HAL open science

Incentive-Based Robust Reputation Mechanism for P2P Services

Emmanuelle Anceaume, Aina Ravoaja

To cite this version:

Emmanuelle Anceaume, Aina Ravoaja. Incentive-Based Robust Reputation Mechanism for P2P Services. International Conference on Principles of Distributed Systems (OPODIS), 2006, France. pp.12. hal-00916744

HAL Id: hal-00916744

https://hal.science/hal-00916744v1

Submitted on 10 Dec 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Incentive-based Robust Reputation Mechanism for P2P Services

Emmanuelle Anceaume and Aina Ravoaja*

IRISA, Campus Universitaire de Beaulieu, Rennes, France
{anceaume,aravoaja}@irisa.fr

Abstract. In this paper, we address the problem of designing a robust reputation mechanism for peer-to-peer services. The mechanism we propose achieves high robustness against malicious peers (from individual or collusive ones) and provides incentive for participation. We show that the quality of the reputation value of trustworthy and participating peers is always better than the one of cheating and non participating ones. Finally we formally prove that, even when a high fraction of peers of the system exhibits a collusive behavior, a correct peer can still compute an accurate reputation mechanism towards a server, at the expense of a reasonable convergence time.

Key words: Reputation, credibility, free-riding, collusion, peer-to-peer systems

1 Introduction

With the emergence of e-commerce in open, large-scale distributed marketplaces, reputation systems are becoming attractive for encouraging trust among entities that usually do not know each other. A reputation system collects, distributes, and aggregates feedback about the past behavior of a given entity. The derived reputation score is used to help entities to decide whether a future interaction with that entity is conceivable or not. Without reputation systems, the temptation to act abusively for immediate gain can be stronger than the one of cooperating. In closed environments, reputation systems are controlled and managed by large centralized enforcement institutions. Designing reputation systems in P2P systems has to face the absence of such large and recognizable but costly organizations capable of assessing the trustworthiness of a service provider. The only viable alternative is to rely on informal social mechanisms for encouraging trustworthy behavior [9]. Proposed mechanisms often adopt the principle that "you trust the people that you know best", just like in the word-of-mouth system, and build transitivity trust structures in which credible peers are selected [19–21]. However such structures rely on the willingness of entities to propagate information. Facing free-riding and more generally under-participation is a well known problem experienced in most open infrastructures [2]. The efficiency and accuracy of a reputation system depends heavily on the amount of feedback it receives from participants. According to a recognized principle in economics, providing rewards is an effective way to improve feedback. However rewarding participation may also increase the incentive for providing false information.

* Aina Ravoaja is partially supported by a grant from Brittany region

Thus there is a trade-off between collecting a sizable set of information and facing unreliable feedback [7]. An additional problem that needs to be faced with P2P systems, is that peers attempt to collectively subvert the system. Peers may collude either to discredit the reputation of a provider to lately benefit from it (bad mouthing), or to advertise the quality of service more than its real value to increase their reputation (ballot stuffing). Lot of proposed mechanisms break down if raters collude [8].

In this paper we address the robust reputation problem. Essentially this problem aims at motivating peers to send sufficiently honest feedback in P2P systems in which peers may free-ride or be dishonest. This work has been motivated by a previous one in which the proposed architecture is built on top of a supervising overlay made of trusted peers [3]. The mechanism we propose achieves high robustness to attacks (from individual peers or from collusive ones), and provides incentive for participation. This is accomplished by an aggregation technique in which a bounded number of peers randomly selected within the system report directly observed information to requesting peers. Observations are weighted by a credibility factor locally computed. Incentive for participation is implemented through a fair differential service mechanism. It relies on peer's level of participation, a measure of peers' contribution over a fixed period of time, and on the credibility factor, assessing the confidence one has in a peer.

Our results are promising: We prove that through sufficient and honest cooperation, peers increase the quality of their reputation mechanism. We show that the reputation estimation efficiently filters out malicious behaviors in an adaptive way. Presence of a high fraction of malicious peers does not prevent a correct peer from computing an accurate reputation value, at the expense of a reasonable convergence time. Furthermore, the trade-off between the sensitivity of the mechanism facing up malicious peers and the duration of the computation is tuned through a single input parameter. These properties, combined with the incentive scheme, makes our mechanism adapted to P2P networks. Finally, we provide a full theoretical evaluation of our solution. For space reasons proofs of correctness are given in the full version of the paper [4].

The rest of the paper is organized as follows. In Section 2 related work is reviewed. Section 3 presents the model of the environment, and the specification of the robust reputation problem. Section 4 presents the incentive-based mechanism. Section 5 analyses its asymptotic behavior, its resistance to undesirable behavior and its convergence time.

2 Related Work

There is a rapidly growing literature on the theory and applications of reputation systems, and several surveys offer a large analyze of the state of art in reputation systems [15, 11, 8]. According to the way ratings are propagated among entities and the extent of knowledge needed to perform the needed computations, reputation systems fall into two classes, namely centralized or distributed. An increasing number of online communities applications incorporating reputation mechanisms based on centralized databases has recently emerged. The eBay rating system used to find traders allows partners to rate each other after completion of an auction. Despite its primitive reputation system, ebay is the largest person-to-person online auction with more than 4 millions auctions open at a time [16]. Regardless of this success, centralized approaches (see for exam-

ple, [21, 18]) often pay little attention to misbehaving entities by assuming that entities give honest feedback to the requesting entity. More importantly, they rarely address non-participation and collusive behaviors.

Regarding decentralized p2p architecture, several research studies on reputation-based P2P systems have emerged. Among the first ones, Aberer and Despotovic [1] propose a reputation mechanism in which trust information is stored in P-Grid, a distributed hash table-based (DHT) overlay. Their mechanism is made robust by guaranteeing that trust information is replicated at different peers, and thus can be accessed despite malicious entities. However, the efficiency of their approach relies on peers propensity to fully cooperate by forwarding requests to feed the P-Grid overlay. Additionally, as for most of the DHT-based approaches, peers have to store data they are not concerned with. Thus, malicious peers may discard it to save private resources, leading to a loss of information. Other systems relying on the trust transitivity approach face false ratings by assuming the presence of specific faithful and trustworthy peers (e.g. [13]), or by weighting second-hand ratings by senders' credibility [7, 19, 20]. Opposed to the aforementioned works, Havelaar reputation system [10], exploits long-lived peers by propagating reports between sets of well defined peers identified through hash functions. A report contains the observations made during the current round, the aggregated observations made by the predecessors during the previous round, and so on for the last r rounds. By relying on such an extensive aggregation, false reports hardly influence the overall outcome. Furthermore by using hash functions collusion is mostly prevented. The efficiency of their approach mainly relies on the readiness of peers to store and propagate large amount of data, and to remain in the system for relatively long periods of time. To motivate peers to participate, Jurca and Faltings [12] propose an incentive-compatible mechanism by introducing payment for reputation. A set of brokers, the R-agents, buy and sell feedback information. An entity can receive a payment only if the next entity reports the same result. Weakness of such an approach is the centralization of the whole information at R-agents, and its robustness against malicious R-agents. Finally, Awerbuch et al. [5, 6] give lower bounds on the costs of the probes made by honest peers to find good objects in eBay-like systems, and propose algorithms that nearly attain these bounds.

In contrast to these works, we propose a fully distributed mechanism based on local knowledge that provides malicious and non-participating entities an incentive for participation and honest behavior.

3 Model

3.1 A P2P Service Model

We consider a P2P service system where *service providers* (or *servers*) repeatedly offer the same service to interested *peers*. We assume that the characteristics of a server (capabilities, willingness to offer resources, etc) are aggregated into a single parameter θ called type. This type influences the *effort* exerted by the server through a cost function c. The effort determines the Quality of the Service (QoS) provided by the server. We assume that the effort exerted by a server is the same for all the peers that solicit him and takes its value within the interval $[0, 1]$.

Definition 1 (Effort). *The effort of a service provider s is a value q_s^* that determines the quality of the service offered to the peers that interact with s.*

After each interaction with server s, each client (or peer) has an imperfect observation of the effort exerted by s. Peers may have different tastes about a server QOS. But basically these observations are closely distributed around s's effort. Thus, we reasonably assume that an observed quality of service takes its value within the interval $[0, 1]$ and follows a normal distribution of mean q_s^* and variance σ_s^*.

Definition 2 (Observed Quality of Service Level). *The Observed Quality of Service Level of a service provider s observed by peer p at time t is a value $obs_p^s(t)$ which is drawn from a normal distribution over $[0, 1]$ with mean q_s^*. The value 1 (resp. 0) characterizes the maximal (resp. minimal) satisfaction of p.*

Estimation of the expected behavior of a server is based on its recent past behavior, that is, its recent interactions with the peers of the system. Such a restriction is motivated by game theoretic results and empirical studies on ebay that show that only recent ratings are meaningful [8]. Thus, in the following, only interactions that occur within a sliding window of width D [1] are considered. This approach is also consistent with Shapiro's work [17] in which it is proven that in an environment where peers can change their effort over time the efficiency of a reputation mechanism is maximized by giving higher weights on recent ratings and discounting older ratings. Using a sliding time window is approximately equivalent to this model. Every time a peer p desires to interact with a server s, p asks for feedback from peers that may have directly interacted with s during the last D time units. We adopt the terminology *witness* to denote a peer solicited for providing its feedback. If $\mathcal{P}_p^s(t)$ represents the set of peers k whose feedback has been received by peer p by time t, then the reputation value of a server is defined as follows:

Definition 3 (Reputation Value). *The reputation value $r_p^s(t)$ of server s computed by peer p at time t is an estimation of the effort q_s^* exerted by s based on the feedbacks provided by the peers in $\mathcal{P}_p^s(t)$.*

3.2 Specification of Undesirable Behaviors

In practice, peers may not always reveal their real ratings about other peers. They can either exaggerate their ratings (by increasing or decreasing them), or they can simply reveal outright ratings to maximize their welfare. This behavior is usually called *malicious*, and can either be exhibited by a node independently from the behavior of other peers, or be emergent of the behavior of a whole group. By providing false ratings, malicious peers usually try to skew the reputation value of a server to a value which is different from its true effort. Let $\overline{q}$ be this value, and d be the distance between the true effort of the server and the false rating ($d = |q_s^* - \overline{q}|$). Then, we characterize the behavior of a peer by $w_q^s = 1 - d^\alpha$, with α a positive real value which represents the sensitivity of w_q^s to the distance between the effort and the expected observation given by q. A malicious peer tries to skew the reputation value to $\overline{q}$ by sending ratings that are distributed around $\overline{q}$.

[1] D can have any pre-defined length of time, i.e., a day, a week or a month. In the sequel, we suppose that D is insensitive to clock drift.

Definition 4 (Malicious). *A peer p is called malicious if it lies on the reputation of peer s or creates s's reputation out of thin air. Formally :*

$$\mathrm{E}(obs_p^s(t)) = \overline{q} \neq q_s^*, \forall t.$$

Definition 5 (Collusive Group). *A group of peers is called a collusive group if all the peers of this group behave maliciously towards a same goal. Formally, the set $\mathcal{C}$ is a colluding group if :*

$$\mathrm{E}(obs_k^s(t)) = \overline{q} \neq q_s^*, \forall t, k \in \mathcal{C}.$$

Another common behavior in P2P systems is peers non-participation. There are two main reasons why peers may not participate: either because they believe that their work is redundant with what the others in the group can do, and thus their participation can hardly influence the group's outcome, or because they believe that by not contributing they maximize their own welfare (note that information retention could be another pretense of not participating, however this is out of the scope of the paper). The latter behavior depicts what is typically called *free-riding*, while the first one is described in the Collective Effort Model (CEM) as *social loafing* [14]. Note that although effects of both behaviors are similar, i.e., "non-participation", their deep cause is different. Peers exhibiting one of these two behaviors are called in the following *non-participating* peers and are characterized as follows:

Definition 6 (Non Participating). *A peer is called* non participating *if it exerts less effort on a collective task than it does on a comparable individual task or consumes more than its fair share of common resources.*

A peer is called *correct* if during the time it is operational in the system it is neither malicious nor non-participating. Note that a malicious peer may not participate, on the other hand, a non participating one is not malicious.

3.3 Specification of the Robust Reputation Problem

Within this context, we address the problem of evaluating the reputation of a service provider in a dynamic environment in which peers are not necessary correct. This problem is referred in the sequel as the *robust reputation problem*. A solution to this problem should guarantee the following two properties. The first one states that eventually correct peers should be able to estimate the reputation value of a target server with a good precision. The second one says that with high probability, correct peers have a better estimation of the reputation value of a target server than non correct ones. Formally:

Property 1 (Reputation Value ϵ-Accuracy). Eventually, the reputation of server s, evaluated by any correct peer reflects s's behavior with precision ϵ. That is, let $\beta \in]0, 1[$ be some fixed real, called in the sequel confidence level, then:

$$\exists \overline{t} \; s.t. \; \forall t \geq \overline{t}, \; \mathrm{Prob}(|r_p^s(t) - q_s^*| \leq \epsilon) \geq 1 - \beta$$

Let $|\mathrm{E}(r_p^s(t)) - q_s^*|$ be the bias of the reputation value $r_p^s(t)$ estimated by peer p. Suppose that two peers p and q interact with the same target servers at the same time, solicit the

same witnesses, and get the same feedbacks at the same time. That is from the point of view of their interaction p and q are indistinguishable. However p is correct while q is not. Then, we have:

Property 2 (Incentive-Compatibility). Eventually, the bias of the reputation value of server s estimated by p is greater than or equal to the one estimated by peer q. That is, for a given level of confidence β, we have:

$$\exists \overline{t}\ s.t.\ \forall t \geq \overline{t},\ \mathrm{Prob}(|\mathrm{E}(r_p^s(t)) - q_s^*| \geq |\mathrm{E}(r_q^s(t)) - q_s^*|) \geq 1 - \beta$$

4 The Reputation Mechanism

We propose a distributed reputation service which builds a social network among peers. Briefly, every peer records the opinion about the late experiences it has had with a target server. Peers provide their information on request from peers willing to interact with that server. Providing a feedback based on direct observations (also called first-hand observations) prevents the *rumors* phenomenon, i.e., the propagation of opinions about others, just because they have been heard from someone else [19], however is better adapted to applications with modest churn. Upon receipt of ”enough” feedback, the requesting peer aggregates them with its own observations (if any) to estimate the reputation of the target server, and provides this estimation to its application. Information is aggregated according to the trust the requesting peer has in the received feedback. Pseudo-code of the reputation mechanism is presented in Algorithm 1. The efficiency of the reputation mechanism fully depends on *i)* the number of received feedbacks (i.e., aggregating few feedbacks is not meaningful and thus not helpful), and *ii)* the quality of each of them (i.e., the trustworthiness of the feedback). The contribution of this work is the design of a reputation mechanism that enjoys both properties. The analysis presented in Section 5 shows the importance of each factor on the convergence time and accuracy of the reputation mechanism.

The solution we propose is a reputation mechanism, and therefore independent of the rewarding strategy used by the application built on top of this mechanism. That is, the willingness of a peer to interact with a server results from the application strategy, not from the peer’s one. Clearly, the strategy of the application is greatly influenced by the reputation value but other factors may also be taken into account.

4.1 Collecting Feedbacks

When a peer decides to evaluate the reputation value of a service provider, it asks first-hand feedback from a set of witnesses in the network. Finding the right set of witnesses is a challenging problem since the reputation value depends on their feedback. Our approach for collecting feedbacks follows the spirit of the solution proposed by Yu et al [20], in which feedbacks are collected by constructing chains of referrals through which peers help one another to find witnesses. We adopt the walking principle. However, to minimize the ability of peers to collude, witnesses are randomly chosen within the system. We assume, in the following, that the network is regular. Specifically, our approach is based on a random walk (RW) sampling technique. We use the random

walk technique as shown in Algorithm 1. Function `query` is invoked by the requesting peer that wishes to solicit x witnesses through r random walks bounded by ttl steps. The requesting peer starts the random walks at a subset of its neighbors, and runs them for ttl steps. Each peer p involved in the walk is designated as witness, and as such sends back to the requesting peer its feedback. When a peer q receives a request from p to rate server s, it checks whether during the last sliding window of length D, it has ever interacted with s. In the affirmative, p sets its feedback to $F_p^s(t) = \{(obs_p^s(t_0), t_0), \ldots, (obs_p^s(t_l), t_l)\}$ with $obs_p^s(t_i)$ the QoS of s observed at time t_i, where $t_i \in [\max(0, t-D), t]$. In case p has not recently interacted with s, p sends back to q a default feedback $F_p^s(t) = \{(obs_{max}, \perp)\}$. As will be shown later, this feedback prevents p from being tagged "non participant" by q.

Because of non-participation (volunteer or because of a crash), random walks may fail: it suffices that one of the peers in the walk refuses to participate or crashes to prevent the walk from successfully ending. If we assume that among its d neighbors, a fraction μ of them do not participate, then among the r initial peers that start a random walk, the expected number of peer that may "fail" their random walk is μr. Then during the next step, $\mu(1-\mu)r$ walks may "fail", and so on until the TTL value is reached. In consequence, only x feedbacks may be received, with $x = \sum_{t=1}^{ttl}(1-\mu)^t r$. By setting r to $\frac{x}{\sum_{t=1}^{ttl}(1-\mu)^t}$ the requesting peer is guaranteed to receive at least x feedbacks (see line 5 in Algorithm 1). In addition to its feedback, each peer sends to the requesting peer p (through the `witness` message, see lines 19 and 46) the identity of the next potential witness on the walk, i.e., the peer it has randomly chosen among its neighbors. Sending this piece of information allows p to know the identity of all potential witnesses. As will be shown in Section 4.4, this allows to detect non participants (if any) and so to motivate their participation through a "tit-for-tat" strategy. As for feedbacks, non participation may prevent the requesting peer from receiving `witness` messages. A similar analysis to the preceding one shows that if r random walks are initiated then $y = \sum_{t=0}^{ttl-1}(1-\mu)^t r$ `witness` messages will be received.

Note that a requesting peer can adapt its collect policy according to its knowledge of the target server, or of its neighborhood. Specifically, to get x witnesses, a peer can either increase ttl and restrict r, or increase r and lower ttl (assuming that $r > \mu d$ holds). Enlarging ttl would be more sensitive to colluding peers that bias the random walk. However, this technique would increase the set of crawled witnesses, and thus would afford new peers the opportunity to be known by other peers and consequently to increase both their participation and their credibility. Conversely, enlarging r would crawl only peers in the neighborhood of the requesting peer. However, this technique would increase the chance to find a path that does not contain colluding peers [2].

[2] Remark that selecting peers according to their credibility should be more efficient in the sense that only "highly" credible peers would be selected, however, newcomers may be penalized by this filtering. Furthermore, the resilience of the crawling technique to collusion highly relies on the way the graph of witnesses is constructed. Studying these issues is part of our future work.

4.2 Reputation of a Server

Estimation of the reputation value of a target server is based on the QoS directly observed at the server (if any) and on the feedbacks received during the collect phase. The accuracy of the estimation depends on the way these informations are aggregated. The aggregation function we propose answers the following qualitative and quantitative preoccupations: First, to minimize the negative influence of unreliable information, feedbacks are weighted by the credibility of their senders. Briefly, credibility is evaluated according to the past behavior of peers and reflects the confidence a peer has in the received feedback. Credibility computation is presented in the next subsection. Second, to prevent malicious nodes from flooding p with fake feedback and thus from largely impacting the accuracy of its estimation, p keeps only a subset of each received feedback. More precisely, among the set of observations sent by each witness over the last D time units, only the last f ones are kept, with f the size of the smallest non-empty set of non-default feedbacks received by p (i.e., $f = \min_{k \in \mathcal{P}_p^s(t)}(|F_k^s(t)|)$ with $t \in [\max(0, t-D), t]$). Finally, if among all the witnesses (including p) none has recently directly interacted with s (i.e., $f = 0$), then p affects a maximal value obs_{max} to s's reputation value. Affecting a maximal value reflects the key concept of the Dempster-Shafer theory of evidence which argues that "there is no causal relationship between a hypothesis and its negation, so lack of belief does not imply disbelief". In our context, applying this principle amounts in fixing an *an priori* high reputation to unknown servers, and then updating the judgment according to subsequent interactions and observations [20].

We can now integrate these principles within the aggregation function we propose. Let us first introduce some notations: Let $\mathcal{F}_k^s(t)$ be the union of the last f non-default feedbacks received from k during the last D time units ($t \in [\max(0, t-D), t]$); $\mathcal{P}_p^s(t)$ be the set of witnesses k for which $\mathcal{F}_k^s(t)$ is non empty; $\rho_k^s(t)$ represent the mean value of the observations drawn from $\mathcal{F}_k^s(t)$; and $c_{p,k}^s(t)$ the credibility formed by p at time t about k regarding s. Then, at time t, p estimates s's reputation value as follows:

$$r_p^s(t) = \begin{cases} \frac{1}{\sum_{k \in \mathcal{P}_p^s(t)} c_{p,k}^s(t)} \sum_{k \in \mathcal{P}_p^s(t)} c_{p,k}^s(t).\rho_k^s(t) & \text{if } f \neq \emptyset \\ obs_{max} & \text{otherwise} \end{cases} \quad (1)$$

with, $$\rho_k^s(t) = \frac{1}{f} \sum_{(obs_k^s(t'), t') \in \mathcal{F}_k^s(t)} obs_k^s(t')$$

4.3 Trust in Witnesses

In this section, we tackle the issue of malicious peers. As remarked in the Introduction, malicious peers may alter the efficiency of the reputation mechanism by sending feedbacks that over-estimate or sub-estimate the observed QoS of a server to inflate or tarnish its reputation. This is all the more true in case of collusion. We tackle this issue by evaluating peers credibility. Credibility is a [0,1]-valued function which represents the confidence formed by peer p about the truthfulness of q's ratings. This function is local and is evaluated on the recent past behavior of both p and q peers. It is locally used to prevent a false credibility from being propagated within the network. Specifically, peer p estimates at time t how credible q is regarding server s as a decreasing

function of the distance between q's feedbacks on s's effort and p's direct observations on s's QoS. As for the reputation value computation, the distance is computed on the last f observations made by both p and q during the last D time units. Note that in case p has not recently observed s's QoS, then credibility of all its witnesses are set to a default value c_0. Indeed, p cannot evaluate the distance between its own observations and those observed by witnesses. Determining c_0 value needs to solve the following trade-off: by affecting a high value to the default credibility one increases the vulnerability of the system to the *whitewashing* phenomenon, that is, the fact that peers change their identity in order to reset their credibility to the default value. However, by setting this variable to a low value the mechanism tends to filter out new witnesses and thus, loses the benefit of the potential information a new peer can afford, which clearly decreases the usefulness of the reputation mechanism. In order to cope with that, we set c_0 to the value of a decreasing function of ϕ, with ϕ an estimation of the number of whitewashers in the network. By adopting the notations of Equation 1, $c^s_{p,q}(t)$ represents the credibility formed by p at time t about q regarding the target server s, and is given by:

$$c^s_{p,q}(t) = \begin{cases} 1 - |\rho^s_q(t) - \rho^s_p(t)|^\alpha & \text{if } f \neq \emptyset \\ c_0 & \text{otherwise} \end{cases} \tag{2}$$

where $|\rho^s_q(t) - \rho^s_p(t)|^\alpha$ represents the distance between q and p's observations. Note that α is the variable introduced in Section 3. Then we have the following lemma:

Lemma 1. ***(Credibility Accuracy)*** *Eventually, credibility of a peer q evaluated by any correct peer p reflects q's behavior with a precision ϵ. That is, let $\beta \in]0,1[$ be some fixed real, there exists $\overline{t}$ such that, for all $t \geq \overline{t}$,*

$$\text{Prob}(|c^s_{p,q}(t) - w^s_q(t)| \leq \epsilon) \geq 1 - \beta.$$

4.4 Incentive for Participation

Non participation may jeopardize the efficiency of the reputation mechanism. A certain amount of participation is required before reputation can induce a significant level of cooperation. Facing non-participation in the reputation problem is challenging and has deserved few attention [20]. To motivate peers to send their feedback we adopt a "tit-for-tat" strategy. We introduce the level of participation notion as the propensity of a peer for replying to a rating request. It is described by function $l^s_{p,q}$ such that $l^s_{p,q}(t)$ represents the percentage of times q provided its feedback to p's queries regarding server s's QoS over the last D time units, with $l^s_{p,q}(t = 0) = l_0 = 1$. Its computation is performed after p's collect phase (see line 10 of the algorithm. Note that factor μ prevents correct peers from being penalized by walking breaks.).

We apply the tit-for-tat strategy during the collect phase. When a peer p receives a rating request for s from peer q, then with probability $l^s_{p,q}(t)$ p provides its feedback to q, otherwise it sends a default feedback $(\perp, \perp)$ to prevent p from being tagged as non-participant. By providing this default feedback, p lets q knows that its recent non-participation has been detected. Consequently, by not participating, requesting peers drive correct witnesses providing them worthless feedback, which clearly makes their reputation mechanism useless. Hence there is a clear incentive for non participating

peers to change their behavior. The following lemma proves that participation decreases the bias of the reputation value. As previously, let us consider two peers p and q such that both peers are indistinguishable from the point of view of the servers with which they interact, that is both p and q observe the same QoS from these servers at the very same time, solicit and are solicited by the same set of peers at the same time; However, p is correct while q is non participating. Then we claim that:

Lemma 2. *Participation decreases the bias of the reputation value. That is,*

$$|\mathrm{E}(r_p^s(t)) - q_s^*| \leq |\mathrm{E}(r_q^s(t)) - q_s^*|$$

4.5 Incentive for Truthful Feedbacks

We now address the problem of motivating peers to send truthful feedbacks. So far we have presented strategies aiming at improving the quality of the reputation value estimation by aggregating more feedbacks and by weighting feedback according to the credibility of their sender. We have shown that by using both strategies, utility of correct peers increases. However, none of these solutions have an impact on the effort devoted by a witness to send a truthful feedback. To tackle this issue we use the credibility as a way to differentiate honest peers from malicious ones. As for non-participating peers, when peer p receives a request to rate server s from a requesting peer q then p satisfies q's request with probability $c_{p,q}^s(t)$. By doing so, p satisfies q's request if it estimates that q is trustworthy, otherwise it notifies q of its recent faulty behavior by sending it the $(\perp, \perp)$ feedback. As previously, by cheating, a malicious peer penalizes itself by pushing correct witnesses to send meaningless feedbacks to it, leading to its effective isolation. We claim that this social exclusion-based strategy motivates q to reliably cooperate.

Lemma 3. *High credibility decreases the bias of the reputation value. That is,*

$$|\mathrm{E}(r_p^s(t)) - q_s^*| \leq |\mathrm{E}(r_q^s(t)) - q_s^*|$$

Finally, to elicit sufficient and honest participation, both strategies are combined, i.e., upon receipt of a rating request from peer q, with probability $\min(c_{p,q}^s(t), l_{p,q}^s(t))$ p provides its feedback, otherwise it sends the default feedback $(\perp, \perp)$ (see line 31 in Algorithm 1).

Theorem 1. *The reputation mechanism described in Algorithm 1 is Incentive-Compatible in the sense of Property 2.*

5 Analysis

Computing the reputation of a peer reduces to estimating, in the statistical sense, its effort. Our algorithm falls into the category of robust estimation algorithms. Indeed, robust estimation techniques consider populations where a non-negligible subset of data deliberately pollute the system. This analysis describes the asymptotic behavior of the reputation mechanism and its convergence time according to undesirable behaviors. In the following, we assume that a fraction γ of witnesses are malicious.

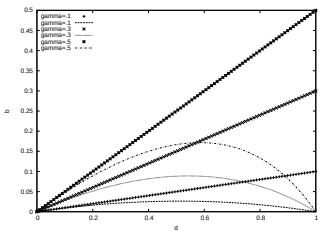

Fig. 1. Bias for $\alpha = 1$ and $n = 10$.

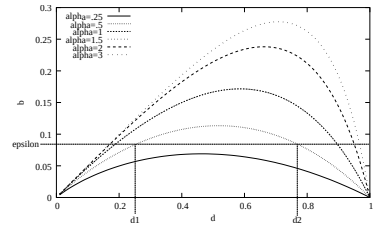

Fig. 2. Bias for $\gamma = .5$ and $n = 10$.

5.1 Asymptotic Behavior

In this section we determine the accuracy of the reputation mechanism with respect to parameters α and ϵ. Thus for the purpose of this analysis, we assume that the number of aggregated feedbacks f is infinite. Recall that w_q denotes the characterization of q's behavior, with $w_q = 1$ if q is correct, and $w_q = 1 - d^\alpha$ otherwise. By Lemma 1, $c^s_{p,q}(t) = w_q$, when $t \to \infty$. Moreover, the expected number of correct witnesses is $(1-\gamma)n+1$ while the expected number of malicious ones is γn, with n the expected number of witnesses (Figures are plotted for $n = 10$). Thus, by replacing $c^s_{p,q}(t)$ with their asymptotic values in Equation 1, the expected reputation value of a server s estimated by a correct peer p when $t \to \infty$ is given by Equation 3:

$$r^s_p(t)_{t\to\infty} = \frac{1}{1-\gamma d^\alpha + \frac{1}{n}} \left(\left(1 - \gamma + \frac{1}{n}\right) q^*_s + \gamma \left(1 - d^\alpha\right) \overline{q} \right). \tag{3}$$

The bias of the reputation value, when $t \to \infty$ is given by the following equation:

$$|r^s_p(t)_{t\to\infty} - q^*_s| = \gamma d \frac{1 - d^\alpha}{1 - \gamma d^\alpha + \frac{1}{n}}. \tag{4}$$

Figures 1 and 2 show the bias of the reputation value with respect to d for increasing values of γ (resp. increasing values of α). Recall that $d = |q^*_s - \overline{q}|$ reflects colluders' behavior. Unlike mean-based reputation value estimation for which the bias linearly increases with d (as shown by the crossed curves in Figure 1), our algorithm bounds the power of colluders whatever their percentage (dotted curves). Indeed, witnesses' ratings are weighted by a decreasing function of d which filters out false ratings.

Figure 2 shows the impact of α on the bias of the reputation value. As can be observed, the bias decreases with decreasing values of α, reflecting the sensitivity of the reputation value to the distance between direct observations and received feedbacks. Thus, decreasing values of α makes the reputation mechanism very sensitive to false feedbacks.

Theorem 2. *The reputation value is ϵ-accurate, with $\epsilon > \gamma d \frac{1-d^\alpha}{1-\gamma d^\alpha + \frac{1}{n}}$.*

From the above, assuming an upper bound on γ, by setting the maximal bias to ϵ and solving the corresponding equation one can derive an upper bound on α under which the reputation value converges to the true effort with an accuracy level of ϵ. Hence, for

α with $\alpha \leq \overline{\alpha}$, the reputation value converges to the true effort with an accuracy level of ϵ, with $\overline{\alpha}$ given by:

$$\overline{\alpha} = \frac{\ln\left(\frac{n+1-\sqrt{(1-\gamma)n^2+(2-\gamma)n+1}}{\gamma n}\right)}{\ln(\epsilon)} \tag{5}$$

To conclude, one can always find a value of α such that eventually the reputation value is accurate. This parameter, however, significantly influences the convergence time of the algorithm. The next Section addresses this issue.

5.2 Convergence

In this section, we study the convergence time of the reputation mechanism. To do so, we assume that the ratings of a malicious peer are drawn from a normal distribution with mean $\overline{q}$ and variance $\overline{\sigma}$ over $[0, 1]$. This assumption includes a wide range of possible behaviors. Indeed, a small value of $\overline{\sigma}$ depicts peers that try to rapidly skew the reputation value to $\overline{q}$ by giving reports tightly distributed around $\overline{q}$. In contrast, a high value of $\overline{\sigma}$ depicts peers that try to hide their mischievous behavior to other peers by giving sparse reports. While the first behavior is easily detected, the second one hardly skew the reputation value to $\overline{q}$.

Recall that the reputation value is estimated by aggregating the last f interactions witnesses have had with the target server during a sliding time window of length D. Finding the optimal value of D is important. Indeed, it determines the resilience of the mechanism to effort changes and the confidence level in the estimation. The optimal value of D is the one for which the estimation is at most ϵ-far from the true effort with a given confidence threshold β. To determine such a value, let us first assume that f is known, and determine a lower bound on $\mathrm{Prob}(|r_p^s(t) - q_s^*| \leq \epsilon)\ \forall t \in D$. Suppose that the credibility $c_k^s(t)$ is ϵ'-far from w_k^s for all the witnesses k. Then, because of Bayes' Theorem, we know that $\mathrm{Prob}(|r_p^s(t) - q_s^*| \leq \epsilon) \geq \mathrm{Prob}(|r_p^s(t) - q_s^*| \leq \epsilon|_{|c_k^s(t)-w_k^s(t)|\leq\epsilon',\forall k\in\mathcal{P}_p^s(t)}) \cdot \mathrm{Prob}(|c_k^s(t) - w_k^s(t)| \leq \epsilon', \forall k \in \mathcal{P}_p^s(t))$. Remark that, assuming that the witnesses' credibility are ϵ'-far from w_k^s, the probability that $r_p^s(t)$ is ϵ-far from q_s^* is maximal under the following condition (C): credibility of correct witnesses is minimal, i.e., equal to $1 - \epsilon'$, and the one of malicious ones is maximal, i.e., equal to ϵ'. Then, we have:

$$\begin{aligned}&\mathrm{Prob}(|r_p^s(t) - q_s^*| \leq \epsilon) \geq \mathrm{Prob}(|r_p^s(t) - q_s^*| \leq \epsilon|_{(C)})\cdot\\ &\mathrm{Prob}(c_k^s(t) \geq 1 - \epsilon'|_{kcorrect})^{(1-\gamma)n} \cdot \mathrm{Prob}(c_k^s(t) \leq \epsilon'|_{kmalicious})^{\gamma n}\end{aligned} \tag{6}$$

By Lemma 1, the probability that the witness's credibility is at most ϵ'-far from w_k^s converges to 1 when t, and thus f, increase. Thus, this bound approaches $\mathrm{Prob}(|r_p^s(t) - q_s^*| \leq \epsilon)$ when f increases. Knowing the probability distribution of the reports, deriving a closed form of the lower bound can be done. Then, given a desired confidence threshold β for $\alpha \geq \overline{\alpha}$, solution of Equation 7 provides two threshold values of d ($d1$ and $d2$ on Figure 2) beyond which the false reports are eliminated:

$$\epsilon = \gamma d \frac{1 - d^\alpha}{1 - \gamma d^\alpha + \frac{1}{n}} \qquad \text{within } [0, 1]. \tag{7}$$

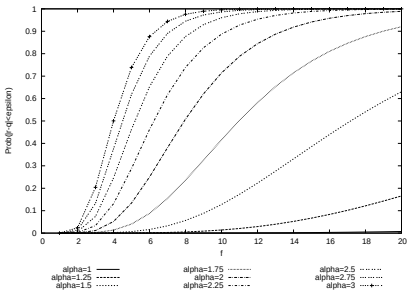

Fig. 3. Confidence level for $\gamma = .5$, $n = 10$, $\epsilon = .1$, $d = 1$, $\sigma_s^* = .2$, and $\overline{\sigma} = .2$.

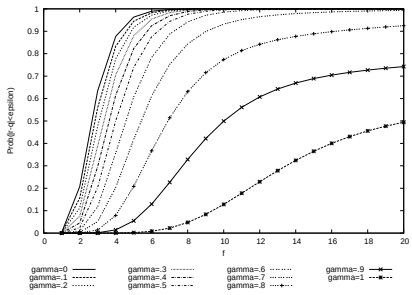

Fig. 4. Confidence level for $\alpha = 3$, $n = 10$, $\epsilon = .1$, $d = 1$, $\sigma_s^* = .2$, and $\overline{\sigma} = .2$.

Using Equation 6, we can see the effect of α on the convergence time of the reputation mechanism to reach the exact effort exerted by the server (see Figure 3). Decreasing values of α significantly increases the convergence time while it decreases the bias of the reputation mechanism as shown in Figure 2. Thus a trade-off exists between the robustness of the mechanism and its convergence time. By setting the value of D, the application designer may derive the corresponding minimal number of interactions f and finally tune the value of α such that the desired confidence level is achieved within f steps through Equation 6. The resulting reputation estimation is less sensitive to false reports, but still eliminates peers that try to skew the reputation of the server to a value that is far from the true effort, by filtering out extreme values of d (see Figure 2).

Finally, Figure 4 shows the impact of malicious peers on the convergence time. As can be seen, a relatively small percentage of malicious peers has a minor impact on the convergence time since the number of correct feedbacks is hardly influenced by false ones. On the other hand, whenever a requesting peer has to face a large proportion of malicious peers, it can only rely on its own feedback to estimate the effort exerted by the target server which clearly takes longer than when helped by correct witnesses. The same result applies for non-participating peers.

6 Conclusions

In this paper we have proposed a reputation mechanism that achieves high robustness to attacks and provides incentive for participation. This is achieved by an aggregation function in which a subset of the information provided by randomly chosen peers is kept and weighted by a confidence factor locally computed. We have proposed a simple and local incentive mechanism that guarantees a better quality of the reputation value estimation. Lessons learned from simulations are twofold: first, decreasing values of α guarantees a greater sensibility of the mechanism to false ratings. It however increases the number of required feedbacks as well, and thus the time to get an accurate estimation of the effort exerted by the target server. Second, the presence of a large number of malicious and non-participating peers does not prevent the mechanism from being accurate, however has an impact on its convergence time.

input : p: requesting peer; s: target server; x: expected number of feedbacks
output: r_p^s: estimation of s reputation value

$r \leftarrow \frac{x}{\sum_{l=1}^{ttl}(1-\mu)^l}$; $y \leftarrow \sum_{l=0}^{ttl-1}(1-\mu)^l r$;
$ttl \leftarrow$ default value; $t \leftarrow$ getTime(); let D be the the time interval [max(t-D,0),t];
`query(`*p,s,ttl,r,t*`)`;
wait until *(($F_q^s(t)$ messages are received from x peers) and ($witness(s,w,t)$ messages are received from y peers))*;
$P^s(t) \leftarrow \bigcup \{q$ such that a feedback message is received from $q\}$;
$F^s(t) \leftarrow \bigcup \{F_q^s(t)$ for all $q \in P(t)\}$;
$W^s(t) \leftarrow \bigcup \{w$ for all $witness(s,w,t)$ that have been received$\}$;
foreach $k \in W(t)$ **do**
    $l_{p,k}^s(t) \leftarrow \min(\frac{\sum_{t\in D}(|\mathcal{P}_{|k}(t)|}{\sum_{t\in D}|W_{|k}(t)|}(1-\mu) + \mu.l_0, 1)$;
end
return $r_p^s(t)$ to application;

`query(`*p,s,ttl,r,t*`)` **begin**
    $New \leftarrow$ pick a random subset of r peers from p's neighbors;
    forall *($next \in New$)* **do**
        send a rw (p, s, $ttl-1$,t) message to $next$;
        send a witness (s,$next$,t) message to p;
        if *p has interacted with s at time $t_0,\ldots,t_l$ in the last D time units* **then**
            $F_p^s(t) \leftarrow \{(obs_p^s(t_0),t_0),\ldots,(obs_p^s(t_l),t_l),p\}$;
        else
            $F_p^s(t) \leftarrow \{(obs_{max},\perp),p\}$;
        end if
        send $F_p^s(t)$ to p;
    end
end

upon *(receipt of a* rw *(p,s,ttl,t) message at peer q)* **do**
    with *(probability $\min(l_{p,q}^s(t), c_{p,q}^s(t))$)* **do**
        if *q has interacted with s at time $t_0,\ldots,t_l$ in the last D time units* **then**
            $F_q^s(t) \leftarrow \{(obs_q^s(t_0),t_0),\ldots,(obs_q^s(t_l),t_l),q\}$;
        else
            $F_q^s(t) \leftarrow \{(obs_{max},\perp),q\}$;
        end if
    otherwise
        $F_q^s(t) \leftarrow \{(\perp,\perp),q\}$;
    end do
    send $F_q^s(t)$ to p;
    if *($ttl \neq 0$)* **then**
        $next \leftarrow$ pick one of q's neighbor with probability $\frac{1}{d}$, $d = q$'s neighbors #;
        send a rw (p,s,$ttl-1$,t) message to $next$;
        send a witness (s,$next$,t) to p;
    end if
end do

Algorithm 1: Estimation of the reputation value of server s by peer p

References

1. K. Aberer and Z. Despotovic. Managing trust in a peer-to-peer inforation system. In *Proc. of the Tenth Int'l ACM Conference on Information and Knowledge Management (CIKM)*, 2001.
2. E. Adar and B. Huberman. The impact of free-riding on gnutella. Online at http://www.firstmonday.org/issues/issue5_10/adar/index.html, 2000.
3. E. Anceaume, M. Gradinariu, and A. Ravoaja. Incentives for p2p fair resource sharing. In *Proc. of the Fifth IEEE Int'l Conf. on Peer-to-Peer Computing (P2P)*, 2005.
4. E. Anceaume and A. Ravoaja. Incentive-based robust reputation mechanism for p2p services. Technical Report 1816, IRISA, 2006.
5. B. Awerbuch, B. Patt-Shamir, D. Peleg, and M. Tuttle. Collaboration of untrusting peers with changing interests. In *Proc. of the Fifth ACM Conference on Electronic Commerce (EC)*, 2004.
6. B. Awerbuch, B. Patt-Shamir, D. Peleg, and M. Tuttle. Adaptive collaboration in peer-to-peer systems. In *Proc. of the Twenty-Fifth IEEE International Conference on Distributed Computing Systems (ICDCS)*, 2005.
7. S. Buchegger and J.Y. Le Boudec. A robust reputation system for p2p and mobile ad-hoc networks. In *Proc. of the Second Workshop on the Economics of Peer-to-Peer Systems*, 2004.
8. C. Dellarocas. Reputation mechanisms. In T. Hendershott, editor, *Handbook on Information Systems and Economics*. Elsevier Publishing, 2006.
9. Z. Despotovic. *Building trust-aware P2P systems : from trust and reputation management to decentralized e-commerce applications*. PhD thesis, EPFL, 2005.
10. D. Grolimund, L. Meisser, S. Schmid, and R. Wattenhofer. Havelaar: A robust and efficient reputation system for active peer-to-peer systems. In *Proc. of the First Workshop on the Economics of Networked Systems (NetEcon)*, 2006.
11. A. Josang, R. Ismail, and C. Boyd. A survey of trust and reputation systems for online service provision. *Decision Support Systems*, 2006. Forthcoming. Accessible at http://sky.fit.qut.edu.au/ josang/publications.html.
12. R. Jurca and B. Faltings. An incentive compatible reputation mechanism. In *Proc. of the IEEE Conference on E-Commerce (CEC)*, 2003.
13. S. Kamvar, M. Schlosser, and H. Garcia-Molina. The eigentrust algorithm for reputation management in p2p networks. In *Proc. of the 12th Int'l Conference on World Wide Web (WWW)*, 2003.
14. S. Karau and K. Williams. Social loafing: A meta-analytic review and theoretical integration. *Journal of Personality and Social Psychology*, pages 681–706, 1993.
15. L. Mui, M. Mohtashemi, and A. Halberstadt. Notions of reputation in multiagent systems: a review. In *Proc. of the First ACM-SIGART Int'l joint conference on Autonomous agents and multiagent systems (AAMAS)*, 2002.
16. P. Resnick, R. Zeckhauser, J. Swanson, and K. Lockwood. The value of reputation on ebay: A controlled experiment. *Experimental Economics*, 2006.
17. C. Shapiro. Consumer information, product quality, and seller reputation. *Bell Journal of Economics*, 1981.
18. B. Yu and M. Singh. A social mechanism of reputation management in electronic communities. In *Proc. of the Seventh Int'l Conf. on Cooperative Information Agents*, 2000.
19. B. Yu and M. Singh. Detecting deception in reputation management. In *Proc. of the Second ACM-SIGART Int'l joint Conf. on Autonomous agents and multiagent systems (AAMAS)*, 2003.
20. B. Yu, M. P. Singh, and K. Sycara. Developing trust in large-scale peer-to-peer systems. In *Proc. of First IEEE Symposium on Multi-Agent Security and Survivability*, 2004.
21. G. Zacharia. Trust management through reputation mechanisms. In *Proc. of the Third Int'l Conf. on Autonomous Agents (Workshop on Deception, Fraud and Trust)*, 1999.