Cooperation in P2P Systems through Sociological Incentive Patterns

Sebastian Kaune[1], Konstantin Pussep[1], Gareth Tyson[2], Andreas Mauthe[2], and Ralf Steinmetz[1]

[1] Technische Universität Darmstadt, Germany
[2] Lancaster University, United Kingdom

Abstract. While the performance of peer-to-peer (p2p) systems largely depend on the cooperation of the member nodes, there is an inherent conflict between the individuals' self interest and the communal social welfare. In this regard, many interesting parallels between p2p systems and cooperation in human societies can be drawn. On the one hand, human societies are organized around a certain level of altruistic behavior. Whilst, on the other hand, individuals tend to overuse public goods, if they are free to do so. This paper proposes a new incentive scheme that extracts and modifies sociological incentive patterns, based on the Tragedy of Commons analogy, to work efficiently in a p2p environment. It is shown through simulations that this scheme encourages honest peers whilst successfully blocking non-contributors.

1 Introduction

It has long been understood that the performance of peer-to-peer (p2p) systems rely on the cooperation of the member nodes. This realization creates a social dilemma for the users of such systems, as the necessity to altruistically provide resources goes against the selfish desire to limit one's own personal sacrifice. Consequently, users' self interests results in the free-riding problem [1, 2] by trying to exploit others while not contributing themselves. Hence, cooperation amongst peers becomes sparse unless an incentive scheme can encourage participants to contribute their resources.

Considering the tensions between individuality and communal social welfare in human societies, many interesting parallels to p2p systems can be drawn. On the one hand, individuals tend to overuse public goods resulting in the Tragedy of Commons [3]. On the other side, human societies are organized around cooperative interactions and a certain level of altruism. Rich analysis in evolutionary sociology has tried to answer this issue and has largely concluded that indirect reciprocity explains the evolution of cooperation among unrelated individuals [4, 5]. In an extensive study, [6] analyzed how the concept of reputation is used in human societies to encourage cooperation. As an outcome, important incentive patterns were identified that are mandatory for the evolution of cooperation.

Inspired by these findings and the similarities observed between p2p systems and human societies, we propose a new reputation-based incentive scheme that aims to encourage honest users to participate in the system whilst successfully blocking free-riders. Our major contribution can be summarized as follows: we present a new point in the design space of reputation systems by using extremely limited, non-local reputation information, amounting to a single bit per participant. By adopting insights of sociologists, we show that the classification of nodes as either good or bad offers high potential to encourage cooperation while still encoding as much information as necessary to prevent rational/malicious attacks. We further introduce a similarity-based approach to filter out false recommendations submitted by dishonest nodes.

The paper is organized as followed: in Section 2 an overview of related work is provided. Section 3 describes the design of our system, highlighting both the representation of reputations and how they are utilized. Section 4 then outlines a number of practical deployment issues and how we resolve them. Subsequently, Section 4 evaluates, using game theoretic modeling, the effectiveness of our approach whilst, Section 5 concludes the paper, outlining future work in the field.

2 Background

Any participant in a p2p system is both a service provider and a service consumer. A *transaction* is the process in which a provider voluntarily grants a service to a consumer. Accordingly, the consumer benefits from this service whilst the provider pays the cost (e.g. upload bandwidth).

In general, a well-designed incentive scheme has to meet several challenges in order to be robust, notably:

- *Different user types*: Users can be classified into two categories: *obedient* and *dishonest*. The former are consistent with the system specifications and thus contribute to the system whereas the latter try to maximize their benefit at the expense of others.
- *Asymmetry of interests*: For example, peer A is interested in receiving a service from peer B whilst not being able to offer a valuable service in return.
- *Newcomers*: In general, it is impossible to distinguish dishonest nodes from so called legitimate newcomers. Thus, a newcomer policy is mandatory.
- *Untraceable actions*: In decentralized systems, it is impossible to monitor all occurred transactions. Thus, decentralized mechanisms are required to prove that two peers were involved in a distinct transaction.

2.1 Prior Incentive Schemes for Cooperation in P2P

In the area of p2p, various incentive schemes have been proposed to encourage users to contribute their own resources. Some of them are based on *monetary payment schemes* in which peers have to pay for the resources they consume [7–9]. However, many of these algorithms require a centralized infrastructure to enable micro payments and accounting.

An alternative is *reciprocity-based schemes* in which peers use historical information of past behavior of other peers to decide whether they want to share resources or not. These schemes can be further separated into direct and indirect reciprocity. In *direct reciprocity*, user offer resources only to those who have helped them before based on local observations, e.g., BitTorrent [10]. However, it assumes frequent repeated meetings between the same peers which might not be the case in large, diverse p2p environments.

In contrast, *indirect reciprocity* [11, 12] allows peers to claim back their cooperativeness from any peer as each participant is associated with a reputation. Users earn a reputation based on the feedback from others they have interacted with; this, in turn, is used to differentiate between contributors and free-riders. These schemes, accordingly, rely on local observations, and additionally on second-hand information distributed among all nodes in the system [13, 14]. However, this share of own experiences enables malicious nodes to disseminate false information about cooperative participants. To tackle this problem, a sound solution is to determine transitive chains of trust among known and reputable nodes [15, 16]. On the other side, the share of information additionally introduces the collusion problem in which peers artificially increase each other's reputation. A countermeasure against this threat is to apply the computational expensive min-cut max-flow theorem, as proposed by [17, 18].

Different from all studies above, our system design neither relies on transitive trust nor on the often applied min-cut max-flow theorem.

2.2 The Tragedy of Commons Analogy

Many social scientists, as well as psychologists, have tried to explain cooperation in human societies. This problem is also known as the *Tragedy of Commons*. From the societies' point of view, the community does best if all individuals mutually cooperate. However, it can be observed that individuals or groups will exploit the generosity of others, if they are free to do so.

By means of game theory, sociologists try to find evolutionary stable behavioral strategies (ESS) to explain the question of cooperation. In particular, [19] found indirect reciprocity to enable cooperative ESSs in human societies. Further, [6] identified certain key properties of successful reputation schemes to encourage cooperation among unrelated individuals. In particular, these incentive patterns have been proven to be highly robust and stable against different patterns of defection, even in the presence of observational errors concerning individuals' reputation.

In spite of the fact that our work is inspired by these observations, we are aware that the aforementioned studies are carried out in environments that differ from p2p systems in the following points: (*i*) permanent identities (players do not leave the system), and (*ii*) traceable actions (both defection and cooperation). However, both conditions are challenged in p2p systems, and the transfer of these insights to p2p systems must be deliberate.

3 Reputation-based Incentive Scheme

The fundamental aspects of reputation-based p2p systems can mainly be divided into: (*i*) a reputation-based incentive scheme and (*ii*) a distributed reputation infrastructure. The former is of major importance as it describes how reputation is computed within the system. The underlying mechanisms are therefore crucial for the scheme's overall performance and must be carefully designed. The latter,

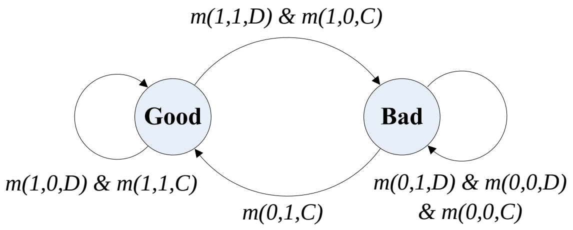


Fig. 1. The 8 reputation transitions

on the other hand, is responsible for implementing (i) in a fully distributed manner; it maintains and stores reputation values, and allows peers to access the reputation of the others.

3.1 Representation of Reputation

In our work, reputation values are represented by a globally binary digit that can be either 0 or 1, indicating a good (G) and bad standing (B) respectively. Let N be the population of peers in our system. Then, the global reputation score r of an individual is given by $r : n \to \{0, 1\},\ n \in N$.

3.2 Assignment of Values

Reputation values are dynamically assigned to peers based on their *last action* when performing the role of service provider. In more detail, if a node takes an action A, either to cooperate (C) or to deny cooperation (D), when there is the option of providing a service, our system assesses the goodness of this action by using *reputation transitions.* In general, each reputation transition m depends on three factors:

- the current reputation value of the service provider r_p,
- the reputation score of the service consumer r_c,
- the taken action A (either C or D) by the service provider.

Thus, each transition is well-defined by a triple m which is mapped to either 0 or 1 as defined in the following:

$$m(r_p, r_c, A) \to \{0, 1\} \tag{1}$$

Fig. 1 shows a state diagram of this transition process, highlighting the 8 possible steps between states leading a service provider to either a good or a bad standing[3]. The design of these transitions is inspired by the observations made in [6]. As stated before, this extensive study identified important incentive patterns that are mandatory to encourage cooperation in human societies. The so called "keys to success" have been defined in the following properties: *being nice* (maintenance of cooperation among contributors), *retaliatory* (identification of dishonesty, punishment and justification of punishment), *forgiving*, and *apologetic.* All of them are incorporated in the depicted transitions:

[3] Depending on the application, a good standing is bounded on predefined time interval, in order to encourage nodes to continuously take the role of a provider. However, we will not pursue this issue any further in this paper.

(1) *Maintenance of cooperation: m(1,1,C)=Good.* If two nodes in a good standing cooperate, the donor should maintain its good standing.

(2) *Identification of Dishonesty: m(0,1,D)=Bad, (1,1,D)=Bad.* Nodes not providing services have to fall into bad standing, irrespective of their reputation.

(3) *Apology and Forgiveness: m(0,1,C)=Good.* Once (mistakenly) fallen into bad standing, there should be an opportunity to allow immediate forgiveness to regain a good standing again.

(4) *Punishment and Justification of Punishment: m(1,0,D)=Good.* When a dishonest node is detected and identified, other nodes contributing to the system should refuse to provide services to it, and should not be punished for this.

The remaining three transitions are degrees of freedom, which we fixed running several experiments measuring the impact of each combination.

3.3 Behavior of Nodes

We define the way a peer uses the reputation scores as its *behavioral strategy* denoted by $\vec{s}$. In more detail, each peer uses a decision function f to decide how to behave towards requesting service consumers. f takes as input parameters the reputation score of both itself and the consumer. There are four possible situations in which a peer i would want to assess another peer j with respect to the reputation scores ($f_{ij} := f(i, j)$):

- f_{00}: both peers are in bad standing
- f_{01}: peer i is in bad standing whereas peer j is in good standing
- f_{10}: peer i is in good standing whereas peer j is in bad standing
- f_{11}: both peers are in good standing.

Thus, $\vec{s}$ consist of four components ($=(f_{00}, f_{01}, f_{10}, f_{11})$) whilst each of them describes whether to cooperate (C) or to deny cooperation (D).

$$\vec{f} : \{0, 1\}^2 \rightarrow \{C, D\} \tag{2}$$

For example, altruistic peers would follow behavioral strategy $\vec{s}_{alt}$ = (C, C, C, C) whereas free-riders are described by $\vec{s}_{free}$ = (D, D, D, D). The built-in incentive in our scheme is based on the assumption that cooperative peers will favor each other. Thus, peers are encouraged to take the role of a service provider in order to gain a good standing. In turn, this greatly enhances the probability of obtaining services provided by others. As shown later on, peers using the *discriminator* strategy $\vec{s}_{disc}$ = (D, C, D, C) can successfully block non-contributors.

3.4 Newcomers

Up to now, we have assumed that nodes already have a standing within the system. Newcomers, however, do not have a transaction history, and are therefore marked as *strangers*. In order generate a good standing, they have initially to cooperate with another stranger or a peer already enjoying a good standing. When requesting services, discriminators will deny to provide services. Accordingly, our system assigns no profit to newcomers.

4 Reputation Infrastructure

Here, we address the practical issues of our approach. In particular, it is specified which nodes are authorized to update reputation values, how reputation values can be globally accessed, and how peers are able to protect themselves against false reports. We assume that users participating in the system are characterized by anonymous identities. Each node owns a public/private key pair suitable for establishing signed messages between nodes. In addition, each participant in the system is identified by a random unique overlay identifier (OId). To ensure that node Ids are chosen randomly from the identifier space, we use trusted certification authorities (CA). These CA's bind a random node id to the node's public key, a process conventionally done offline.

4.1 Replica Set

Due to the lack of a centralized authority, the task to reliably store and update global reputation values is none-trivial and challenging. The peer's reputation must not be stored locally, where it can become subject to manipulation. Storage on a randomly chosen peer similarly does not guarantee that this one is honest. Thus, we assign this task to multiple nodes in the system.

Each peer i is assigned a *replica set* R_i, consisting of a small number of k random peers. To this end, we interconnect all participants in the system using a distributed hash table, e.g. Chord [20]. The members of R_i are then determined by applying a set of k one-way secure hash-functions $h_0(i), h_1(i), ..., h_{k-1}(i)$ to i's overlay id. The hashes derived from these functions constitute the overlay identifiers of the replica set nodes. This ensures that peer i cannot select the members of its own replica set R_i.

If a peer wants to request the reputation of another one, it individually contacts the responsible replica set members. The provided information is *legitimate* if, and only if, more than half of the reports are identical. This implies that the majority of the replica set members must be obedient.

To quantify this, we define a replica set as *reliable* if more than half of the nodes are obedient. Let o and m be the amount of obedient and malicious peers in the system, respectively. The probability to chose an obedient peer for a replica set is $\frac{o}{o+m}$. From this, the probability of obtaining at least $\lceil \frac{k}{2} \rceil$ obedient peers in a replica set is given by $\sum_{n=\lceil \frac{k}{2} \rceil}^{k} \binom{k}{n} \left(\frac{o}{o+m}\right)^n \left(1 - \frac{o}{o+m}\right)^{k-n}$.

For example, the probability of obtaining a reliable replica set in a population consisting of 100.000 nodes, of which $m = 5.000$ nodes are malicious, is 99,88% for $k = 5$. It can easily be verified that k must only be slightly adapted with continuing increase of m.

4.2 Transaction Process

Consumers must submit experiences about the outcomes of transactions (whether a distinct provider p has delivered a service or not) to the provider's replica set R_p. As stated above, this replica set is then authorized to update the reputation value of the provider based on its decision (cf. Sect. 3.2). Since R_p constitutes a third party not directly involved in the transaction process, a mechanism is needed that proves that two distinct peers have interacted with each other. Each transaction therefore consists of five sequential steps:

Step 1. The consumer c creates a service request message containing the following fields $< r_c, pKey_c, OId_c, OId_{lv} >$, where r_c is the consumers current reputation value; $pKey_c$ is its public key; OId_c is its overlay id; and OId_{lv} is the overlay id of the peer that has lastly rated c in the role of provider. Afterwards, c signs the request with its private key and sends it to the chosen provider.

Step 2. Upon receipt, the provider contacts the consumer's replica set R_c to verify the correctness of the information contained in the message.

Step 3. If correct, the provider signs the message and sends it back to the consumer. Thereafter, the service delivery takes place.

Step 4. After the transaction phase is completed, the consumer rates the cooperativeness of the provider (*1*= service received or *0*=service not received) and submits its decision to the provider's replica set R_p.

Step 5a. Each replica set member $R_p(x)$, $\forall x \in [h_0..h_{k-1}]$, first checks whether the service request has been actually signed by provider p. Afterwards, it stores the OId of c as the one of the provider p's last voter, and updates p's reputation value by applying the appropriate reputation transitions.

Step 5b. Additionally, each member of R_p is mapped in the x-th position to its respective counterpart in R_c forming k-pairs. For each pair, the provider's replica set member contacts its counterpart to inform it about the rating c has given on p; this is matter of consequence, as explained in the following.

4.3 Similarity-based Trustworthiness

To reflect the personal experience a consumer has had with distinct providers, each peer i in the system owns a global vector $\vec{t_i}$, where $\vec{t_i} = (t_{1i}, ..., t_{|N|i})$ for all $n \in N$. Each component of $\vec{t_i}$ contains in the n-th position the arithmetic mean of all ratings peer i has submitted on a distinct peer n. Hence, this value describes the *subjective trust* peer i places in peer n. Since each rating can either be 0 or 1, the component values will also be between 0 and 1. According to Step 5b, these trust vectors are stored and maintained by the peer's replica set and are publicly available.

The purpose of these vectors is to define a notion of trust Peer A places in the recommendations of Peer B. To this end, we introduce a similarity function

$$sim(\vec{t}_A, \vec{t}_B) = \frac{1}{|N|} \sum_{i=1}^{|N|} 1 - |t_A(x_i) - t_B(x_i)| \in [0, 1] \tag{3}$$

which in its basic functionality component-wise compares whether both peers have rated the same provider. If so, the deviation between both ratings is calculated and summed up to an overlay similarity S. We define the recommendations of Peer B as *trustworthy* for Peer A, if S exceeds a certain similarity threshold t. In our system, providers apply this function on trust vectors of requesting consumers, in order to determine whether they have maliciously rated obedient providers as bad. Also, it is applied on the last voter of a distinct peer to determine the trustworthiness of his recommendations.

5 Evaluation

In the following, the performance of our scheme is examined against threats of selfish users. Our main goal is to explore which behavioral strategy is the dominant one among a set of chosen strategies. Further, we will examine the effectiveness of our reputation infrastructure against malicious attacks. For that reason, we adopt a game theoretical approach as explained in the following.

5.1 Generalized Prisoner's Dilemma

To model a p2p system by means of game theory, we use the Generalized Prisoner's Dilemma (GPD) that includes two players who interact once in a *one-shot game*, as described in [17]. Unlike the original Prisoner's Dilemma GPD includes the social dilemma and the asymmetry of interests. In particular, each player i follows a *behavioral strategy* by having the choice to cooperate(C_i) or defect(D_i) its opponent. Depending on their actions, each payer receives one of the following *payoffs*: R_i (the reward for mutual cooperation), S_i (the sucker's payoff), T_i (the temptation to defect), and P_i (the punishment for mutual defection). In our context one of the peers acts as provider (P) and the other as consumer (C). The payoff matrix for both consumer and provider is shown in Figure 2(a). To create a social dilemma, the payoffs must fulfill the following criteria:

- Mutual cooperation among peers yields a higher payoff than mutual defection: $R_C + R_P > P_C + P_P$
- Mutual cooperation yields a higher payoff than alternating cooperation-denial cycles: $R_C + R_P > S_C + T_P$ and $R_C + R_P > S_P + T_C$
- In a one shot interaction, defection dominates cooperation as the costs for the service provisioning can be saved: $T_P > R_P$ and $P_P > S_P$

Let $u_{A|B}$ denote the achieved payoff of a behavioral strategy A when interacting with behavioral strategy B.

Definition 1. *Strategy A is said to be dominant if for all B holds $u_{A|A} \geq u_{B|A}$ and $u_{A|B} \geq u_{B|B}$.*

Under this definition, defection would be the dominant strategy for the provider in the one-shot GPD game. Hence, cooperation will never take place and the consumer will only have the choice between the payoffs S_C and P_C.

5.2 Simulations

To assess the performance of our scheme, we have implemented a simulator that corresponds to the above stated game theoretical model. We assume time to be divided into slots, and each slot lasts long enough to allow each peer to provide exactly one service to a requesting consumer. The evaluative scenario we utilize is a file-share application. The assignment of files and queries to peers follows a Zipf distribution ($\alpha = 0.9$). Each file is subdivided into equally sized file segments (chunks) constituting services peers are sharing in the network. Participants fall into two categories: *obedient* and *dishonest*. Obedient nodes

Payoff-Table *General Form*		**Service Provider**	
		Cooperate	Defect
Service Consumer	Cooperate	R_C / R_P	S_C / T_P
	Defect	T_C / S_P	P_C / P_P

(a) Asymmetric payoff matrix

Payoff-Table *Simulations*		**Service Provider**	
		Provide	Deny
Service Consumer	Request	*2 / -1*	*0 / 0*
	Don't Request	*0 / 0*	*0 / 0*

(b) Payoff matrix used in simulations

Fig. 2.

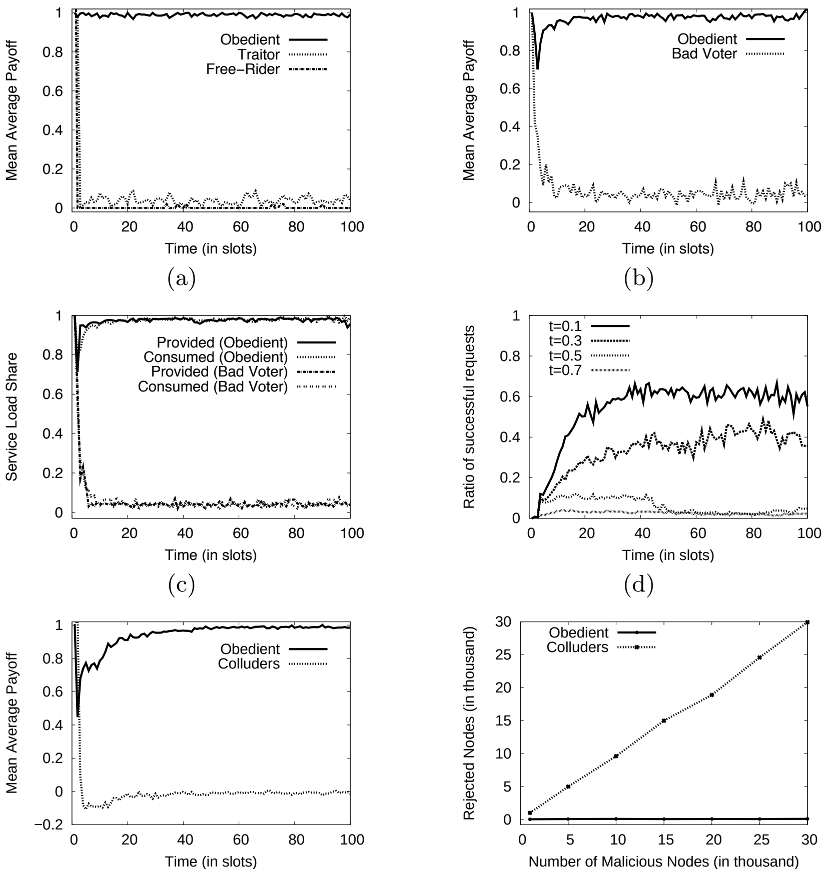
follow the discriminator strategy $\vec{s}_{disc}$, and their similarity threshold is set to $t = 0.7$. The strategy of dishonest nodes will be varied as described later on.

In each slot, each peer has the opportunity to simultaneously act as service provider and consumer. Based on the service a consumer is interested in, it selects the most desired provider and sends a request. Each provider, on the other hand, favours the most appropriate consumer from its upload queue that enjoys a good standing and passes the similarity checks mentioned in Section 4.3. The behavioral strategy of a provider then defines the action (either C or D) she will take by considering the reputation of both herself and the consumer. Depending on how the provider acts, both the consumer and the provider will receive a payoff from the matrix depicted in Fig. 2(b). This matrix satisfies the inequalities stated in the previous section. It is assumed that providing a service incurs the same costs c (-1) to all providers, and consumers receive the same benefit b $(+2)$, respectively. Finally, the reputation and trust vectors are updated after each time slot, and it is assumed that peers can leave or join the system with a probability of 5%.

5.3 Performance under Rational Attacks

In our first experiments, we assume that users do not break down the system specifications (e.g. submit false reports), but try to exploit the generosity of obedient nodes by means of two types of selfish attacks. We consider the first type as *traitors* since these nodes acquire a good standing before turning into defectors. The second type is represented by free-riders who never contribute themselves. Accordingly, we equally divided the population in three groups: obedient peers, free-riders $\vec{s}_{free}$ = (D, D, D, D), and traitors $\vec{s}_{trait}$ = (D, C, D, D).

Fig. 3(a) shows the achieved mean payoff of each strategy per time slot. The highest level of cooperation would be 1 indicating that all peers following the respective strategy are contributing to the system and everyone is able to receive a service. It can be seen that the discriminator strategy applied by obedient nodes achieves the highest payoff over time. More precisely, our simulations revealed that this strategy obtains a mean average payoff of 0.98, indicating that nearly all obedient nodes continuously obtain services. In contrast, free-riders are successfully blocked never receiving a service after the first time slot. Traitors acquire a mean average payoff of 0.05. In particular, only 3%-6% of these nodes are able to receive a service. This stems from the fact that traitors deny to provide services after they have generated a positive standing. Accordingly, they will be subsequently ignored by obedient peers when acting in the role of consumer in the next slot.



(a) (b) (c) (d) (e) (f)

Fig. 3. Simulation results for (a) rational attacks, (b-d) bad voters, and (e-f) colluders.

In conclusion, both types of attackers cannot gain ground in the system as they achieve payoffs close to zero. Obedient nodes, applying the dominant strategy $\vec{s}_{disc}$, self-organize themselves into a robust and cooperative group in which non-contributors are efficiently detected and excluded.

5.4 Effectiveness of Reputation Infrastructure

In the second set of simulations, we study the effectiveness of our reputation infrastructure against malicious nodes falling into two categories: (i) bad voters and (ii) colluders. *Bad voters* follow the discriminator strategy $\vec{s}_{disc}$, but always rate cooperative providers as bad. Accordingly, they are mainly interested in lowering the providers' reputation to encourage other participants to exclusively use their own services. *Colluders*, instead, form a malicious collective and provide services to obedient nodes only with a probability of 20%. Moreover, they boost the reputation values of all peers in the collective by submitting fake transactions. To study these attacks, we assume that 30% of the population consist of malicious nodes from either of both presented categories.

The results of the *bad voter* experiments are as follows. Fig. 3(b) depicts the mean average payoff achieved by obedient nodes and bad voters. It can be observed that obedient peers achieve the highest mean average payoff over time amounting to 0.97 whilst that of the bad voters is close to zero. Fig. 3(c) measures the *service load share* of both strategy types. This metric determines in each time slot the fraction of peers that provided and were able to receive a service, subject to a distinct user group. It can be seen that nearly all obedient peers are continuously able to receive a service whereas only 5-7% of the bad voters are supplied with data. To explain this, Fig.3(d) plots the mean ratio of successful requests experienced by bad voters while varying the similarity thresholds applied by obedient nodes. That is, this ratio measures how often a bad voter was unrecognized when requesting a service by an obedient provider, related to all send requests. For $t = 0.7$ on average 97% of all request carried out by bad voters are detected when applying the similarity function. Accordingly, bad voters are almost never served by obedient nodes.

The experiments with *colluders* strengthens our findings that the similarity-based comparison of global trust vectors very efficiently detects nodes trying to compromise the system. Fig. 3(e) plots the achieved payoffs of both colluders and obedient peers. As in our previous experiments, the discriminator strategy clearly dominates the attacker strategy. Colluders are quickly detected as the trust vectors of both user groups highly differ from each other. In fact, the determined overall similarity between the trust vectors of both users groups is on average 0.23. Accordingly, obedient peers do not trust ratings submitted by colluding peers but favour honest peers. To confirm this, Fig. 3(f) plots the total amount of consumers that have been rejected by obedient providers after 50 transactions. At this point of time, nearly all malicious consumers are rejected by obedient providers, irrespective of the size of the malicious collective. Instead, the number of rejects to obedient consumers is nearly zero in all simulated scenarios.

We conclude that the usage of the similarity function enables the system to efficiently filter out spurious reports from malicious nodes. Moreover, bad voters are immediately punished when submitting false reports on obedient nodes; since their global trust vectors very quickly deviate to the one of peers conforming to system's norm, they are immediately rejected by these nodes.

6 Conclusion

This paper has investigated the correlations between p2p environments and cooperation in human society. Through this, a new reputation-based incentive scheme has been designed, utilizing extremely limited binary reputation representations. Alongside this, we have also proposed a fully decentralized reputation infrastructure capable of securely managing reputations and protecting against malicious collusion and false reports. This approach was evaluated, through simulation, showing that nearly all peers wishing to gain services must contribute to the system, eliminating free-riding. It was further shown that malicious peers, solely interested in disrupting the network, were also quickly ostracized.

There are a number of areas of future work. Firstly, detailed overhead studies are necessary to investigate the impact that utilizing such a scheme has on the overall system. Further investigation into improving the infrastructure is also planned to protect against extremely high levels of malicious users ($> 50\%$) of the replica set. Lastly, more detailed evaluative scenarios will be performed to investigate the reliability of the infrastructure against bad voters, especially if these nodes selectively or randomly change their misbehavior per transaction.

Acknowledgement

The collaboration on this paper has been funded through the European Network of Excellence CONTENT, FP6-0384239.

References

1. E. Adar, B. Huberman: Free riding on gnutella. First Monday (2000)
2. S. Saroiu, P. Gummadi, S. Gribble: A measurement study of p2p file sharing systems. Technical report, Washington University (2002)
3. G. Harding: Tragedy of commons. Science (1968)
4. M. A. Nowak, K. Sigmund: Evolution of indirect reciprocity. Nature (2005)
5. O. Leimar, P. Hammerstein: Evolution of cooperation through indirect reciprocation. Proc. R. Soc. Lond. (2001)
6. H. Othsuki, Y. Iwasa: How should we define goodness? - reputation dynamics in indirect reciprocity. Journal of Theoretical Biology (2004)
7. Z. Zhang, S. Chen, M. Yoon: MARCH: A distributed incentive scheme for p2p networks. In: INFOCOM. (2007)
8. Jakobsson et al.: A micro-payment scheme encouraging collaboration in multi-hop cellular networks. Lecture Notes in Computer Science (2003)
9. Wilcox-O'Hearn: Experiences deploying a large-scale emergent network. In: IPTPS. (2002)
10. B. Cohen: Incentives build robustness in bittorrent. Technical report (2003)
11. A. Habib, J. Chuang: Service differentiated peer selection: an incentive mechanism for p2p media streaming. IEEE Transactions on Multimedia (2006)
12. M. Srivatsa, L. Xiong, L. Liu: Trustguard: countering vulnerabilities in reputation management for decentralized overlay networks. In: WWW. (2005)
13. E. Damiani et al.: A reputation-based approach for choosing reliable resources in peer-to-peer networks. In: CCS. (2002)
14. L. Xiong, L. Liu: Peertrust: Supporting reputation-based trust for peer-to-peer electronic communities. IEEE Trans. on Knowledge and Data Engineering. (2004)
15. S. Kamvar, M. Schlosser, H. Garcia-Molina: The eigentrust algorithm for reputation management in P2P networks. In: WWW. (2003)
16. S. Lee, R. Sherwood, B. Bhattacharjee: Cooperative peer groups in nice. In: INFOCOM. (2003)
17. M. Feldman, K. Lai, I. Stoica, J. Chuang: Robust incentive techniques for p2p networks. In: CECOMM. (2004)
18. E. Eftstathiou, P. Francgoudis, G. Polyzos: Stimulating participation in wireless community networks. In: INFOCOM. (2006)
19. M. A. Nowak, K. Sigmund: Evolution of indirect reciprocity by image scoring. Nature (1998)
20. I. Stoica et al.: Chord: A scalable p2p lookup service for internet applications. In: SIGCOMM. (2001)