We address this in Subsection 3.2.

3.1 The identity changing problem

In peer-to-peer networks, it is usually easy for a peer to change its identity by leaving the community and coming back as a supposed newcomer. By this method a peer with a bad reputation can easily start afresh with a newcomer's reputation. Cryptographic or other guarantees of a peer's identity generally can do little to stop this. One solution to the problem is stringent admission control. For example, a network for school children might require permission from a child's form teacher before the child could join the network, and a network for which growth was not a goal might ban all newcomers after an initial start-up phase. However, we would like our reputation scheme to impose as few barriers to growth of OurGrid as possible, so this solution is not suitable for us.

Our approach is quite simple. We award zero reputation to newcomers, and only allow positive and zero local reputations, thus eliminating the incentive for peers to re-enter as a newcomer.

If we allowed negative reputations, this would increase the system's ability to distinguish newcomers with zero reputation from peers who have behaved badly. However, if it is easy for a peer with negative reputation to "zero" their reputation by changing identity, then awarding negative reputations would just give rise to frequent identity changes, and would reward the more determined malicious peers at the expense of legitimate peers with low reputation (as a result of faults in their local machine, for example).

In Yamagishi and Masuda's experiment [15] with an online auction market, a reputation system with negative reputations was successful in the initial phase but rapidly led to a state in which there was a high number of dishonest dealers who frequently returned to the system as newcomers, whereas using a system with just positive (and zero) reputations led to steady improvements in the honesty of the dealers and the quality of goods sold. Over time, the system approached the quality possible if buyers had full knowledge of the quality of the goods sold. This experiment suggests that using only non-negative reputations may be enough to achieve optimum results, even though allowing negative reputations as well as non-negative ones can give more information.

3.2 Calculating the local reputation for a peer

In the Network of Favors, a peer A calculates $r_A(B)$, the local reputation of peer B, using just two pieces of information: the value of favors A has received from B, and the value of favors B has received from A.

Let $v(A,B)$ be the total value of the processing power donated from peer A to peer B over the past history of the system. We want $r_A(B)$ to be a function of $v(A,B)$ and $v(B,A)$, and we want the value of this function to increase when B does a favor for A, to decrease when A does a favor for B, and to be zero if A has never interacted with B.

The simplest function of $v(A,B)$ and $v(B,A)$ that satisfies these conditions is:

$$r_A(B) = v(B,A) - v(A,B) \tag{1}$$

This is just the balance of favors that A owes to B. In a previous work [2] we have shown that through this very simple autonomous reputation mechanism, the emergent behaviour of the community is that the peers who contribute more than they consume are prioritized, promoting equity. However, in that previous work, we did not consider the problem of malicious identity changing. As discussed in Subsection 3.1, a simple and effective solution to this problem is to require the value of $r_A(B)$ to always be greater than or equal to zero, and zero for newcomers. This gives us the slightly more sophisticated function:

$$r_A(B) = max\{0, v(B,A) - v(A,B)\} \tag{2}$$

Using this reputation function makes it possible to avoid prioritizing malicious ID-changing peers over collaborating peers who have consumed more resources than they have contributed. However, under this new reputation function a collaborator A cannot distinguish between a malicious ID-changing peer who never donates any resources and a collaborating peer B that has donated resources to A in the past but consumed at least the same amount of resources from A as it has donated to A. To distinguish between these types of peers, we introduce another term in the reputation function $r_A(B)$, (we call it a history term), which reflects for peer A the history of its donations from peer B. To avoid creating a difference between the reputations of long-known peers and newcomers that is too high, and therefore too costly for a newcomer to overcome, we use a sublinear function of $v(B,A)$

as the history term in $r_A(B)$: for example

$$r_A(B) = max\{0, v(B,A) - v(A,B) + log(v(B,A))\} \quad (3)$$

or

$$r_A(B) = max\{0, v(B,A) - v(A,B) + \sqrt{v(B,A)}\} \quad (4)$$

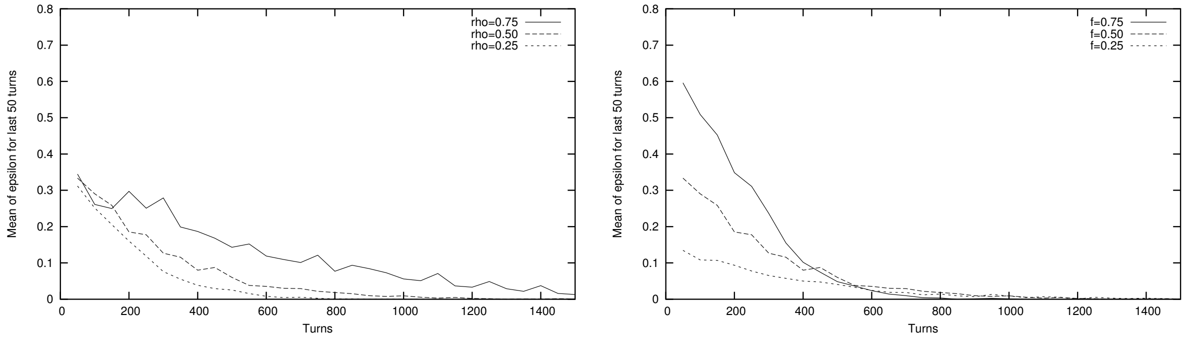
For these functions, there is a relatively large difference in the history term between peers who have not donated to A at all and peers who have donated a little, but not much difference between two peers who both have long histories of reciprocating donations from A. This corresponds to intuition on how the relative values that people attach to favors varies with the amount of past interaction with the person granting these favors. Since the history terms take large positive values for large values of $v(B,A)$, they can make it possible to identify a collaborator even if the collaborator has consumed more resources than it has donated, provided that it has donated enough in the past.

In order to calculate $r_A(B)$, we do assume that A has reliable information about $v(B,A)$ and $v(A,B)$, the value of favors received from and provided to B. Specifically, we assume that A can both (i) measure the value of a favor done by B for A; and (ii) verify that the work done was valid, i.e. that the data returned was not bogus. These assumptions are no stronger than the assumptions made for decentralized reputation schemes. To ensure the integrity of the information, A can use replication to both verify that the work was valid and that the value of the work was as reported by B. A detailed study of this approach applied to voluntary computing called *credibility-based fault tolerance* is presented by Sarmenta [12]. Using this scheme, a peer replicates each task on different service providers until at least a predetermined number of returned results is equal. Also, small fake tasks can be used periodically to verify a resource donator's correctness. By acting correctly, a donator gains credibility in a consumer's view, and the consumer gains confidence about its results. In OurGrid, we intend to implement this credibility-based mechanism both to check for sabotage and to verify other peers' informed accounting. Note that implementing sabotage tolerance for returned results is necessary in any resource sharing system.

4 Simulation Results

This section describes the results of some simulations that show that the autonomous reputation scheme used in OurGrid is effective at distinguishing collaborators from free-riders, and hence promotes equitable resource sharing. We simulated the effects of all four of the reputation functions given in Subsection 3.2. The results for the two reputation functions with history terms were very similar, so we will not report those for the function given in Equation 4.

We start by showing that even the very simplest reputation function – the one given by Equation 1 — makes the amount of resources donated to free-riders when there is resource contention tend to zero. After this, we introduce the case when a free-rider changes identity by leaving the community and returning as a newcomer, in order to get rid of its bad reputation, and show that in this case the non-negative reputation schemes can successfully



(a) $f = 0.5$ and different ρ values

(b) $\rho = 0.5$ and different f values

Figure 1: Measurement of $\epsilon(t)$ for different values of f and ρ, where all 100 peers use $r_A(B) = v(B, A) - v(A, B)$ as reputation function

deal with this problem. Finally, we show that the reputation schemes with history terms have enhanced performance.

Our simulation scenario is a community of 100 peers that, in a time line divided in turns, share their resources. On each turn, each of the peers may be in consumer state with the same probability ρ. Of the hundred peers, $(1 - f).100$ are collaborators and $f.100$ are free-riders. When not in consumer state, each collaborators donates all its resources to one peer chosen among the consumers in the current turn according to their local reputation. The free-riders, on the other hand, never donate. When not consuming, they go idle.

We denote by $\epsilon(t)$ the probability that a peer donating resources at time t will donate them to a free-rider. This can be estimated by measuring the proportion of the available resources that were consumed by the free-riders in the simulation, averaged over the last 50 turns. Figure 1 shows this measurement for the simulation of a 100-peer community where $f = 0.5$ and $\rho = 0.5$. All peers are using the simple balance of the favors exchanged with other peers (as in Equation 1) as their reputation functions. As time advances, the community identifies the free-riders, $\epsilon(t)$ tends to zero, and the free-riders stop getting resources.

Figures 1(a) and 1(b) show the behaviour of the reputation system for varying values of ρ and f. These parameters affect the time the system takes to reach the steady state where the free-riders are all identified and the early values of $\epsilon(t)$ before this state is reached, respectively. The time needed to reach the state where the community has already identified the free-riders well enough for $\epsilon(t)$ to be negligible is proportional to ρ. This happens because the community distinguishes a collaborator when it donates, and high values of ρ indicate that collaborators donate less frequently. In other words, the closer ρ is to 1, the more similar the behaviour of collaborators is to that of free-riders, and the longer it takes for the community to determine that a peer is a collaborator.

On the other hand, for larger values of f, the early values of $\epsilon(t)$ are larger, because if a collaborator donates a resource to a peer with whom the collaborator has not previously had any interactions, the probability that this peer is a free-rider is large for large f. However the

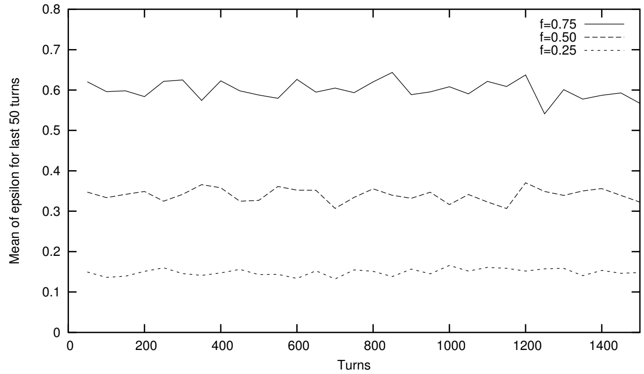


Figure 2: Measurement of $\epsilon(t)$ in a 100-peer community with $\rho = 0.5$ and different f values, but where all free-riders are ID-changers, when all peers use the $r_A(B) = v(B, A) - v(A, B)$ as reputation function

value of f does not appear to significantly affect the time that the system takes to identify the free-riders.

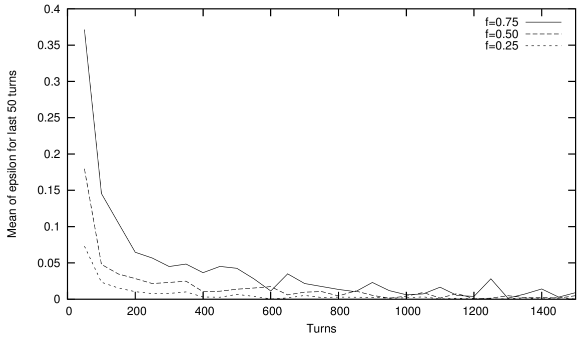
As the second step, we introduce another type of peer in the system, the *ID-changer*. This type of peer is a free-rider that assumes a new identity on every turn, making it impossible for the community to keep track of its consumption. In Figure 2 we show how changing the 50 free-riders with stable ID in the community of Figure 1 ($n = 100$, $f = 0.5$ and $\rho = 0.5$) into ID-changers alters the emergent behaviour of the system.

As can be seen, the capacity to distinguish the free-riders in the community greatly decreases, and $\epsilon(t)$ becomes close to f, which means that the probability that a donating peer selects a free-rider as recipient is close to the probability that a peer selected at random is a free-rider: the reputation information gives no significant help to the donating peer in distinguishing ID-changers from collaborators.

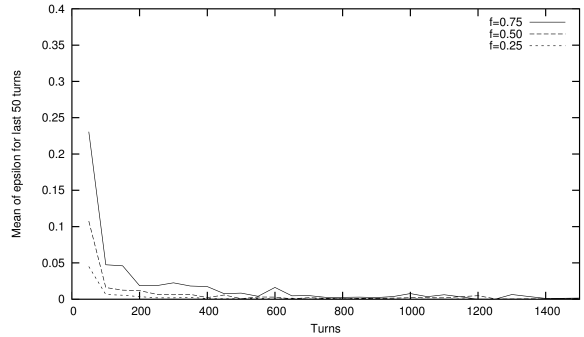
Figure 3 shows the same scenario as Figures 1 and 2 ($n = 100$, $f(t) = 0.5$ and $\rho = 0.5$), but in a community where the collaborators use a non-negative reputation function with and without a history term — to be precise, the reputation functions given by Equations 2 and 3. Figure 3(a) illustrates how the use of a non-negative reputation function improves the robustness of the community to the ID-changing behaviour. Adding the history term $log(v(B, A))$ further improves the ability of collaborators to identify each other, as can be seen in Figure 3(b).

Another interesting effect of using non-negative reputations is that ρ does not significantly affect the behaviour of ϵ in communities that use this kind of reputation. We believe that this happens because, in contrast to systems that use positive and negative reputations, free-riders (and ID-changers) cannot have a reputation that is higher than that of a collaborator, and thus collaborators are more easily differentiated. Moreover, all it takes for a provider to not donate to free-riders in a turn it that one collaborator among the consumers is distinguished. This condition seems to be easily satisfied for any value of ρ.

Finally, we have investigated whether the changes to the original reputation function of OurGrid affected the equity of the Network of Favors, as investigated in [2]. To do this, we



(a) $r_A(B) = max\{0, v(B, A) - v(A, B)\}$



(b) $r_A(B) = max\{0, v(B, A) - v(A, B) + log(v(B, A))\}$

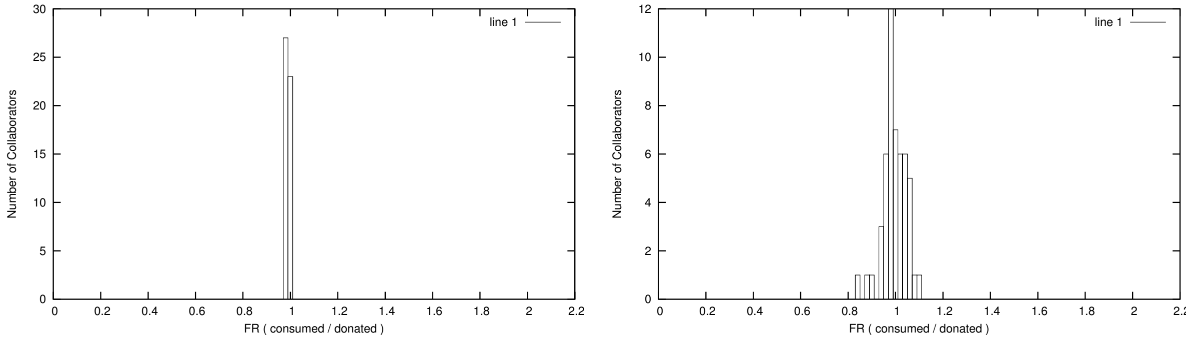
Figure 3: Measurement of $\epsilon(t)$ for two 100-peer communities with $\rho = 0.5$ and different f values. In the first community, peers use a simple non-negative reputation function. In the second they use a non-negative reputation function with history term $log(v(B, A)$, achieving a better performance.

used a metric defined in previous analysis of the system, the *Favor Ratio* (FR). The Favor Ratio for a given peer is defined as the ratio of the total amount of resources it has consumed from the community to the the total amount of resources it has donated to the community. If we have FR=1, there is equity of resource distribution. If the peers use the simple reputation function given by Equation 1, and there is enough contention for resources, then FR= 1 for all peers.

In Figure 4 we show that the Network of Favors still promotes equity when peers use the non-negative and non-negative-with-history reputation functions given by Equations 2 and 3. The histograms show the distribution of FR for all collaborators in the community at the end of 3000 turns of simulation. We observe that FR in both cases converges to one, so equity still holds.

Note that in Figure 4(b), where peers use the reputation function with a history term, FR has a looser convergence than in Figure 4(a), where the community uses the reputation function without a history term. Besides, the histogram of FR does not change significantly for communities with reputations that consider history even if we run the simulation for larger number of turns. This happens because when the history term is introduced, the peers prioritize collaborators with whom they have a long history of interactions over consumers with greater balances with whom they have had less interaction. That is, peers start creating long-term relationships of trust and preference, trading some of the equity of the system for more stable relationships. Nevertheless, in the scenarios investigated, this trade-off has not significantly affected the equity of the network of favors.

In our simulations we have assumed that peers do not change their strategies from collaborator to free-rider, or from free-rider to collaborator. If they did change their strategies, the value of f would not be fixed, but would vary according to the number of free-riders. The



(a) $r_A(B) = max\{0, v(B, A) - v(A, B)\}$

(b) $r_A(B) = max\{0, v(B, A) - v(A, B) + log(v(B, A))\}$

Figure 4: FR measured for all collaborators in two 100-peer communities where $f = 0.5$ and $\rho = 0.5$, after 3000 turns. In (a), peers use a simple non-negative reputation function. In (b), they use a non-negative reputation function with $log(v(B, A)$ as a history term. In both FR converges to 1, denoting equity.

simulations show that for different values of f, the system tends to a state where free-riders do not gain resources, and so are at a disadvantage with respect to collaborators, provided that the cost to collaborators of donating resources is smaller than the benefit they gain from resources donated to them. This suggests that if peers in OurGrid change their strategies according to their own economic interest, free-riding will eventually die out. We will prove an analytical result in Section 5 that supports this hypothesis.

5 Resource sharing among eager consumers

In this section we prove a general analytical result about resource sharing among eager consumers: if the community has some mechanism (not necessarily the mechanism used in OurGrid) by which it can identify collaborators with sufficient accuracy, and known collaborators get priority in access to the resources, then it pays to be a collaborator. As a consequence, if peers change their strategy to or from free-riding if it is in their interest to do so, then the community evolves to a state where there are no free-riders.

Let n be the number of peers in the community (we do not necessarily assume that n is large). Since we allow peers to change their strategy, the number of free-riders will vary over time. Let $f(t).n$ be the number of peers that are free-riders at time t. The other $(1 - f(t)).n$ peers are collaborators at time t, donating all their spare resources to the community.

Since the peers are eager consumers, a peer that is donated a resource always gains some positive utility as a result, no matter how large the quantity of resources it is simultaneously consuming from other sources. It is therefore reasonable to assume that the utility lost by the donor as a result of donating the resource is a fixed multiple v of the utility gained by the recipient, with $0 < v < 1$.

If $f(t) = 0$, then all peers are collaborators. If $f(t) = 1$ the community contains only free-riders, so no resources are donated and there is no incentive for any peer to remain in the community. Suppose now that $0 < f(t) < 1$ at some time t. Let $\epsilon(t)$ be the "error" probability at time t that if a collaborator has a spare resource, it will donate the resource to a free-rider. This may happen either because the collaborator cannot distinguish another collaborator to whom it can donate its resources to or because there are only free-riders requesting resources at time t.

Now if the utility gained by the recipient of a particular resource donation is u, then the total expected utility gain to the set of collaborators as the result of the donation is $(1 - \epsilon(t)).u - v.u$, where t is the time that the donation takes place (the donor must be a collaborator, because free-riders do not donate resources), and the total expected utility gain to the set of free-riders arising from the donation is $\epsilon(t).u$. Since there are $(1 - f(t)).n$ collaborators, the expected utility gain to an average collaborator is $(1 - \epsilon(t) - v).u.(1 - f(t))^{-1}.n^{-1}$. Similarly, since there are $f(t).n$ free-riders, the expected utility gain to an average free-rider is $\epsilon(t).u.(f(t).n)^{-1}$. Therefore, it is better to be a collaborator provided that $(1 - \epsilon(t) - v).u.(1 - f(t))^{-1}.n^{-1} > \epsilon(t).u.(f(t).n)^{-1}$, which happens if and only if $\epsilon(t) < (1 - v).f(t)$. It follows that if $0 < f(t) < 1$ and the error probability $\epsilon(t)$ is less than $(1 - v).f(t)$, then it pays to be a collaborator.

We assume that peers will gradually change their strategies to or from free-riding if it is in their interest to do so — that is, if the expected utility with the new strategy is greater than the expected utility with the old strategy. As free-riders change their strategy to collaboration, $f(t)$ will decrease. Therefore if the value of $\epsilon(t)$ is less than or equal to some value $\epsilon' < (1 - v).f(t)$ for all time after t, then the system should eventually reach a state in which the number of free-riders is at most $\lceil n.\epsilon'/(1 - v)\rceil$. It follows that if the community has a strategy to ensure that the value of $\epsilon(t)$ is bounded above by $1 - v$ and tends to zero as time progresses, then free-riding will eventually die out.

The simulation results for OurGrid in Section 4 with fixed numbers of free-riders showed the measured value of $\epsilon(t)$ tending to zero over time, and bounded below $1/2$ (which should be lower than $1 - v$, since v is close to zero) after a sufficient time interval. This, combined with the analytical result that we have just proved, supports our claim that if peers in OurGrid change their strategies according to their own economic interest, then free-riding will die out.

6 On autonomous reputation

The main example that we have given in this paper for the use of autonomous reputation schemes has been the sharing of processing power for bag-of-tasks applications, in OurGrid. We now discuss whether such reputation schemes could be effective for promoting equity and donation in other resource-sharing networks.

We believe that the key property of our application that allows autonomous reputation schemes to be effective is that there is eager consumption of the resources. An eager consumer tries to consume resources from as many other peers as possible. As each consumer will potentially interact with all other peers in the system, eager consumers can gain first-hand information about the behaviour of other peers more quickly and easily than would be the case for a system where a peer contacts only one service provider at a time, as is the case for some non-eager resource sharing.

For example, it is possible to try to prioritize the peers who contribute more to a file-sharing application by using an autonomous reputation scheme. However, the effectiveness of such a scheme would depend dramatically on the community workload (demand and availability of files) and the capacity of the provider nodes. If peers are not eager consumers, but instead there is a limit on the amount of resources that a consuming peer could profitably use, and it is common for donated resources not to be contested, then peers with low resource requirements might be able to gain all the resources they could use without building up their reputations through donation. In this case, the reputation scheme would not succeed in giving peers with low resource requirements an incentive to donate. Peers with high requirements for extra resources *would* have an incentive to donate – and hence would be a continuing source of donated resources for the free-riding low-requirement peers.

On the other hand, there are other situations resources are shared among eager consumers: a community sharing CPU cycles whose applications are not bag-of-tasks may still have eager consumption if, for example, replication of tasks among free processors can be used to reduce the application's makespan. There are other possible examples in which the resource shared is a higher-level service. For example, there might be eager consumption for the use of a software package or specialized hardware. Even peer-to-peer file-sharing may exhibit eager consumption if peers download strips from a file from several peers and there is a reasonably large set of highly-demanded files present in most of the network providers.

The key difference between eager and non-eager consumers is that for an eager consumer, the utility gained while consuming resources increases with the number of peers willing to donate resources to the consumer – that is, the number of peers for which the consumer has a good reputation. On the other hand, for a non-eager consumer, maximum utility is attained if just a minimum of the providers who have the desired service, with the desired quality of service, are willing to donate resources to the consumer. For an application such as basic file-sharing, this minimum is usually one. An autonomous reputation scheme does not require consensus between providers on a peer's reputation. As a result, an autonomous scheme is likely to be of little use in an a non-eager consumption scenario where the number of providers needed to fulfill a consumer's request is much smaller than the number of servers with the means to do so.

In the light of this discussion, it is interesting to consider some existing systems that use autonomous prioritization to share resources but do not appear to exhibit eager consumption. eMule [7] uses autonomous reputation for simple file-sharing. We suspect that the eMule reputation system may only have limited influence on the community's behaviour. BitTorrent [4] uses an autonomous tit-for-tat mechanism to decide to whom to upload, and at what allocated bandwidth. Peers constantly test others to find better partners, prioritizing those who are currently good collaborators. However, BitTorrent does not use long-term reputation records: it does not need them, because the BitTorrent community is very dynamic, and long relationships between peers are unlikely. As a final example, GNUnet [8] uses autonomous reputation for prioritizing requests, which are forwarded by neighbors of the requester. The quality of service is proportional to how many neighbors forward the message. So in order to gain maximum utility, a requester needs to send the request to as many peers as it can within its credit limit. The situation for GNUnet is therefore similar to that of eager consumption in the sense that a consumer can benefit by interacting with many different peers in order to fulfill a single request.

Based on the above discussion and examples, we argue that using autonomous reputation is a way of achieving efficient results while maintaining simplicity of implementation for several situations. Whenever the system is characterized by eager consumption of shared resources, autonomous reputation mechanisms are a simple and efficient solution for providing incentives for contributions to the network, and their simplicity is an important feature if implementing more sophisticated reputation mechanisms is costly, as is the case in most peer-to-peer networks.

7 Conclusions and Future Work

We have shown that an autonomous reputation scheme can be sufficient to promote equitable sharing of resources in a peer-to-peer community of eager consumers. In particular, it discourages free-riding, and can successfully deal with free-riders who change identity to try to fool the system. Our scheme is very lightweight and does not require centralized storage or an agreed cryptographic infrastructure. The only implementation issue that we have identified as potentially imposing difficulties for autonomous reputation schemes is sabotage tolerance, which is in fact an issue for any resource sharing system.

We have also given suggestions for the functions used to calculate the local reputation in an autonomous reputation scheme. We have shown how using non-negative reputation functions makes the system robust to malicious identity changing, and that adding a sublinear history term can improve further the system's ability to marginalize free-riders. However, we have not looked in detail at how the choice of reputation function affects the system, or at its role in the formation of long-term trust relationships between peers. As future work we intend to investigate these issues further.

References

[1] Adar, E., and Huberman, B. A. Free riding on Gnutella. *First Monday 5*, 10 (2000). http://www.firstmonday.dk/.

[2] Andrade, N., Cirne, W., Brasileiro, F., and Roisenberg, P. OurGrid: An approach to easily assemble grids with equitable resource sharing. In *Proceedings of the 9th Workshop on Job Scheduling Strategies for Parallel Processing* (June 2003).

[3] Cirne, W., Brasileiro, F., Sauvé, J., Andrade, N., Paranhos, D., Santos-Neto, E., Medeiros, R., and Silva, F. Grid computing for Bag-of-Tasks applications. In *Proceedings of the I3E2003* (September 2003).

[4] Cohen, B. Incentives build robustness in BitTorrent. In *Proceedings of the Workshop on Economics of Peer-to-Peer Systems* (June 2003).

[5] DAMIANI, E., DI VIMERCATI, S. D. C., PARABOSCI, S., AND SAMARANTI, P. Managing and sharing servents' reputations in P2P systems. *IEEE Transactions on Data and Knowledge Engineering 15*, 4 (July/August 2003), 840–854.

[6] EBAY, INC. Reputation - eBay feedback: Overview, 1995-2003. http://pages.ebay.com/help/confidence/reputation-ov.html.

[7] EMULE-PROJECT.NET. eMule site. http://www.emule-project.net/.

[8] GROTHOFF, C. An Excess-Based Economic Model for Resource Allocation in Peer-to-Peer Networks. *Wirtschaftsinformatik* (June 2003).

[9] KAMVAR, S. D., SCHLOSSER, M. T., AND GARCIA-MOLINA, H. EigenRep: Reputation management in P2P networks. In *Twelth International World Wide Web Conference* (Budapest, Hungary, May 2003). Preprint at http://dbpubs.stanford.edu:8090/pub/2002-56.

[10] MENNECKE, T. The latest Kazaa Hack - version 2.5. Slyck News, from http://www.slyck.com/news.php?story=217, August 2003. Available on August 12, 2003. Also available at http://lsd.ufcg.edu.br/~nazareno/download/latest-kazaa-hack.ps.

[11] RIPEANU, M., AND FOSTER, I. Mapping the Gnutella network: Macroscopic properties of large-scale peer-to-peer systems. In *First International Workshop on Peer-to-Peer Systems (IPTPS)* (2002).

[12] SARMENTA, L. F. G. Sabotage-tolerance mechanisms for volunteer computing systems. *Future Generation Computer Systems 18*, 4 (2002), 561–572.

[13] SAROIU, S., GUMMADI, P. K., AND GRIBBLE, S. D. A measurement study of peer-to-peer file sharing systems. In *Proceedings of Multimedia Computing and Networking 2002 (MMCN '02)* (San Jose, CA, USA, January 2002). http://citeseer.nj.nec.com/saroiu02measurement.html.

[14] SHARMAN NETWORKS. KaZaA site. http://www.kazaa.com/.

[15] YAMAGISHI, T., AND MATSUDA, M. Improving the lemons market with a reputation system: An experimental study of internet auctioning, May 2002. http://joi.ito.com/archives/papers/Yamaghishi_ASQ1.pdf.