

Perseus – A Personalized Reputation System

Petteri Nurmi

Helsinki Institute for Information Technology HIIT

Department of Computer Science, P.O. Box 68, FI-00014 University of Helsinki, Finland

petteri.nurmi@cs.helsinki.fi

Abstract

We propose Perseus, a personalized reputation system. In Perseus, reputations comprise of three aspects: how much I personally trust another individual, how trustworthy others think the individual is, and how much I trust the opinions of others. Perseus is adaptive in the sense that user feedback is used to modify the way the different aspects are considered. We also present simulation experiments, which indicate that Perseus is robust and able to survive under extreme conditions of misbehavior. In addition, Perseus encourages individuals to rate the other party and give fair ratings. We also compare Perseus against other well-known reputation systems.

1. Introduction

The Internet has brought about new opportunities for buying products. As an example, many online portals (e.g., eBay¹ and Amazon²) offer their users the possibility to auction second-hand items. Another example is shopping at online stores. In online transactions, the buyer and the seller often have no interaction history and hence no prior trust can exist between the parties. According to Jarvenpaa et al. [6], trust is a critical factor in any relationship where the truster (buyer) has no control over the actions of the trustee (seller), the decision is important and the environment is uncertain. In online transactions, the prospect of financial loss implies that buying decisions are important³. The uncertainty, on the other hand, results from lack of cues for assessing the trustworthiness of the other party. For example, the buyer cannot engage in face-to-face communications with the seller or examine the quality of the products beforehand.

¹<http://www.ebay.com>

²<http://www.amazon.com>

³Also deciding to whom to sell can be important. For example, negative feedback to the seller causes loss of income in future transactions [5].

Online transactions offer various opportunities for misbehavior. First of all, as the buyer cannot examine the products beforehand, there is an information asymmetry between the buyer and the seller. The seller can take advantage of the asymmetry, e.g., by exaggerating the quality of products. Another possibility for misbehavior results from the procedure that is commonly used for carrying out online transactions. The standard procedure is that the buyer pays for the items before the seller ships them. The problem with this procedure is that after the buyer has paid, the seller can refuse to ship the items. Also the buyer can be dishonest, e.g., by badmouthing the seller [2] or by refusing to send the money.

To foster trust between the customers and to weed out dishonest behavior, many marketplaces employ a so-called *reputation system* [3, 12]. A reputation system can be understood as a computational "word of mouth" system that gathers feedback from customers' past interactions and aggregates this information. The aggregated information is then shown to other customers who can use the information to support their buying and selling decisions.

Most reputation systems for online marketplaces represent the reputation of a user as a single value, which is computed collectively from the available feedback information. A problem in using a single value is that a set of sellers can form a coalition and raise each others reputation by performing bogus transactions where they rate each other positively. The individuals can then misuse their good reputation with customers that are outside the coalition. Another problem is that if individuals know their reputation, they can build up a high reputation and act dishonestly for a while before returning to honest behavior [17].

A possible way to overcome these problems is to use personalized reputations, i.e., to make the reputation values depend on the person who examines them. In this paper we follow this line of thought and propose Perseus – a personalized reputation system. Perseus considers three aspects when computing reputation estimates: how much I personally trust another individual, how trustworthy other (trustworthy) individuals think the person is, and how much

I trust the opinion of others. In Perseus, we compute the personal opinion and the opinion of others using a stochastic approximation algorithm. These two opinions are then combined using a weighted linear sum. Finally, we use another stochastic approximation to adapt the weights of the linear sum over time. We also present simulation experiments, which indicate that Perseus is robust and able to survive under extreme conditions of misbehavior. In addition, the experiments show that Perseus encourages individuals to rate the other party and give fair ratings. We also compare Perseus against other well-known reputation systems. In this paper our focus is exclusively on online auction sites, but Perseus can be easily adapted to other environments.

The rest of the paper is organized as follows: Section 2 discusses related work. In Section 3 we present the general mechanisms that Perseus uses to calculate reputation values. Section 4 presents simulation experiments and Section 5 concludes the paper.

2. Related Work

The notions of trust and reputation play an important role in various fields; see the surveys [10, 15]. A reputation system is a computational mechanism that collects, distributes and aggregates feedback information about the past behavior of individuals [12]. The best known example of a reputation system is employed by eBay where customers can give positive or negative ratings to the other party and the sum of ratings is shown to other users⁴. The eBay system has been studied both theoretically [4] and empirically [5, 13]. Empirical studies have suggested that the reputation system of eBay has been a key ingredient for the success of the service.

The eBay system is known to have several vulnerabilities. For example, individuals can build up a good reputation and then misuse it once they get a high enough offer. Or certain individuals may provide false feedback in order to bias the estimates of the reputation system [2]. To this end, several improved reputation systems for online marketplaces have been suggested; e.g., the Beta reputation system [7], the average system used, e.g., in [8], the Yu-Singh system [19] and the Sporas system [20]. Related mechanisms that have been proposed for other domains include Eigentrust [9] and Confidant [1]. Closest to our approach is the system proposed by Sakai et al. [16]. This mechanism is also based on stochastic approximation, however, it only provides global estimates, and it does not adapt the way different information sources are considered over time.

Most of the reputation systems for online marketplaces provide a global reputation value for individuals. Exceptions of this are the mechanisms that have been designed for

closely connected communities; e.g., Histos [20] and Regret [14]. These mechanisms provide a personalized value that is derived through the social connections of an individual. Our mechanism combines advantages of both personalized and global mechanisms: although our estimates are personalized, they also have a global component that provides information about persons with whom we have not interacted.

3. Perseus: Personalized Reputation System

In this section we describe the Perseus reputation system. In Perseus, reputations are personalized so that the reputation of an individual depends on the person who queries it. The reputation value is a combination of two estimates. One of the estimates is derived from personal experiences whereas the other is derived from the experiences of others. Sections 3.2 and 3.3 discuss how these estimates are computed and Section 3.4 discusses how the estimates are combined. In Section 3.5 we discuss how the estimates are used to determine whether someone is (non-)trustworthy and in Section 3.6 we discuss how to modify the contribution of the different information sources over time. Section 3.7 discusses how we cope with unfair ratings. We start by describing our problem setting.

3.1. Problem Setting

For concreteness, we fix our problem setting to an online marketplace, such as eBay. We assume that the market consists of individuals who can act either as buyers or as sellers. Let i and j be arbitrary individuals who act in different roles, i.e., a buyer and a seller, and let t denote the number of transactions they have carried out with each other.

We assume that the market rules are as follows: after the buyer and seller have agreed on a price, the buyer needs to pay before the seller sends the items. The seller can be dishonest and refuse to send the items or send items of lower quality. After the transaction is concluded, i.e., the participants have received what they wanted or they have given up hope of getting the items, the individuals are asked to rate each other. We require that the ratings are given independently so that neither party sees how the other rates it. Independent ratings are vulnerable to *spammers* who carry out transactions honestly, but rate the other person negatively. Coping with spammers is discussed in Section 3.7.

We use $\rho_k(t)$ to denote the rating given by person k (i.e., i or j) in the t :th transaction. We assume that a rating can be either positive (+1), negative (-1) or neutral (0). When no feedback is given, we consider this as a neutral rating.

In Perseus, we compute a reputation value both for the seller and for the buyer. The reputation value depends on the role of an individual and thus, a person acting both as a

⁴In addition to the sum of ratings, customers are also shown, e.g., comments of the individuals who have interacted with the person recently.

buyer and as a seller has separate reputation values for the different roles. The motivation for having ratings also for the buyers is that we want to support the sellers and provide them information about bad buyers who, for example, give unfair ratings or who bid higher than others, but never transfer the money.

3.2. First Hand Reputation

Personal experiences are undoubtedly the most important source of information for forming opinions about others. When we have good experiences of a person, we expect also our future encounters to be beneficial. Thus, the level of trust we assign to the individual is high – regardless of the opinions of others. On the other hand, when our personal experiences are bad, e.g., when we have been cheated, the level of trust we assign to the individual is low – again, regardless of the opinions of others.

In Perseus, personal experiences are the main source of information for deriving reputation estimates. Reputation estimates derived from personal experiences are called *first hand reputation*. We use $\gamma_i^t(j)$ to denote person i 's first hand reputation of j after t transactions. The values of $\gamma_i^t(j)$ are assumed to lie in the interval $[0, 1]$ so that $\gamma_i^t(j) = 1$ corresponds to complete trust and $\gamma_i^t(j) = 0$ to complete distrust.

Our approach for first hand reputation is motivated by a game theoretic model which assumes that the behavior of each individual j is governed by some parameter θ_j . The parameter θ_j defines the rate with which the person j acts honestly and the goal of the others is to estimate θ_j from observed behavior; see [11] for more information.

In Perseus, the values of $\gamma_i^t(j)$ are computed using a stochastic approximation algorithm. When the feedback is positive, we assume that the other person has acted honestly and update the estimate towards one. Respectively, when the feedback is negative, we assume that the person was dishonest and decrease the estimates towards zero.

Assume first that the behavior patterns of the individuals do not change over time. The parameter θ_j is now fixed and we can estimate it using

$$\gamma_i^t(j) = \gamma_i^{t-1}(j) + \frac{1}{t} (\rho_i(t) - \gamma_i^{t-1}(j)). \quad (1)$$

The term $\alpha = 1/t$ is called the *step-size* of the estimator. This estimator can be seen as an online variant of the maximum likelihood estimator for a binomial distribution as $\gamma_i^t(j)$ converges to the ratio of positive ratings to all ratings. In the experiments we use 0.50 as the initial value, which corresponds to the assumption that we have no knowledge about how likely honesty is in comparison to dishonesty. Also other initial values can be used. For example, if we assume that the ratio of honest transactions to dishonest transactions is 9:1, we could use $\gamma_i^0(t) = 0.90$ as the initial value.

The behavior of a seller can also change over time, and the reputation system should be able to track the changes. A common approach for this setting is to use a constant step-size with the estimator in Eq. 1. When a constant step-size is used, the estimates do not converge to a fixed point, but to a neighborhood near the true value. When the value of θ_j changes, the estimator is able to leave this neighborhood and converge to a neighborhood near the new value. In the experiments we use $\alpha = \max\{0.01, 1/t\}$ as the step-size.

3.3. Third Party Reputation

When individuals have no interaction history, trust and distrust must be based on other sources of information. In large communities, a common way is to consider observations from other members of the community. Sabater and Sierra [15] refer to this information source as *witness information*. A problem with witness information is that it is vulnerable to collusions and false ratings [2, 19]. In small and medium sized communities, the vulnerability is more severe and an alternative is to consider *social information* as has been done, e.g., in [14]. In this paper we assume that the size of the community is large enough and focus only on witness information. We refer to reputation estimates derived from witness reports as *third party reputation*. We use the variable $\phi_i^t(j)$ to denote person i 's view of the third party reputation of person j .

In our system, also third party reputations are personalized so that only the opinions of trustworthy and neutral individuals are taken into account for computing the estimates. This makes our system more resistant to spammers who inject false ratings into the system. We use also other mechanisms to combat unfair ratings; see Section 3.7.

Let $W(j)$ denote the set of individuals (witnesses) who have given at least one positive or negative rating to j . We define $T \subseteq W(j) \setminus \{i\}$ as the subset of witnesses that person i finds trustworthy or neutral. For an individual i , we compute the third party reputation of j using the average sum of ratings, i.e.,

$$\phi_i^t(j) = \frac{1}{\#T} \sum_{k \in T} \gamma_k^t(j). \quad (2)$$

Here $\gamma_k^t(j)$ is used (with a slight abuse of notation) to denote the first hand reputation k assigns to j when i and j have carried out t transactions. Alternatively we can use the first hand reputation estimates as weights for the terms in the sum. This would correspond to the weighted majority algorithm used by Yu et al. [19].

3.4. Reputation Aggregation

In order to provide a single summary to the user, we need a mechanism that aggregates first hand and third party rep-

utations. We combine these two information sources using a weighted linear sum, which is also the most common approach found in the literature, e.g., [11, 19]. We use π_1 to denote the weight for first hand reputation and π_2 to denote the weight for third party reputation. The trustworthiness of person j from the point of view of person i is then given by

$$R_i^t(j) = \frac{\pi_1}{\pi_1 + \pi_2} \gamma_i^t(j) + \frac{\pi_2}{\pi_1 + \pi_2} \phi_i^t(j). \quad (3)$$

Although the normalized weights sum up to one, we maintain two separate weight terms. This can be understood so that we maintain estimates of the level of trust an individual assigns to the individual information sources, and we then use the ratio of the trust levels to aggregate reputation values. This makes it also easier to update the coefficients π_1 and π_2 over time; see Section 3.6. In the experiments we use $\pi_1 = 0.25$ and $\pi_2 = 0.75$ as the initial values for the weights. Note that although we combine the values into a single value, in systems with human users we do not have to aggregate the information, but we can show separately the reputation from different information sources.

3.5. Determining Trustworthiness

The third party reputation estimates rely on a method for determining whether another individual is considered trustworthy. In addition, this kind of mechanism is essential for simulation purposes. In our case, the main role of this mechanism is to make it possible to prune out opinions of irrelevant agents and thus to speed up computations.

Currently we use a threshold based approach for determining whether an agent is trustworthy or not. The reputation estimates $R_i^t(j)$ lie within the interval $[0, 1]$ at all times and the closer we are to (zero) one the more (non-)trustworthy the other person is considered. In the simulations we used 0.75 as the threshold for trustworthiness and 0.25 as the threshold for non-trustworthiness. Values between 0.25 and 0.75 were considered neutral. In practice, these values should be set using background information about the rate of dishonesty in the marketplace.

3.6. Updating Coefficients

As discussed in Section 3.4, Perseus combines different information sources into a single reputation score using a weighted linear sum. The weight terms, which correspond to variables π_1 and π_2 in Eq. 3, can be understood as a measure of the level of trust an individual assigns to the different information sources. Intuitively, the level of trust should vary over time and it should take into account, e.g., our increased knowledge and the quality of the feedback. For example, after we have interacted several times with a seller, our personal experience is enough to judge the honesty of the seller. As another example, when majority of

the feedback information is faulty, we should trust our own experiences more than the opinions of others.

In Perseus, we use a stochastic approximation algorithm to update the weights over time. When the user provides a positive or a negative rating, we first check whether the rating *disagrees* with one the reputation estimates, i.e., with the first hand reputation or with the third party reputation. We say that a rating and an estimate *disagree* when one of them is positive and the other one is negative. When both the rating and the estimate are positive (or negative), we say that they *agree*.

In Perseus, we update a coefficient towards one whenever the reputation source agrees with the rating provided by the user. Respectively, when the rating and the reputation source disagree, we update the coefficient towards zero. Thus, for each coefficient π_i ($i = 1, 2$), we perform an update using

$$\pi_i^{t+1} = \pi_i^t + \begin{cases} \beta (1.0 - \pi_i^t) & \text{if agree} \\ \beta (0.0 - \pi_i^t) & \text{if disagree} \end{cases} \quad (4)$$

In the experiments, we use $\beta = 0.001$ as the step size. This ensures that the changes in the coefficients are relatively small. Thus, trust builds up relatively slowly and the reputation scores are relatively stable. If we would use a bigger step size, the reputation scores can fluctuate rapidly, which can be confusing to users.

3.7. Coping With Unfair Ratings

The performance of a reputation system ultimately depends on the quality and correctness of the feedback information that the users provide. Occasionally, the sellers and the buyers may intentionally provide false or otherwise unfair feedback in order to bias the reputation estimates [2]. Dellarocas has identified two scenarios where this problem occurs. The first scenario is *ballot stuffing* where a group of individuals colludes and performs bogus transactions with each other in order to raise the reputation of the coalition's members. The second scenario concerns the case where buyers (or sellers) provide unfair negative feedback to the other party. The most common way to deal with this problem is to perform statistical analysis of the ratings and to exclude ratings that are considered outliers. For example, Whitby et al. [18] use quantiles to detect whether a particular rating is an outlier.

Perseus uses a rather different approach to combat unfair ratings. Similarly as with the updates to reputation coefficients (Section 3.6), our first step is to determine whether the two ratings disagree. If the ratings disagree, we always give a negative rating to the seller. When the negative rating was given by the seller, we also give a negative rating to the buyer. The motivation for our scheme is that it makes transactions between the buyer and the seller less likely in

the future. If the buyer gives a negative rating, the system interprets this as a sign that the person does not want to transact with the seller anymore. This does not prevent the seller from conducting transactions as from a new buyer's perspective the overall reputation remains neutral until at least one transaction is conducted between the new buyer and the seller. On the other hand, if the seller gives a negative rating, Perseus considers this feedback so that the buyer should be avoided. Our scheme also encourages the seller to rate the buyer as otherwise it can be falsely accused of misbehavior.

If the ratings agree or (at least) one of the ratings is neutral, we update only the reputation of the seller. If we were to update also the reputation of the buyer, a group of individuals could perform bogus transactions to increase the reputation of a buyer. The buyer could then get cheaper prices and a competitive advantage on the market.

4. Experiments

In order to evaluate Perseus, we have conducted a set of simulation experiments. Our simulation setup has been slightly adapted from Schlosser et al. [17], and we describe it in Section 4.1. In the simulations we model individuals as agents that have specific behavioral patterns. The behavioral patterns that we use are described in Section 4.2. The metrics that we use to assess the performance of Perseus are described in Section 4.3. The results of the simulation experiments are presented in Section 4.4.

4.1. Simulation Setup

Our simulation setup models an online marketplace. The individuals are modeled as agents with specific behavior patterns. Each agent can act both as a buyer and as a seller. The items that the sellers provide are assumed to be homogeneous so that initially buyers have no reason to prefer one seller over the other. We assume that all items are equally priced; thus we do not consider the possibility that the value of transactions evolves as a function of reputation.

During single simulation iteration, we consider each agent in turn as a potential buyer and randomly match it with another agent who acts as a potential seller. The agents decide whether they want to conduct a transaction with the other agent. The decisions of the agents depend on their behavior models; see the next section. If the agents approve the transaction, the seller determines the outcome of the transaction. We consider only binary decisions: the seller either ships the items (true) or does not send anything (false). Thus we do not consider the possibility of having items of different quality. The decision of the seller is communicated to the buyer, after which both agents are asked to rate the other agent. The ratings are given independently

and they are not revealed to the other agent. The ratings are used to update the reputation estimates and the mechanism described in Section 3.7 is used to handle unfair ratings.

4.2. Behavior Models

In the experiments we consider five types of behavior models: *Honest*, *Evil*, *Selfish*, *Disturbing* and *Spamming*. With the exception of spamming, these models are adapted from Schlosser et al. [17]. In the following we briefly describe each of these models.

Honest Honest agents accept transactions with neutral and trustworthy agents. They always carry out transactions honestly. Honest agents always rate the other party and the ratings they give are fair.

Malicious When acting as a buyer, a malicious agent accepts transactions with neutral and trustworthy agents. When acting as a seller, a malicious agent accepts all transactions and acts honestly or dishonestly by chance. Malicious agents always rate the other negatively.

Selfish Selfish agents are passive sellers who never buy anything, but accept to sell items to neutral and trustworthy agents. Selfish agents never rate their opponents.

Disturbing Disturbing agents attempt to build up a good reputation by acting honestly and misbehave once they have obtained a high reputation. Since our system does not reveal reputation values to other agents, disturbing agents must try to estimate their own reputation.

Spamming Spamming agents act otherwise similarly as the honest agents, except that as buyers they always rate negatively. By giving negative ratings to sellers, they attempt to make themselves more attractive to buyers.

4.3. Evaluation Metrics

As the evaluation metrics we consider the *transaction rate* and the *number of transactions* conducted. The metrics are evaluated separately for each combination of agent type and role (buyer and seller). The transaction rate measures the portion of transactions that have been completed successfully. This measure alone can be a bit misleading, if the agents conduct very few transactions. To this end, we also consider the number of transactions that the agents conduct. Schlosser et al. [17] also consider metrics that are based on the reputation values of the agents. In our case these metrics are not applicable as the reputation scores of the agents are not unique.

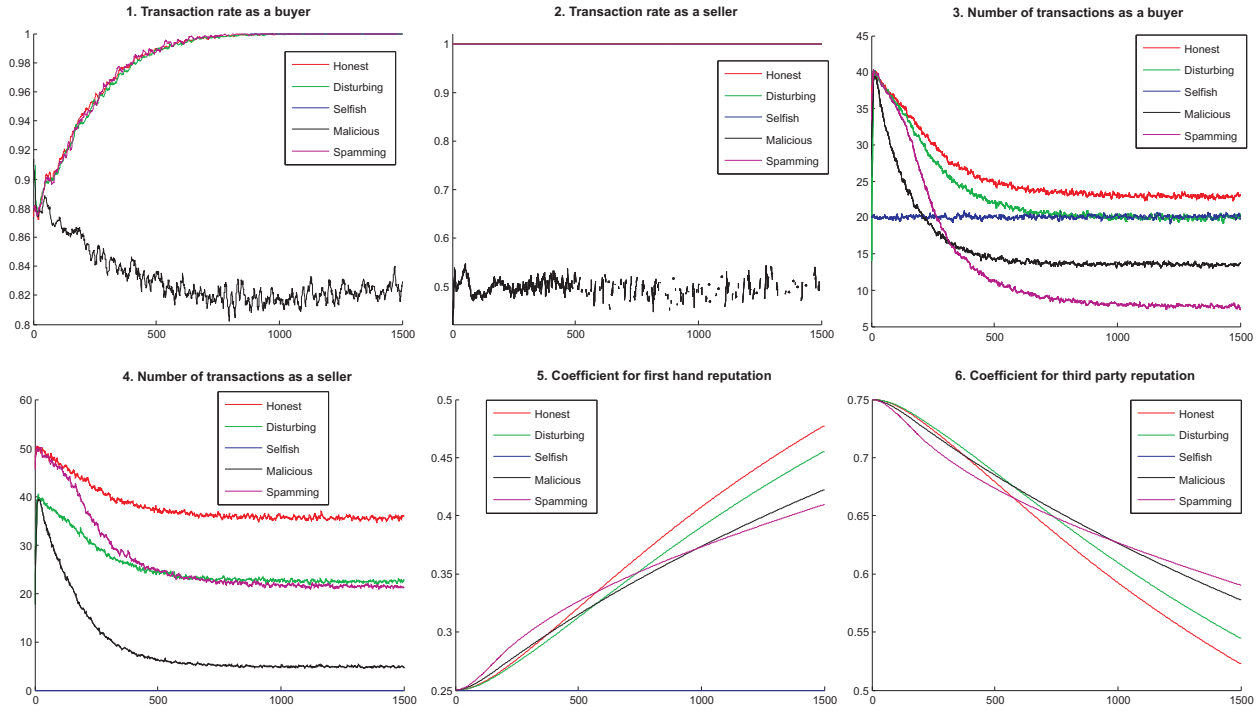


Figure 1. Results of the simulation experiment with 250 agents. In all the figures, x-axis corresponds to iterations and y-axis to the value of the metric.

4.4. Results

In the first experiment setup, we kept the number of agents and the distribution of behavior models fixed. The goal of these experiments was to give insights into the performance of Perseus. In this setup, we had 50 agents per behavior model. Thus the total amount of agents in the simulation was 250. We ran 1500 iterations and repeated the same setup for 50 times. The results of these experiments are shown in Figure 1.

As the results indicate, the rate of successful transactions (as buyers) is relatively high over time and converges rapidly to 100%. Perseus is also able to correctly identify malicious agents and it is able to cope with spammers as the amount of transactions the spammers conclude decreases rapidly. Respectively, the honest agents learn to rely on their own experiences the most whereas the spammers are the least confident in their own experiences.

In the second experiment setup, we used a population of 200 agents, of which the majority was somehow misbehaving. More precisely, we set 60% of the agents to be of a specific misbehaving type, and the rest of the behavior models were uniformly distributed among the remaining 40% of the agents. The goal of these simulations was to measure the performance of Perseus under extreme conditions. We also repeated the same experiments using some well-

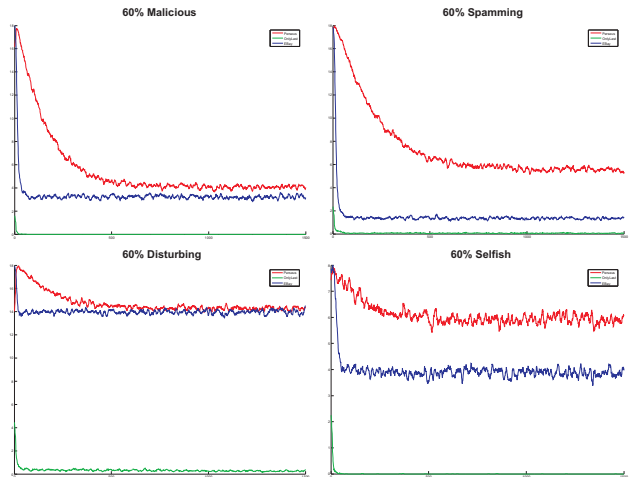


Figure 2. Comparison against other reputation systems under extreme conditions. The x-axis corresponds to iterations and the y-axis to the total number of transactions that honest agents were able to conduct.

known reputation systems. We evaluated our system against the eBay system and the OnlyLast system of Dellarocas [4] since these were shown to perform well in the simulations of Schlosser et al. [17]. The results of the comparison are shown in Fig. 2.

For each mechanism, we used the same approach for coping with unfair ratings (see Section 3.7). As the results indicate, Perseus is the best mechanism in the simulations and it is especially robust against spamming and selfishness. When the number of malicious or disturbing nodes is high, the performance of Perseus is only slightly better than the performance of other mechanisms.

As an additional remark on the simulations, we note that eBay has been shown to be vulnerable to disturbing agents who use their *global* reputation for misbehavior [17]. With Perseus, the reputation of this kind of agents decreases rapidly and, furthermore, the agents cannot obtain information about their global reputation at any point.

5. Conclusions

We have proposed Perseus, a personalized reputation system. We evaluated Perseus using simulation experiments, which show that Perseus is robust against various kinds of misbehaviors and performs well even under extreme conditions. Furthermore, since individuals do not have a global reputation value, they cannot misuse their reputation.

6. Acknowledgments

This work was supported in part by the IST Programme of the European Community, under the PASCAL network of excellence, IST-2002-506778. The publication only reflects the authors' views.

References

- [1] S. Buchegger and J.-Y. L. Boudec. Performance analysis of the CONFIDANT protocol: Cooperation of nodes fairness in dynamic ad-hoc networks. In *Proceedings of the 3rd ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, pages 226–236. ACM Press, 2002.
- [2] C. Dellarocas. Immunizing online reputation reporting systems against unfair ratings and discriminatory behavior. In *Proceedings of the 2nd ACM conference on Electronic Commerce (EC)*, pages 150–157. ACM Press, 2000.
- [3] C. Dellarocas. The digitization of word of mouth: Promise and challenges of online feedback mechanisms. *Management Science*, 49(10):1407–1424, October 2003.
- [4] C. Dellarocas. Efficiency and robustness of binary feedback: Mechanisms in trading environments with moral hazard. MIT Center for eBusiness Working Paper #170, 2003.
- [5] D. Houser and J. Wooders. Reputation in auctions: Theory, and evidence from eBay. *Journal of Economics & Management Strategy*, 15(2):353–369, 2006.
- [6] S. L. Jarvenpaa, N. Tractinsky, and M. Vitale. Consumer trust in an Internet store. *Information Technology and Management*, 1(1-2):45–71, 2000.
- [7] A. Jøsang and R. Ismail. The Beta reputation system. In *Proceedings of the 15th Bled Electronic Commerce Conference*, 2002.
- [8] R. Jurca and B. Faltings. Towards incentive-compatible reputation management. In *Trust, Reputation and Security: Theories and Practice*. Springer, 2003.
- [9] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina. The Eigentrust algorithm for reputation management in P2P networks. In *Proceedings of the 12th international conference on World Wide Web (WWW)*, pages 640–651. ACM Press, 2003.
- [10] L. Mui, A. Halberstadt, and M. Mohtashemi. Notions of reputation in multi-agent systems: A review. In *Proceedings of the 1st International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 280–287. ACM Press, 2002.
- [11] P. Nurmi. A Bayesian framework for online reputation systems. In *Proceedings of the Advanced International Conference on Telecommunications and International Conference on Internet and Web Applications and Services (AICT/ICIW)*. IEEE Computer Society, 2006.
- [12] P. Resnick, K. Kuwabara, R. Zeckhauser, and E. Friedman. Reputation systems. *Communications of the ACM*, 43(12):45–48, 2000.
- [13] P. Resnick, R. Zeckhauser, J. Swanson, and K. Lockwood. The value of reputation on eBay: A controlled experiment. *Experimental Economics*, 9(2):79–101, June 2006.
- [14] J. Sabater and C. Sierra. Social regret, a reputation model based on social relations. *ACM SIGecom Exchanges*, 3(1):44–56, 2001.
- [15] J. Sabater and C. Sierra. Review on computational trust and reputation models. *Artificial Intelligence Review*, 24(1):33–60, 2005.
- [16] T. Sakai, K. Terada, and T. Araragi. Robust online reputation mechanism by stochastic approximation. In *Adaptive Agents and Multi-Agent Systems II: Adaptation and Multi-Agent Learning*, volume 3394 of *Lecture Notes in Computer Science*. Springer, 2005.
- [17] A. Schlosser, M. Voss, and L. Brückner. On the simulation of global reputation systems. *Journal of Artificial Societies and Social Simulation*, 9(1):1, Jan 2006.
- [18] A. Whitby, A. Jøsang, and J. Indulska. Filtering out unfair ratings in Bayesian reputation systems. In *Proceedings of the 7th International Workshop on Trust in Agent Societies*, 2004.
- [19] B. Yu and M. P. Singh. Detecting deception in reputation management. In *Proceedings of the Second International Joint Conference on Autonomous Agents & Multiagent Systems*, pages 73–80. ACM Press, 2003.
- [20] G. Zacharia and P. Maes. Trust management through reputation mechanisms. *Applied Artificial Intelligence*, 14(9):881–907, 2000.