Reputation Lending for Virtual Communities

Anurag Garg Alberto Montresor Roberto Battiti

Dipartimento di Informatica e Telecomunicazioni
Università di Trento
Via Sommarive 14, 38050 Povo (TN), Italy
{garo,montreso,battiti}@dit.unitn.it

Abstract

Reputation management schemes have recently emerged as a mechanism for improving trust and security in peer-to-peer networks. A new entrant in a network with reputation management can either be given the benefit of doubt and be treated like a trusted (or semi-trusted) peer until it misbehaves or it can be assumed to be untrusted and have to earn the trust of others. The former case provides an incentive for misbehaving peers to cast off their old identity and assume a new one regularly. The latter case presents the problem of new peers being unable to bootstrap and to gain a foothold in the network.

In this paper, we present a mechanism in which existing peers can choose to "lend" part of their reputation to new peers that they know in order to give them a start. This mechanism mirrors the real world where a reference or an introduction often gives one a foothold into an otherwise closed group. If the new peer behaves well, the old peer is rewarded for adding value to the group and if the new peer misbehaves the old peer loses the reputation it lent. In this way, existing peers have a stake in introducing new peers into the network and get a return on their investment for introducing honest peers.

1. Introduction

In a cooperative peer-to-peer (p2p) system participating nodes (peers) are expected to share their resources, such as storage, computing power, bandwidth, and content, in exchange for access to resources provided by other nodes. The ultimate goal is to increase the overall utility of the system.

The reality, however, is not so idyllic: the human users behind the nodes may follow personal agendas, and the nodes they control may deviate from the expected behavior. Misbehavior, or rather behavior that does not conform to system goals, can be classified in two main categories [2].

Rational users may be *selfish*, and modify their nodes in order to increase their personal utility. Here, personal utility may be informally defined as the ratio between the remote resources consumed and the local resources shared.

Malicious users, on the other hand, may be guided by a different notion of utility, or behave completely irrationally. For example, record companies that sell music CDs may be interested in disrupting the functioning of a file sharing network, for example using denial of service (DoS) attacks, or by inserting bogus files in the system.

The problems caused by these two types of "disobedient" users are of different scales and levels of severity. While malicious attacks are more dangerous, they require significantly more technical expertise, time and effort from a node. Behaving selfishly is much easier; for example, in Kazaa it is sufficient to set the participation level to *Master* permanently and a hacked version called KazaaLite does precisely this. Therefore, it is reasonable to assume that the number of malicious agents in a p2p system is likely to be far smaller than the number of freeriders.

The problem of implementing distributed systems in the presence of malicious users has been extensively studied in the past [2, 4]; theoretical results exist on the solvability of specific problems, like consensus, in the presence of a (limited) number of malicious nodes. Furthermore, traditional security techniques may be used to deal with specific kind of attacks; for example, cryptographic signatures for ensuring integrity of data while in transit.

More recently, the use of reputation management systems has emerged as a possible mechanism to enforce fair sharing of resources and honesty in p2p systems where a large fraction of nodes may be guided by selfish interest. Different techniques exist, ranging from simple schemes such as tit-for-tat in BitTorrent [5] to complex distributed schemes such as EigenTrust [12] to credit-based schemes such as Scrivener [13]. In all these systems, the underlying principle remains the same: decide if a peer is trustworthy/cooperative based on its past behavior (hon-

esty/willingness to share its own resources) and provide services to the peer based on how trustworthy/cooperative the peer is; therefore creating an incentive for positive behavior.

In this paper, we focus on a specific problem of reputation that has been poorly analyzed before: the treatment given to new peers when they enter the system. It is in the interests of the system to encourage new entrants into the system as it increases the resources available for sharing and increases the usefulness of the system. At the same time, the system must guard against admitting too many selfish or malicious peers as they can drain system resources without contributing anything in turn and endanger system stability.

The treatment of new peers depends on the reputation model of the system to a large extent. In a system like complaints-based trust [1], it is assumed that the vast majority of peers are trustworthy. The system only records negative feedback and a peer lacking feedback, as is the case with a new peer, is assumed to be trustworthy. However, this kind of system is open to exploitation as a node may discard its old identity when it has collected enough negative feedback, assume a new identity and start afresh.

Another option is to use only positive feedback, where a new entrant has the minimum possible reputation. This model makes it difficult to distinguish between a new peer and a dishonest or non-cooperative peer. If existing peers choose to interact only with peers with a minimum level of positive feedback, a new peer may thus find itself frozen out of the group or find itself being mistreated by older trusted peers.

The third option is to count both positive and negative feedback [10, 12]. In this model a new peer enters at the middle of the trust spectrum and is treated at par with a peer who behaves honestly and dishonestly roughly the same proportion of time. However, if we assume that a large majority of peers are trustworthy, a new peer again runs the risk of being marginalized as older peers may not wish to risk being cheated by interacting with a peer whose reputation is significantly lower than the group average.

In this paper we present a novel technique to enable new peers to surmount the problem of "initiation" into the community. We draw inspiration from the real world where one is often initiated into a group by an existing member. For example, an application for graduate school or a faculty position usually requires recommendation letters from the applicant's professors. The reputation of the existing member is reflected on the new member and other members accord the new member some benefit of doubt. In this respect, our system is not very different from systems like BitTorrent [5] and Scrivener [13] that give a small amount of initial credit to each new peer (or reserve a proportion of resources for altruistic purposes) in order to get them started in the system.

The difference is that in our system each new peer does not automatically get an initial credit as in the above systems but is instead accorded credit only if the peer gains a recommender. The initial credit granted is contingent on the reputation of the recommender. The motivation behind this approach is that peers in the system are more likely to introduce cooperative peers. Our system also differs in that it does not maintain a complex system of credits and debits among peers to make resource decisions. Instead we rely on an underlying reputation management system to make these decisions. The only credit offered is to new peers by the recommender for the purposes of getting started. Subsequently, the new node forms its reputation through normal activity in the p2p system.

A potential criticism of our proposed approach is that it is open to attacks where one member of a group of colluding peers enters the system and behaves honestly to accumulate reputation. It then recommends the other malicious peer into the group. Since these new peers start with a high reputation, they are able to interact with all the other peers in the system and cause significant damage. We tackle this problem by obligating the recommender to put some of its own reputation at risk when it recommends a new member. If the recommended member behaves honestly (fairly) the recommender gets the staked reputation back along with some reward for bringing a useful member into the group. If the new peer is malicious (freerider) the recommender loses the portion of its reputation that it risked.

The rest of this paper is organized as follows. In the next section we present the system model and the adversarial model. Section 3 discusses design and implementation issues. We present our experimental results in Section 4. We discuss related work in Section 5 and Section 6 concludes.

2 System Model

The objective of our scheme is to encourage honest peers to join the p2p community as they will increase total group utility while keeping out freeriders and potentially malicious peers. All peers in the network form a reputation on the basis of their behavior in their interactions with other peers in the community. We use the ROCQ [9] reputation management system to compute reputation values for peers. While in principle the ROCQ algorithm can be applied to both unstructured and structured overlay networks, we assume the existence of a structured overlay that uses distributed hash tables for routing and for selecting score managers that keep track of all feedback pertaining to a peer.

Before each transaction, the peer that is about to provide resources asks for the reputation of the requesting peer. If the requesting peer has a high reputation, implying that it is cooperative, the peer provides the resources requested. If

not, it denies the request. At the end of each transaction both partners send feedback to the other partner's score managers who use this information to construct the peers' reputation. If the system is functioning as desired, the reputation value of all cooperative peers should tend to one whereas that of uncooperative peers should tend to zero. The calculation of reputation values by the score managers depends not only on the opinions reported by the peers but also on credibility and quality values. A detailed explanation of the ROCQ algorithm can be found in [8, 9].

Identity. In the interest of associating good reputation to the actual peers that behave well, each node is assigned a virtual identity. We assume that identities cannot be spoofed; i.e., nobody can acquire the identity of another peer with the intent of stealing its reputation. This requirement can be simply satisfied with the use of asymmetric encryption.

On the other hand, reputation systems are affected by another, more subtle problem. Since identities are used also to associate bad behavior to peers, it is possible that nodes assume successive multiple identities to "whitewash" themselves from any bad reputation they may have accumulated (the so-called Sybil attack [7]). To avoid this problem, a possibile solution is to require a strong mechanism for enforcing *unique* identities, such as the use of public key infrastructure like Verisign. Unfortunately, in open, large-scale systems like P2P and file-sharing applications, this is not always an option. The following paragraphs explain how this problem is dealt with in our proposal.

Bootstrap. In our system each *new* entrant is assumed to start with a reputation value of 0 which is equivalent to the new entrant being uncooperative. Since, the reputation value of a peer is used to decide whether another peer will interact with it, a value of zero signifies that no other peer in the system will wish to interact with it. As a result, the new entrant is denied the opportunity to consume any resources in the system. It is important to note that it is possible for a new peer to start gaining reputation by just providing resources, without consuming them. This is clearly a good initial behavior that can be always be accepted in a system that wants to maximize the overall utility.

Figure 1 shows the basic reputation lending architecture. When a new entrant arrives it sends a request for introduction (a) to its potential introducer. The potential introducer then waits for time T_w (b) before deciding whether to accept the request. If the request is accepted, a message is sent to the introducer's score managers (c) instructing them to lend a part of the reputation. These score managers send signed messages (d) with the lent reputation to each of the nodes that will act as the score managers for the new entrant.

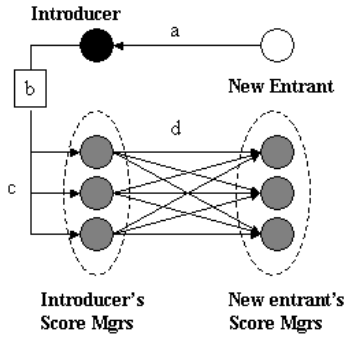

Figure 1. Reputation lending architecture a) introduction request, b) mandatory waiting period T_w, c) introduction message and d) reputation is lent.

On the other hand, a new peer can start in the system with a non-zero reputation by getting an "introduction" from an existing peer. This "introduction" serves as the basis for creating the new entrant's initial reputation and allows it to use the resources provided by the system subject to some constraints. The new entrant's behavior is monitored closely and, if it is uncooperative, its reputation value drops accordingly. If it continues to behave in this manner the initial credit eventually runs out and the peer is gradually excluded from the system. This exclusion is implicit and not explicit as a peer with zero reputation will not find any other peer willing to interact with it.

The overall system is bootstrapped by a small amount of peers that start with maximal reputation. This is reasonable of any community-based system, where the community has to be started by some trusted peers.

Impact on introducer. The peer that introduces a new entrant is termed the *introducer* and is responsible for the consequences of the introduction. If the peer it has introduced becomes a productive member of the community and adds value to the system, the introducer is appropriately rewarded. On the other hand, if the introducer was wrong in its judgment and introduces a misbehaving peer into the system, the introducer is penalized. This penalization reduces the problems related to colluding peers, as the only way to create new reputation is to actually behave well.

Multiple introduction requests. In order to prevent a new peer from bombarding the system with requests for introduction, a waiting period T_w must elapse between the request for introduction and the response, regardless of

whether the introducer decides to introduce the new peer or not. If the introducer decides to introduce the new peer, it waits for T_w time units before contacting its own score manager. It sends a signed message to its score managers telling them to deduct the lent amount R_{lent} from its reputation. The introduction request carries the identity of both the introducer and the new peer to whom this amount is being lent as well as a unique id to prevent duplicate requests. These score managers then send a message to each of the score managers of the new peer telling them to credit the new peer with this amount. Since each score manager of the introducer sends messages to each score manager of the new peer, redundancy is introduced in the system in case a score manager crashes before being able to contact the new peer's score managers.

If the request is denied, the introducer sends a message to the new peer at the end of the wait period T_w informing it. This protocol ensures that the new peer cannot send any more introduction requests before the waiting period is over. The new peer does not know the introducer's decision before the waiting period T_w is over and if it sends an introduction request to a second peer in the system before receiving a response from the first peer, it is possible that both of them may agree to introduce this peer. In this case, the score managers of the new peer would receive two introductions for the same peer. They realize that the new peer is trying to gain unfair advantage and therefore reduce its reputation to zero as a result and may flag it as a malicious peer.

In case the peer can adopt multiple identities, a more sophisticated introduction protocol is required. For example, each request for introduction may require a long computation depending on both the introducer identity and the current identity used by the peer. This would slow down the production of new requests, or require more computational power. Even better, the waiting period between request and response can be used by the introducer to ask for the completion of a task that requires human interaction from the requester. Clearly, any scheme based on prior (personal) knowledge between the requester and the introducer will avoid these kinds of problems; we are investigating the behavior of our protocol in this kind of scenarios.

Attack Model We assume that the attacks that can be launched by a node are limited to 1) behaving uncooperatively (freeriding), and 2) furnishing incorrect or corrupted content. This is in contrast to the more general *byzantine* model that has been proposed in the literature [2, 15] where a malicious peer may attempt to disrupt the system by not forwarding requests, trying to change network topology, launching denial-of-service attacks etc.

It should be pointed out here that the reputation management solution being used, ROCQ, is designed to counter precisely the two types of attacks we mention here [11]. More general malicious attacks cannot be dealt with by reputation management systems in our knowledge and solutions to these kinds of attacks have been proposed by other researchers [3] and can be used in addition to ROCQ.

3 Design and Implementation

We implemented our reputation lending scheme in Java reusing the reputation management code of ROCQ. We implemented a discrete event simulator where exactly one resource transaction is scheduled in each unit of simulation time. We do not model transmission delays or losses and all messages are delivered instantly to the recipient using distributed hash tables. It should be pointed out that the the arrival of new nodes does influence DHT-based routing as the score managers assigned to a peer change over time. However, by using multiple score managers this impact is significantly reduced as was demonstrated in [8].

The requester is chosen at random from the list of peers in the system whereas the respondent is chosen according to the network topology. We model two different topologies: 1) *random* and 2) *scale-free*. In the random topology, all nodes are equally likely to be chosen as the potential respondent. In the scale-free topology, the probability of a node being chosen as the potential respondent is distributed according to a power-law.

A peer decides whether or not to respond to a request from another peer on the basis of the reputation value of the requesting peer. A high reputation value of the requester implies that the system considers the requester to be cooperative. In our system, a correctly functioning peer will respond to a peer requesting the service with a probability that is equal to the requesting peer's reputation R. Therefore, all peers have an incentive to increase their reputation value as that increases their chances of being served. The reputation of a peer, R, corresponds to the proportion of time the peer has offered good service. Hence, a peer is served the same proportion of time that it serves other peers in the network.

After the transaction is completed both parties involved in the transaction report their level of satisfaction to the score managers of its transaction partners. If satisfied, they send a value of 1 to the score manager and if not they send a value of 0. In our model an uncooperative peer would always send a value of 0 for its partners in order to reduce the impact on its own reputation.

New peer arrival. The arrival of new peers is modeled as a Poisson process with the arrival rate equal to λ_a. Of these, cooperative peers arrive at the rate λ_c and uncooperative peers arrive at rate λ_u. Hence the proportion of arriving peers that are uncooperative is $f_u = \frac{\lambda_u}{\lambda_a}$. The arriving peer chooses a potential introducer from the set of peers that are

already in the system. The introducer is also chosen depending on network topology as discussed above.

If the potential introducer decides to introduce the new peer into the network, it lends a portion R_{lent} of its reputation to the new entrant. Introducing a new peer carries a cost as a request from the introducer to another node in the system will now be denied with an additional probability R_{lent}. But the introducer can recoup its reputation in time by behaving cooperatively with other peers. In addition, if the introduced peer is a productive member of the community the introducer is returned this lent reputation and given a small reward.

Types of introducers. We model two kinds of introducers. "Naive" introducers are indiscriminate and will give an introduction to any new entrant that asks for one. "Selective" introducers are more discriminating and only give introductions to peers that they believe will behave in a cooperative fashion. However, the selective introducer also make mistakes in their judgment and introduce a small percentage S_{err} of the dishonest nodes that ask them for an introduction.

Performance audit. The performance of the new entrant is audited after it has been part of the system for some time and has had opportunities to interact with other peers. After the new peer completed *auditTrans* number of transactions its score managers will audit its performance. If the performance is deemed satisfactory based on its reputation value, the introducer is given back the reputation that it had lent R_{lent} along with a small reward R_{rew} for introducing an honest peer to the system. This is communicated to the score managers of the introducing peer who update the reputation value of the introducer subject to the reputation not exceeding 1. If the performance of the new peer is unsatisfactory, the introducer loses the lent reputation and no message to its score managers is sent. The score managers of the new peer also reduce the stored reputation of the new entrant by *introAmt* subject to a minimum of 0.

We do not allow peers whose reputation goes below a certain threshold R_{thresh} to introduce anyone into the system. This prevents peers with low reputations – i.e. uncooperative peers or new peers – from trying to introduce new peers in the network. By keeping R_{thresh} greater than R_{lent} we also prevent peer reputation value from going below zero.

4 Experimental Results

In this section we present simulation results to evaluate the performance of our scheme. Initially, all nodes in the p2p network are assumed to be honest and cooperative. Of these nodes, a fraction f_n are naive introducers and the remainder are selective introducers. We also assume that all new peers that are uncooperative are naive introducers. Among the cooperative new peers, f_n of these are naive introducers and the rest are selective. Each experiment is repeated 10 times and the results shown are the average obtained over the 10 runs.

4.1 Number of Uncooperative Peers Peers vs. Community Growth

In this experiment we compare the number of uncooperative peers in the community as a proportion of the total population. We start with a community of 500 cooperative users. New peers arrive in the system at a rate $\lambda_a = 0.01$ according to a Poisson arrival process. We assume that a new peer will chose a random peer in the system to ask for an introduction subject to the topology. As mentioned above f_n of the original peers in the system are naive and introduce all new peer that make an introduction request into the system.

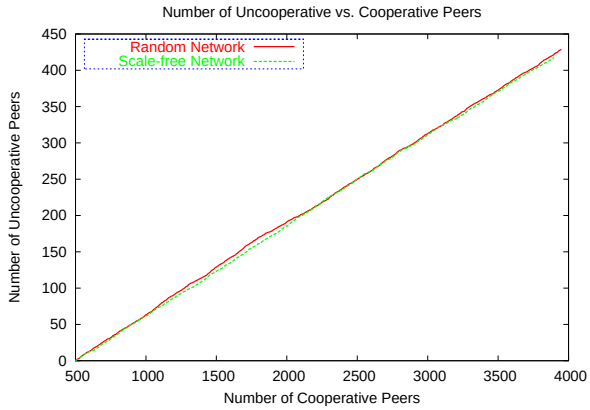

Figure 2. Growth in number of uncooperative vs. cooperative peers

Figure 2 shows the increase in number of uncooperative peers as a function of the number of cooperative peers in the system. We find that the number of uncooperative peers in the system increases linearly with the number of cooperative peers in the system. However, the slope of the increase is significantly less than 0.33 one would expect if all peers were let into the system. Since f_u is 0.25, an uncooperative peer tries to enter the system for every three cooperative peers who try. This is because the selective peers deny introductions to $(100 - S_{err})\%$ of the uncooperative peers that seek introductions from them.

We also find that the rate at which the number of uncooperative peers in the system increases is independent of the

Parameter Name	Description	Default Value
numInit	Initial Number of Peers in the System	500
numTrans	Number of Transactions	500,000
numSM	Number of Score Managers	6
λ_a	Rate of new peer arrival in (simulation time units)	0.01
f_u	Fraction of new entrants who are uncooperative	0.25
f_n	Fraction of cooperative peers who are naive introducers	0.3
S_{err}	Percentage of selective peer introductions that are incorrect	10%
topology	Network topology (Random, Powerlaw)	Powerlaw
T_w	Waiting period for Introductions	1000
auditTrans	Number of transactions after which a new node is audited	20
R_{lent}	Amount of reputation an introducer gives up	0.1
R_{rew}	Reward for introducing a cooperative peer	0.02
R_{thresh}	Minimum Reputation required for introducing a peer	$max(0.5, 2 * R_{lent})$

Table 1. Simulation parameters

network topology. The same amount of uncooperative peers manage to enter the system in a scale-free network as in a random network. We therefore use the scale-free topology for all our remaining experiments.

While the number of uncooperative peers in the system at the end of this experiment is large (about 425) this is much less than the number that would have been allowed in without introductions being required. This experiment ran for 500000 simulation time units. Since $\lambda_a = 0.01$, the total number of nodes trying to enter the system is 5000 of which 1250 are uncooperative. We also see that about 3900 cooperative peers are in the system at the end of the experiment. This is less than the expected 4250 peers (as there were 500 cooperative peers in the system originally). The 350 peers that were turned away include peers who sought an introduction from one of the new entrants who was uncooperative or from one of the naive cooperative peers who lost all of their reputation through inaccurate introductions.

Success Rate. An important measure of our system performance is to look at the proportion of decisions to serve a request or not taken by a cooperative peer that are correct. The ROCQ reputation mechanism is designed to discourage uncooperative behavior as peer with a low reputation are much less likely to be served by another peer in the network. We computed our decision success rate as:

$$Success = \frac{\#Acc_c + \#Rej_u}{Total\#ofRequests}$$

where $\#Acc_c$ is the number requests from cooperative nodes that are accepted and $\#Rej_u$ is the number requests from uncooperative nodes that are requested. We found that when introductions were not required and all nodes were allowed in the system, the success rate was 98.13% whereas when introductions were required the success rate was 98.65%. These success rates were achieved in ROCQ by Garg et al. in [8, 9] as long as cooperative peers are in a majority. Adding the requirement that new entrants be introduced does not change the success rate of ROCQ by a significant amount. We conclude that the introducer requirement is compatible with the ROCQ reputation management scheme.

Peer Reputations. We now look at how peer reputations evolve with time. In Figure 3 we compare the reputations of cooperative peers for different peer arrival rates λ_a over time. As before, the experiment runs for 500000 simulation time units and default values for all other parameters are used (See Table 1). We do not plot the reputation of uncooperative peers as it remains very low (10^{-5} or less) for all arrival rates. We retrieve the reputation values for all cooperative peers every 5000 time units and compute the average. This is then plotted in Figure 3.

We find that the average reputation of cooperative peers (including both peers that were originally part of the system and new entrants) remains more or less constant with respect to time for all values of λ_a. The only exception to this are high arrival rates $\lambda_a \in [0.02, 0.2]$. In these cases the system is overwhelmed by the new entrants. As each new entrant asks for an introduction the reputation of cooperative peers is quickly depleted to a minimum. Thereafter, peer reputations recover as peers interact with each other and cooperative peers give each other positive feedback. This steady state is then maintained for the duration of the experiment.

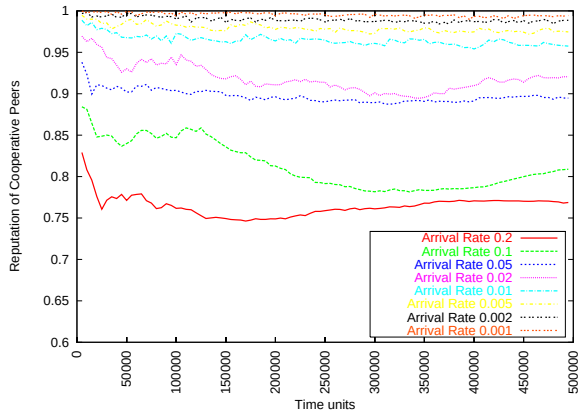

Figure 3. Reputation of cooperative peers with time

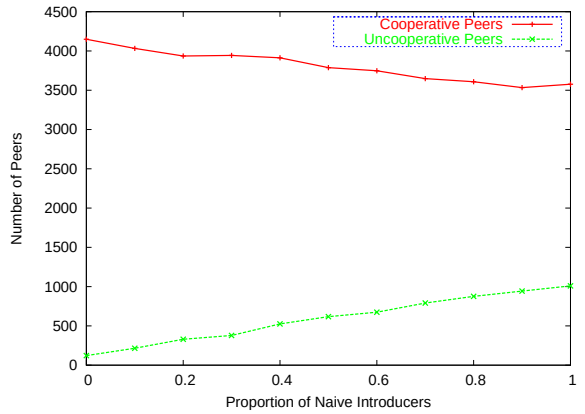

Figure 4. Number of cooperative and uncooperative peers in system with proportion of introducers that are naive

4.2 Proportion of Naive Introducers

The proportion of introducers in the system that are naive has a strong influence on the number of uncooperative nodes that are admitted to the system. In Figure 4 we plot the number of cooperative and uncooperative nodes in the system at the end of 500000 simulation time units. Again, $\lambda_a = 0.01$ for this experiment.

We find that as the proportion of naive introducers increases, the number of cooperative peers in the system decreases from 4200 to 3600 and the number of uncooperative peers increases from 125 to a little over 1000. We can make two significant observations here. Some uncooperative peers enter the system even when all the peers are selective. This is due to the selective peer error rate S_{err} which means that 10% of uncooperative peers that ask for introductions are successful. 125 is indeed 10% of the total number of uncooperative peers that try to enter the system.

We also find that even when all the peers are naive, the number of uncooperative peers admitted to the system is less than 1250. Every time a naive peer introduces an uncooperative new peers, it loses R_{lent} of its reputation. Even though this can be recouped through behaving cooperatively in the network, this is not sufficient. The reputation of some of the naive introducers falls below the threshold required to introduce a new peer in the system and hence new peers are turned away without an introduction. This lack of introductions affects cooperative new entrants equally but that is to be expected as all introducers in the system are naive.

4.3 Amount of Reputation Risked

We now consider the effectiveness our introduction requirement policy as the amount of reputation lent by the introducer changes. Each simulation runs for 500000 time units and we repeat each run 10 times and average the results. The reward for introducing a cooperative member is fixed at 20% of the reputation that is lent ($R_{rew} = 0.2 * R_{lent}$). All other parameters take the default values as noted in Table 1.

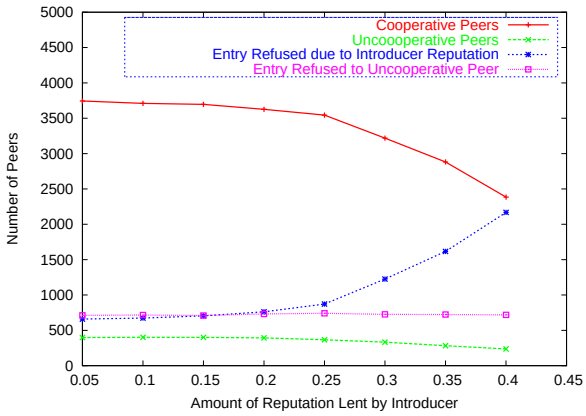

Figure 5. Number of cooperative and uncooperative peers in system with amount of reputation lent by introducer

In Figure 5 we find that as R_{lent} increases, the number of new entrants who are turned away increases. We can conclude this since the number of total peers in the system shows a clear decrease. The number of peers admitted remains more or less the same for $R_{lent} \leq 0.15$ but starts decreasing once R_{lent} becomes larger. Peers can be refused an introduction due to one of two reasons. Either the introducer does not have sufficient reputation to lend to the new

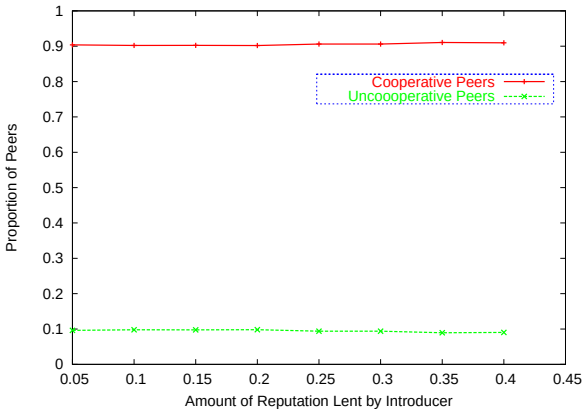

Figure 6. Proportion of cooperative and uncooperative peers in system with amount of reputation lent by introducer

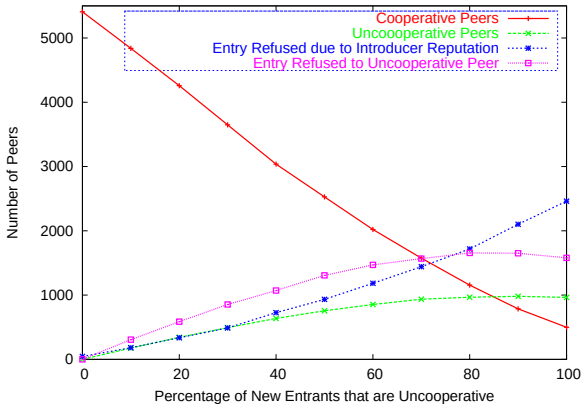

Figure 7. Number of cooperative and uncooperative peers in system with percentage of freeriding new entrants

peer or the new peer is uncooperative and the introducer is selective and decides to withhold the introduction. In Figure 5 we further see that the number of peers being refused entry by selective introducers remains the same. This is to be expected as the proportion of new peers that are uncooperative is not changing. On the other hand, as the amount of reputation being lent upon introduction increases, the number of peers refused entry because their introducer did not have enough reputation increases.

We also see in Figure 6 that the relative proportions cooperative/uncooperative nodes does not change significantly. Therefore, we conclude that increasing R_{lent} beyond 0.15 removes too much reputation from the system. And nodes are prevented from entering the system without distinguishing between cooperative and uncooperative nodes.

4.4 Proportion of Freeriding New Entrants

We now examine how peer introductions are affected by the change in the percentage of new entrants that are uncooperative. Figure 7 shows that as the percentage of uncooperative peers increases among the new entrants, the total number of cooperative peers left in the system at the end of the experiment, decreases. This is to be expected as fewer cooperative peers are trying to enter the system. This curve is almost a straight line with the number of cooperative peers in the system decreasing from 5400 when all new entrants are cooperative to 500 when all new entrants are uncooperative. The former number is 5400 because the remaining 100 peers are still waiting to be admitted into the system when the simulation terminates.

The number of uncooperative peers entering the system does not increase linearly and is bounded at about 1000 uncooperative peers. This illustrates how our reputation lending scheme is successful in preventing the system from being swamped by uncooperative peers. As the graph illustrates, part of this can be attributed to selective peers refusing introductions to uncooperative peers. As the percentage of uncooperative entrants increase, an increasingly important role is also played by the naive and uncooperative peers losing the reputation they lent to other uncooperative peers. Thus, they have no more reputation to lend and cannot introduce any more uncooperative peers.

4.5 Discussion

We have evaluated the performance of our reputation lending scheme over a variety of conditions. We show that our system is successful in reducing the number of uncooperative peers that enter the system without significantly decreasing the number of cooperative peers that enter as long as we use appropriate parameters such as amount of reputation lent to a new entrant and as long as the node arrival rate is not too high. Our simulation also shows that the reputation based decision mechanism is not adversely affected by adding the requirement that all new entrants be introduced. We also find that ROCQ copes with the churn factor due to new peers without any impact on performance.

We find that uncooperative peers are prevented from entering the system by selective introducers who refuse to introduce them. While naive peers do introduce new peers in the system, they gradually lose their reputation as a result of making inaccurate introductions and thus lose their ability to introduce new peers.

We note that as long as there are naive introducers, un-

cooperative peers will not be completely excluded from the system. Moreover, in our model even uncooperative peers are full members of the system and have all rights including introducing new peers. However, uncooperative peers never manage to raise their reputation beyond the threshold required to recommend new peers. On the other hand, even though most interaction requests of new cooperative peers are denied (as are those of new uncooperative peers), they gain positive feedback from the interaction requests that are accepted and gradually gain a high reputation like other cooperative peers.

It is also instructive to note that our experiments assume a random assignment of introducers where new nodes have no control over who introduces them. This is the worst-case scenario as a cooperative node may be routed to an uncooperative node or a node with low reputation for introduction and thus be excluded from the network. In actual applications, it is much more likely that new entrants be recommended by peers that are already known to them.

5 Related Work

There has been much work on reputation management for trust and security and providing incentives for cooperation in p2p systems. Some of these works have addressed the issue of how to deal with new entrants to the P2P system. However, to our knowledge no one has proposed lending reputation to new peers as a means of bootstrapping them.

Systems like BitTorrent [5] and SLIC [16] try to give nodes service levels proportional to their contribution. In BitTorrent, nodes reserve $1/4$ of their bandwidth for serving nodes that may not have uploaded anything to them. This serves to bootstrap the new nodes who then have content which they can share with others.

Samsara [6] ensures fairness by forcing nodes to share as much storage space as they use and challenging nodes periodically to prove that they are actually storing the data they promise to store. Enforcing fairness in storage systems is easier as storage is a relatively stable resource as opposed to bandwidth or computational power and a misbehaving node can be punished by simply deleting its files. KARMA [17] and SeAl [14] track resource usage through distributed auditing mechanisms that keep track of micro-credits.

6 Conclusions

In this paper, we have proposed a new mechanism for bootstrapping new peers that enter a system that uses reputation management for incentivizing cooperation. By restricting admission to peers that are introduced by current members of the system, we show that it is possible to significantly reduce the number of uncooperative peers that enter the system.

Existing network members have dual objectives of increasing their own reputation and maximizing system utility by increasing the number of peers in the system. By introducing new peers, they can fulfill both objectives as bringing a new cooperative peer in the system earns a reward. Conversely, if an uncooperative peer is introduced, the introducer is penalized.

While our current implementation depends on the existence of an underlying reputation mechanism and a DHT-based routing protocol that uses score managers, the basic concept of reputation lending can be extended to other situations as well.

Acknowledgments

This work has been partially funded by the Autonomous Province of Trento through the WILMA project, the European E-NEXT Network of Excellence and the FET unit of the European Commission through projects BIONETS and CASCADAS.

References

[1] K. Aberer and Z. Despotovic. Managing trust in a peer-2-peer information system. In *CIKM*, pages 310–317, 2001.

[2] A. S. Aiyer, L. Alvisi, A. Clement, M. Dahlin, J.-P. Martin, and C. Porth. Bar fault tolerance for cooperative services. In *20th ACM Symposium on Operating Systems Principles*, Oct. 2005.

[3] M. Castro, P. Drushel, A. Ganesh, A. Rowstron, and D. Wallach. Secure routing for structured peer-to-peer overlay networks. In *OSDI '02*, Boston, MA, Dec. 2002.

[4] M. Castro and B. Liskov. Practical byzantine fault tolerance and proactive recovery. *ACM Trans. Comput. Syst.*, 20(4):398–461, 2002.

[5] B. Cohen. Incentives build robustness in BitTorrent. In *1st Workshop on Economics of Peer-to-Peer Systems*, Berkeley, CA, USA, June 5–6 2003.

[6] L. Cox and B. Noble. Samsara: Honor among thieves in peer-to-peer storage. In *Proceedings of the ACM Symposium on Operating Systems Principles*, Oct. 2003.

[7] J. Douceur. The Sybil Attack. In *Proceedings of the 1st International Peer To Peer Systems Workshop (IPTPS 2002)*, Cambridge, Massachusetts, Mar. 2002.

[8] A. Garg and R. Battiti. The reputation, opinion, credibility and quality (ROCQ) scheme. Technical Report DIT-04-104, University of Trento, July 2004.

[9] A. Garg, R. Battiti, and R. Cascella. Reputation management: Experiments on the robustness of ROCQ. In *Proceedings of the 7th International Symposium on Autonomous Decentralized Systems (First International Workshop on Autonomic Communication for Evolvable Next Generation Networks)*, pages 725–730, Chengdu, China, Apr. 2005.

[10] A. Garg, R. Battiti, and G. Costanzi. Dynamic self-management of autonomic systems: The reputation, quality and credibility (RQC) scheme. In *The 1st IFIP TC6*

WG6.6 International Workshop on Autonomic Communication (WAC 2004), Oct. 2004.

[11] A. Garg and R. Cascella. Reputation management for collaborative content distribution. In *Proceedings of the First International IEEE WoWMoM Workshop on Autonomic Communications and Computing, Taormina, Italy*, pages 547–552, June 2005.

[12] S. Kamvar, M. Schlosser, and H. Garcia-Molina. The eigentrust algorithm for reputation management in P2P networks. In *Proceedings of the twelfth international conference on World Wide Web*, pages 640–651. ACM Press, 2003.

[13] A. Nandi, T.-W. Ngan, A. Singh, P. Druschel, and D. Wallach. Scrivener: Providing incentives in cooperative content distribution systems. In *To Appear in ACM/IFIP/USENIX 6th International Middleware Conference (Middleware 2005)*, Grenoble, France, Nov. 2005.

[14] N. Ntarmos and P. Triantafillou. SeAl: Managing accesses and data in peer-to-peer sharing networks. In *Proceedings of the 4th Int'l Conf. on Peer-to-Peer Computing*, pages 116–123, Zurich, Switzerland, 2004.

[15] A. Singh, M. Castro, A. Rowstron, and P. Druschel. Defending against eclipse attacks on overlay networks. In *Proceedings of the 11th ACM SIGOPS European Workshop*, Leuven, Belgium, September 2004.

[16] Q. Sun and H. Garcia-Molina. SLIC: A selfish link-based incentive mechanism for unstructured peer-to-peer networks. In *ICDCS '04: Proceedings of the 24th International Conference on Distributed Computing Systems (ICDCS'04)*, pages 506–515, Washington, DC, USA, 2004. IEEE Computer Society.

[17] V. Vishnumurthy, S. Chandrakumar, and E. Sirer. KARMA: A secure economic framework for p2p resource sharing. In *Proceedings of the Workshop on the Economics of Peer-to-Peer Systems*, Berkeley, California, June 2003.