Pinocchio: Incentives for honest participation in distributed trust management

Alberto Fernandes, Evangelos Kotsovinos, Sven Östring and Boris Dragovic

University of Cambridge Computer Laboratory
15 JJ Thomson Avenue
Cambridge CB3 0FD, UK
{firstname.lastname}@cl.cam.ac.uk

Abstract. In this paper, we introduce a framework for providing incentives for honest participation in global-scale distributed trust management infrastructures. Our system can improve the quality of information supplied by these systems by reducing free-riding and encouraging honesty. Our approach is twofold: (1) we provide rewards for participants that advertise their experiences to others, and (2) impose the credible threat of halting the rewards, for a substantial amount of time, for participants who consistently provide suspicious feedback. For this purpose we develop an honesty metric which can indicate the accuracy of feedback.

1 Introduction

Peer-to-peer systems, on-line auction sites and public computing platforms often employ trust management systems to allow users to share their experiences about the performance of other users in such settings [1, 2]. However, the success of these trust management systems depends heavily on the willingness of users to provide feedback. These systems have no mechanisms to encourage users to participate by submitting honest information. Providing rewards is effective way to improve feedback, according to the widely recognised principle in economics which states that people respond to incentives.

Some of the most popular trust management systems in use currently operate without the promise of rewards for providing feedback, such as the eBay auction site or the used goods trading facility provided by the Amazon marketplace. Our view is that under these conditions the users who participate in the trust management scheme by submitting information about their interactions with others are, in fact, pursuing "hidden" rewards, often with unwanted effects. For instance, in the eBay case, there is strong empirical evidence to suggest that buyers and sellers advertise positive feedback regarding each other, seeking to increase in their reputation via mutual compliments [17]. In this case, the reward implicitly offered by the system is the possibility of getting a positive review about oneself.

Also, people who have had particularly bad experiences will be normally more inclined to advertise their experiences as a form of revenge against the

user that did not provide the desired service. Such hidden rewards bias the feedback system; users who have had average experiences with other users and are not aiming at increasing their reputation or seeking revenge against a bad service provider will have little reason to provide feedback. An explicit reward system has the advantage of attracting those users across the board.

Moreover, in other settings with different parameters, such as public computing environments, the inherent incentives for participation are very limited – as discussed later in the paper. In such cases, a component that will provide explicit incentives for participants to submit feedback about their experiences with others is crucial. However, incentives should not be provided for users that are likely to be dishonest or submit information that has little relevance to reality.

In this paper we introduce Pinocchio; a system which rewards participants that provide feedback that is likely to be accurate, while having mechanisms for protecting itself against dishonest participants. In Section 2, we define the environment in which Pinocchio is designed to operate. In Section 3, we describe how it is possible to spot cheats and use this knowledge to influence participation, and Section 5 summarises our conclusions.

2 Example settings

To understand the operation of Pinocchio, it is important to set the scene in which our system is designed to operate. We will state the general parameters of the environment in which Pinocchio can fit, and then outline a few realistic examples of such environments in the area of trust management architectures operating with global public computing systems. The list of example settings is by no means exhaustive; there are several other similar environments in which our system could function.

2.1 Environmental parameters

There is a group of *participants* that provide services to each other. Whether these participants are organised as peers or as clients and servers makes little difference. The participants are tied to semi-permanent identities – their identities can change but it is a costly operation and cannot happen very often. Obtaining an identity is a result of a *registration* process they had to go through in order to join the group. Participants are *authenticated*. We cannot make assumptions about the duration of each interaction between participants, but we expect participants to have a *long-term presence* in the system, even if they do not use the services provided by other participants or provide services themselves.

Participants are owned and administered by a number of independent organisations, and therefore are *autonomous*, in the sense that there is no central control or strict coordination on the services that these will provide. It can be assumed that some authority has the ultimate right to eject a participant from the platform in cases of serious offences, but the standard of service that each

participant will deliver in each interaction is left to its discretion and cooperativeness. Also, each participant can valuate the services that other participants provide independently and subjectively, without any control on the correctness of its opinion. We term such systems *federated*. We outline a few typical examples of such systems in Section 2.2.

A number of analogies of federated systems can be drawn from the human society; restaurants are administered by different people, provide very diverse qualities of service, and there is little central control on the quality of the food that they provide, apart from making sure that they comply with the basic regulations of food hygiene. There is no control on how tasty the food will be, or on the size of portions. Accordingly, there is no control on the opinions that customers can voice. Each customer is allowed to express any opinion about any restaurant, even if she has never visited it.

A *trust management system*, as described in Section 2.3, is in place to allow participants to share their experiences about interactions with others – that is, to support facility similar to gossipping in the human society. Pinocchio intends to use opinions submitted by participants to the trust management system in order to automatically reward users who report information that is likely to be accurate.

2.2 Global public computing systems

PlanetLab [16] is a global overlay network targeted to support the deployment and evaluation of large-scale distributed applications and services. Resource reservations – such as CPU time or memory space – are made through *resource brokers* that provide the tickets that users can submit to the servers to obtain resources. However, PlanetLab nodes are owned by several different organisations and administered by an even larger number of people. Whether a ticket will be honoured is in each node's discretion. While most nodes will behave as expected, some nodes may not honour slice reservations, and others may fail frequently. It is not hard to see that all nodes may not provide the same level of service. A similar setting is that of Grid computing systems [10].

The XenoServer Open Platform is building a global public infrastructure for distributed computing developing [12]. Clients can deploy untrusted tasks on servers that participate in the platform, and ultimately get charged for the resources their tasks consume. Servers are again owned and administered by a diverse set of organisational entities. The fact that users pay for the services promised by the servers – clients and servers agree on the resources to be provided by the server and the payment to be made by the user beforehand – makes the need for encouraging accurate feedback even more compelling. Some servers may overcharge clients or not deliver the expected service, and on the other side some clients may refuse to pay or abuse the resources given to them.

2.3 Distributed trust management

The overall experience of using the system can be improved if each participant shares her experiences about aspects of the level of services provided by the participants she interacts with. This is done by making quantitative statements about the level of services received. For instance, participant A rates B as 70% regarding property M.

Participants can share their experiences from interactions with other users by subscribing to a *trust management infrastructure* that is in place. Participants can make their opinions public by *advertising* them to the trust management infrastructure in the form of *statements*, and obtain information about others' opinions by *querying* the system. It is assumed that all supported queries have fairly similar complexity. The trust management system can be imagined as a pool, exporting unified interfaces for storing and retrieving statements.

A real-world system that follows the above properties is XenoTrust [9, 8], the system we are developing to allow reputation dissemination in the XenoServer Open Platform. XenoTrust will act as a pool of statements, and export interfaces for submitting statements and querying the system to retrieve and combine them.

We assume that the trust management infrastructure will be able to charge for its services, in some sort of currency. One straightforward example where this would be possible is the XenoServer Open Platform, which encompasses charging and pricing mechanisms. Also, Grid computing projects have recently launched research on providing such functionality [11].

One of the problems that we seek to address is the common *free-riding* problem experienced in most open infrastructures [3], where in this case free-riding refers to the behaviour of participants who submit queries to the trust management system but who do not contribute to the system's knowledge base. The usefulness and reliability of the trust management scheme itself depends heavily on the amount of reputation feedback it receives from its participants. If few participants choose to advertise reputation statements, information in it will be significantly less accurate. Thus a policy that rewards active participation benefits the system.

However, rewarding participation will also provide an incentive for providing inaccurate information. Giving an honest account of a participant's experience takes more time than just feeding random reputation statements back to the system. If both approaches result to the same reward, our incentive for active participation becomes an incentive for inaccurate feedback.

To anticipate the above issues, we propose Pinocchio, a consultant component that can be attached to trust management infrastructures, designated to provide advice on who to reward, as shown in Figure 1, by applying an honesty metric to spot dishonest advertisements.

3 The Pinocchio Framework

Our approach for improving the quality of information in the trust management system is twofold; we encourage users to submit statements, reporting their

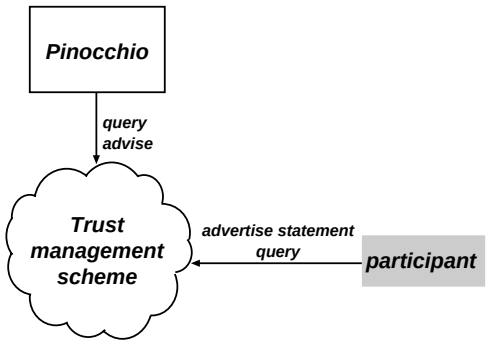

Fig. 1. Pinocchio in the envisaged trust management context

experiences about their interactions with other users, by providing a *reward* for each submitted statement. At the same time, to protect the reward system from users who may submit inaccurate or random statements to obtain rewards we use a probabilistic *honesty metric* to support spotting dishonest users and deprive them of their rewards.

This metric allows weeding out dishonest providers of information, but its main purpose is to prevent it, by the simple advertisement of its existence. Assuming that agents act on self-interest, they will not cheat if perception of risk of exposure and punishment for misbehaviour increases the cost of cheating sufficiently so that it outweighs its benefit.

Section 3.1 establishes a pricing and reward model and Section 3.2 shows how cheats can be detected.

3.1 Reward model

Participants that have subscribed to the trust management scheme can advertise their experiences – in the form of statements – and perform queries that combine, weigh and retrieve statements, in order to obtain information about others' experiences. Each query will incur a fixed cost to the participant, as we expect that the complexity of evaluating individual queries will not vary significantly. To create incentives for participants to provide information regarding the performance of others, the trust management system will provide a reward for each statement submitted, provided that the user submitting it is deemed to be honest.

The trust management system will set up a credit balance for each participant, which will be credited with a reward for each statement advertised and debited for each query made by that user. The trust management system can set a maximum limit to the amount of credit given as rewards to a participant per minute.

If a participant's credit balance is positive, she can use it to get a discount on queries she will make in the future. There is no way to cash the credit for money.

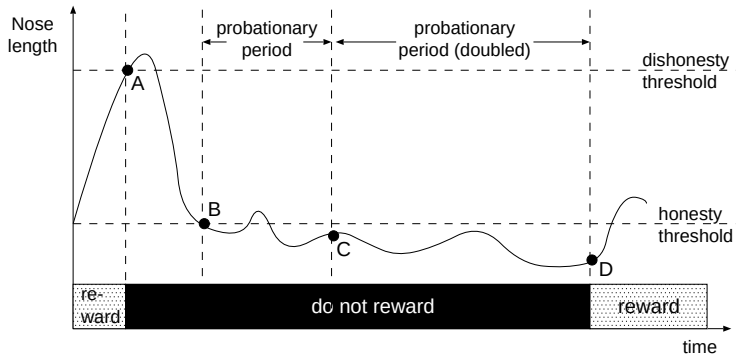

Fig. 2. Honesty and dishonesty thresholds, and probationary period

While the credit provides a tangible incentive to users to participate and submit information to the system, the system does not specifically reimburse the users with monetary repayments. We believe that this feature makes attacks against our system less attractive, as discussed in 3.3.

When Pinocchio determines, using the honesty metric described in the next section, that a participant has been dishonest, the behaviour of the system changes. If the honesty metric rises above a *dishonesty threshold*, then the trust management system will be advised not to reward statements advertised by this participant any more. If her behaviour reverses, with subsequent information being regarded as honest, then once her nose length metric falls below an *honesty threshold* and stays below that threshold until the *probationary period* is completed, the system will resume accumulating credits for the client.

We consider it necessary to have hysteresis in setting the dishonesty and honesty thresholds, as well as the adjustable probationary period, to ensure that participants cannot oscillate, with small amplitude, around a single threshold for their own gain. The probationary period can be doubled each time a participant is estimated to be dishonest, to be long enough to discourage participants from being dishonest several times, but not be too harsh and disappoint first-time cheaters.

An example is shown in Figure 2, where a participant, initially rewarded for every statement she provides, is deemed dishonest at point A. At that point the system stops providing rewards. Once the participant's nose length falls below the honesty threshold – at point B –, she enters a first probationary period, during which she has to remain honest in order to start receiving rewards again. However, her nose length rises above the honesty threshold during that period. Thus, after the end of the first probationary period she enters a second one of double length, starting at point C. The participant is only considered honest again at point D, after demonstrating honest behaviour for as long as the second probationary period required.

3.2 Honesty metric

The metric is based on an intuitive process used by human beings on an everyday basis. To illustrate it, let's introduce Joe. He has not tried out every single make of automobile in the market, but he interacts with his friends and colleagues and hears their opinions about the different brands. He builds in his head a first-level probabilistic model that tells him how likely it is that someone will be pleased by cars made by different brands. For instance, suppose most of the people he interacts with like cars made by ABC and dislike cars made by DEF. If his friend, Adam, buys an ABC and tells Joe he is disappointed, this surprises him, as in his probabilistic model the chance of an ABC being considered low-quality is low.

Joe makes similar intuitive estimates of probabilities for many different car brands. On the basis of these, he also constructs a second-level probabilistic model, built on top of the first, to judge the people he normally interacts with. If Adam always gives Joe opinions that seem bizarre, such as valuing DEF as great and ABC as poor, Joe may stop taking Adam's opinions into account. On the other extreme, there is Miss Sheep, whose opinions always agree with the average opinion about everything. Again, Miss Sheep may lose Joe's respect, because he thinks she does not offer him any new or useful information. Joe finds Mr Goody, who often follows the general opinion but sometimes contradicts it, a useful source of advice.

This is an instinctive self-defence mechanism present in the way humans operate, but not in existing trust management systems. Our approach follows the intuitive process that Joe uses. We build a first-level model that maps opinions to probabilities. In that model, "ABC is poor quality" would be mapped to low probability. The second-level model will look at the history of a participant to estimate how good he is at assessing car manufacturers in general, and whether he may be dishonest – like Adam – or always following the stream – like Ms Sheep. The translation of the very general observations of Joe's behaviour into mathematical models are detailed in the following section.

Our view is that augmenting trust management systems with a component that will be able to suggest which users are worth rewarding is necessary, although not sufficient, to improve the integrity of a trust management system. The main goal of our metric is to protect the reward system against a very specific threat, which is users that take the easiest route to the reward – sending random opinions instead of genuine ones.

Naturally, this threat may occur simultaneously with others; Pinocchio does not intend to protect against conspiracies among participants or bad mouthing. These could be addressed at the trust management system level or by other external consultant components, and there already exist tools that can deal with them, such as [7]. Such conspiracies are not expected to be affected by the existence of a small reward for accurate information providing.

Mathematical Model In this section, we propose a probabilistic model that balances the need to get an accurate assessment of the honesty of information

providers against limited computational resources. We devise an *estimator* of the probability of each participant being dishonest.

Our model fundamentally treats the *perceptions* that participants have about a certain subject as discrete random variables. A single interaction may give rise to many different subjects for opinions – for instance, beauty, safety and reliability of ABC cars or expediency of service and quality of product provided by a server.

All of these subjects are collected in a set of random variables R. When a user interacts with a participant X, she *observes* one sample from all random variables associated with X – i.e. all of X's properties. The user then reports the observed values for each of those random variables, by assigning scores to each property of X.

After collecting a sizable number of observations of each element of R, we fit a probability distribution to each of them. As in Bayesian theory, if we have little information about a variable – because few opinions have been collected about a certain subject –, the distribution will be closer to uniform and will have less weight in our final metric. The collection of the *assumed* probability distributions for all of our random variables forms a database that will be used to check on each user's credibility.

We introduce a new set S of random variables, whose elements are

$$S_{s,p} = ln(P(R_{s,p}))$$

where $P(\bullet)$ stands for *estimated probability*. This is the probability that a score about property p of user s is accurate.

For example, suppose user Bob assigns a score of 0.9 to the performance of user X. Pinocchio will consult the estimate of the probability distribution for the performance of user X, and get an estimate of the probability for a score of 0.9, say 10% probability. So ln(0.10) would be one instantiation, associated with Bob, of $S_{x,performance}$.

At this point we have two values associated with Bob and the "performance of x" subject. The first one is the grade given by Bob, 0.9. The other one is the log-probability – $ln(0.10)$ – with which a score of 0.9 would be reported for X's performance. We are interested in the second value. For every opinion expressed by Bob, we'll have such a log-probability.

The data associated with Bob is limited to a small subset of S, as he quite likely did not provide information on every single participant in the system. So we define a subset of S, $B \subseteq S$, of all elements of S instantiated by Bob. We can further cut this set down by excluding elements of B where data is very sparse, such as where few users have expressed opinions about a particular participant.

Let us assume for the moment that all the variables in B are independent. We can then sum all of them to get a new random variable:

$$T_{Bob} = \sum_{s,p \in B} ln(P(R_{s,p})) \tag{1}$$

This is the log of the probability that our model assigns to a user submitting a particular set of statements about the participants and properties in B. A natural intuition would be to say that the higher the probability our R-distributions assign to Bob's statements, the stronger the evidence for these being true observations from our random variables. We would then choose T_{Bob} as our estimator.

This is not the best estimator, though. In an intuitive way, a typical honest user, when voicing his opinion about several properties of several participants, will in many cases be close to the average opinion in the community, and sometimes far from it. So this naive method would heavily punish honest users that frequently happen to disagree with the community.

Because T_{Bob} is defined as a sum of random variables, we know from the Central Limit Theorem [1] that if the set B is large enough, it will have a distribution close to Gaussian. So we can proceed to estimate its mean and variance via, for instance, Monte Carlo sampling. Our estimator for the honesty of the user, Bob's $Nose length$ statements would be then how much the observed instance of T_{Bob} deviates from its mean, in terms of standard deviations: $Nose length = |Z|$, where

$$Z = (t_{Bob} - \hat{\mu})/\hat{\sigma}, \tag{2}$$

and $\hat{\mu}, \hat{\sigma}$, are our estimates for mean and standard deviations of T_{Bob}; and t_{bob} is the observed sample.

An attentive reader could accuse us of an apparent contradiction. How can our most likely sequences, the ones with a high T_{Bob} score, be somehow considered less probable by $Nose\ length$? For exposition, let us imagine that in a foreign country, on every single day there is a 10% probability of raining. Every day Bob observes if it rained or not and take notes over a year. The single most likely sequence of events is no rain at all. But the expected number of days of rain is $365 \times 0.1 = 36.5$, and a report of zeros days of rain in the whole year would be very suspicious. In our analogy, a "rainy day" would correspond to some statement that is given a low R-probability and we would prefer to see Bob reporting roughly "36.5 rainy days" rather than zero.

Simulation To illustrate this idea, we created 20 discrete random variables with random probability distributions to simulate the behaviour of the variables in set B. We simulated 50 thousand different users, all of them giving a set of 20 opinions, according to our underlying probability distributions. Using our previous knowledge of the distributions, we computed $Nose\ length$. Figure 3 shows that our simulated $Nose length$ behaves like a Gaussian random variable.

In the figure, we show the nose lengths corresponding to sets of statements made by honest users, produced from the true R distributions. The nose lengths of these users cluster together very close to the average (zero), and all of them within a small number of standard deviations from the mean, varying between minus seven and plus three.

[1] Although the random variables are not identically distributed, the CLT still applies as they are bounded (see for instance [5])

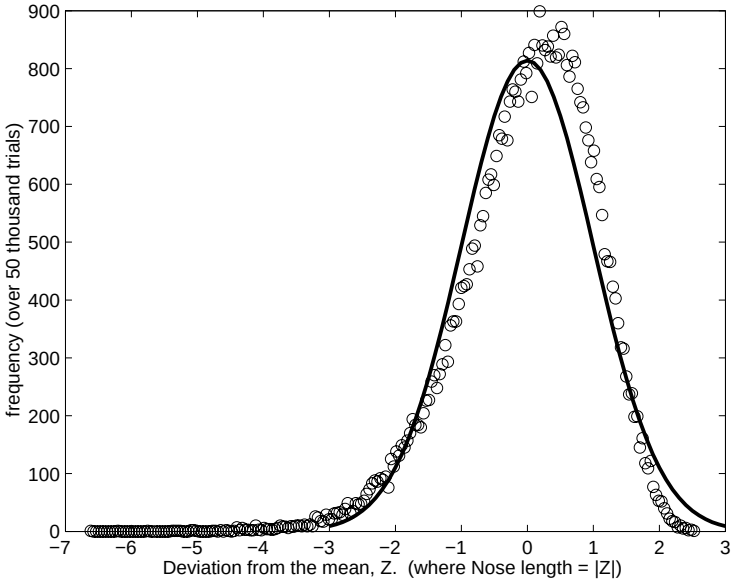

Fig. 3. Simulation of the behaviour of $Noselength$. The circles are a histogram obtained by Monte Carlo sampling. The continuous line is a Gaussian fitted to it. Points to the right of the x axis are those with high probability according to the R variables

Points to the right of the cluster of circles shown in the figure – deviation from the mean more than plus three – would correspond to sets of statements t_{bob} where every single statement is very close to the mean of the other users opinion – behaviour similar to Miss Sheep's. After a certain point, our estimator judges them "too good to be true".

The nose lengths of users whose sets of statements were generated without regard to the true distributions would be larger than seven, and fall way to the left of the circles, This would correspond to "lazy" users that try to obtain the reward by submitting random numbers instead of their true opinions. We simulated these users by assuming a uniform distribution of answers – that they users would be as likely to attribute a "1" as a "10" to any property. In 50 thousand simulations, every time the "dishonest" answers got $Nose\ length$ values between 50 and 300, totally disjoint from the "honest" set. But these are overly optimistic results due to the fact that our R distributions are known and include some regions of very low probability.

Discussion The accepted sets of statements cluster together in a small area of the range of T_{Bob}; completely random responses would be unlikely to fall in this area, and to successfully emulate an acceptable sequence. If we knew the true distributions and the malicious user did not, this probability in most cases – if

the true distributions are distant from a uniform distribution – would be very small.

As we do not have access to the true probability distributions, we expect to use a Maximum Likelihood estimator of these distributions. Any such estimator will have wide error bars if data is sparse, so we propose only including in set B distributions with a sizable amount of data. Conversely, we cannot judge Bob before he provides a reasonable amount of information on several participants/criteria. And because our data is highly subjective, we propose using the estimator described above to cut off information from users only after a relatively high threshold, so that people with unusual opinions aren't punished.

An alternative to this estimator would be to estimate the probability distribution of T_{Bob} directly from Monte Carlo methods or by using convolution over the individual R distributions. The former would have to involve careful line fitting in zones of low probability and the later would have to follow a sensible approach of quantising over some common x-axis.

We assumed earlier that all properties give rise to independent distributions, but in some cases this may not be so. The same ideas still hold, with the difference that a joint probability distribution for those two would be computed and its log incorporated in the sum of logs T_{Bob}.

An additional limitation is the fact the data available is very subjective, because the same performance can lead to different evaluations from different participants.

Regarding a practical implementation of our model, small adjustments may easily be made, depending on the requirements of the particular setting; for instance, in a fast changing environment ageing of feedback should be used.

3.3 Statement engineering

One can anticipate that some participants may try to deceive the system by submitting statements that appear to be honest but are not accurate, just to accumulate credit by collecting rewards. Is there something to prevent a participant from querying the system to find the current views of others on Bob's performance, and then issue statements that are consistent with that view? Alternatively, suppose that a participant asks the system about ABC cars' reliability. The participant is told that the average reliability rating is 90%. If she buys an ABC car that turns out to be broken, why should she report what she sees rather than just 90%?

That is exactly the behaviour that the system is designed to detect. Honest participants will normally agree with others but sometimes disagree, and – as shown in the previous section – our estimator takes that into account. If a participant's opinions are always consistent with the average – possibly as a result of him querying the system and then submitting an opinion based on the result –, our estimator will mark her as dishonest.

Other users may try to maximise their rewards by being as close to dishonest as possible, but without crossing the threshold, thus submitting as few honest statements as possible to remain marginally not dishonest. For instance, one

may find that for every three honest statements she submits she can add another seven random ones without her nose length crossing the dishonesty threshold. However, participants do not have access to their nose length or to the algorithm based on which it is computed, or even to the thresholds themselves – these are all held in Pinocchio. No immediate information about how close or far they are from being regarded as dishonest is available to them.

Although it may be theoretically feasible to build intelligent software that will learn the behaviour of Pinocchio through a lengthy trial and error process – for instance, by incrementing the proportion of random statements until found dishonest, and repeating several times –, we believe that the cost of such an attack would significantly outweigh its potential benefit. The probationary period is doubled every time a participant is found to be dishonest, so after a few errors the punishment for each new error will be heavy. Also, as the value of the nose length for a participant depends not only on the opinions of that participant, but on the opinions of other participants as well, the system's behaviour may be less predictable.

At the same time, the system does not provide any monetary payments to the users. Rewards can only be cashed for discounts on future queries, and users who are not genuinely interested in obtaining useful information from the system – and more likely to be interested in obtaining short-term benefits by attacking it – will probably not be very interested in non-monetary rewards.

4 Research context

To devise a viable rewards model we studied examples present in existing distributed systems [4, 1, 2] and auction sites, such as the `amazon.co.uk` marketplace and eBay.

Providing incentives for participation is a fairly general research avenue, not necessarily coupled with trust management. Recent studies have focused on providing incentives for cooperation between nodes in wireless ad hoc networks [6], rewarding users who participate in ad hoc routing by allowing them to generate more traffic.

Existing trust management systems operate mainly in three categories of settings: traditional anonymous and pseudonymous *peer to peer systems*, on-line *auction systems* and platforms for *public distributed computing*.

Peer to peer systems. In traditional peer to peer systems, free-riding is widely observed [13, 3], as the fact that participants are anonymous or pseudonymous, and in any case not tied to a real-world identity, operates as a disincentive for active participation. While trust management systems for peer to peer infrastructures have been devised [14], we expect that similar free-riding behaviour would be observed in these systems as well. Users may try to obtain as much information as they can about others, without submitting any new information themselves. Users can escape bad reputations by creating new identities, and also operations are very frequent, therefore providing performance ratings for each one can be significant hassle for a user.

On-line auction sites. Auction systems differ considerably; participants in auction systems have semi-permanent identities, as they are usually somehow tied to a real-world identity – for instance, a credit card. This means that they are not indefinitely able to escape bad reputations easily by creating new identities. Participants care about their reputations more because these are more permanent, and often submit positive feedback about others because they expect reciprocity [15]. The incentive for submitting negative feedback is often a feeling of revenge.

Transactions in auction systems happen in much longer timescales than in peer to peer systems. A purchase of an item can take a few days until it is delivered, while an average download would rarely take more than a few hours. Also, interactions in on-line auction sites happen a lot less frequently; users download files from KaZaA much more often than buying a sandwich maker from eBay. Moreover, the process of purchasing items from auction systems is highly manual, and participants are identifiable. The overall relative overhead of rating a seller in eBay and similar environments is significantly smaller than the one for rating a KaZaA node after a file download.

Additionally, the difference between the level of service that a user expects and the level of service that she actually gets after an interaction plays a significant role in her decision to provide feedback or not. On-line auction sites are inherently risky environments, and clients normally are aware of the risks and are prepared to receive bad service. When the service turns up to be better than expected – which happens often because expectations are low –, clients provide feedback. Clients would provide feedback even for average service, just because it is far better than what they had expected. This provides another insight to why eBay users provide feedback so often.

We believe that the high participation observed in the eBay ratings scheme as [17] can be explained by the reasons mentioned above. Semi-permanent reputations lead to reciprocal behaviour, submitting opinions incurs a much smaller overhead, and clients are happy enough about an interaction to report it more often, as the level of their expectations is low.

Public computing systems. We have outlined some public computing settings in Section 2.2. Participants – peers, or users and servers – are identifiable, and their identities are not subject to very frequent changes. In public computing systems, as in on-line auction sites, users and servers are registered with an infrastructural authority, and this registration often requires binding them with real, legal identities or other forms of semi-permanent identification – for instance, credit cards.

In public computing systems, users take good service for granted. Computing resources are regarded as a *utility* by the users, and the expectations are bound to be high. An analogy can be drawn with other utilities; customers would expect to have electricity at home at any time and electricity providers always expect that customers will pay. The customer will almost exclusively report negative experiences and vary rarely positive ones. In another example, how often does a

regular guest of high-end hotels provide spontaneous positive comments about the experienced quality of service unless it fails to meet his expectations?

One of the consequences is that trust management systems for public computing platforms can not rely on the high participation observed in the eBay ratings scheme and expect spontaneous feedback and Pollyanna-style behaviour. Quite the contrary, as interactions happen frequently and in short timescales – as in peer to peer systems – and the level of expectations is high. There are few inherent incentives for participants to submit feedback. We believe that devising a system to provide explicit incentives for honest participation is crucial for the quality of information held in the trust management system.

5 Conclusion

In this paper we have examined a system for providing incentives for active and honest participation of components in trust management schemes. We propose Pinocchio, a module that has an advisory role, complementary to trust management systems. We suggest rewarding the publication of information and charging for the retrieval, and show that it is possible to provide a credible threat of spotting dishonest behaviour.

Pinocchio is a system that is general enough to co-operate with a large number of trust management schemes in advising when feedback should be rewarded. We have focused more on trust management settings operating in global public computing, but our techniques are generic enough to be applied in other environments.

As an initial experimental setting, we envisage implementing and evaluating Pinocchio as a consultant component attached to XenoTrust [8], the trust management architecture we are developing in the context of our global public computing project, the XenoServer Open Platform [12].

Acknowledgements

We would like to thank Jon Crowcroft, Tim Harris and the anonymous reviewers for their valuable suggestions, as well as Marconi PLC and New Visual Inc for the financial support of Evangelos Kotsovinos' and Alberto Fernandes' research.

References

1. Alfarez Abdul-Rahman and Stephen Hailes. Supporting Trust in Virtual Communities. In *Proceedings of the Hawaii International Conference on System Sciences 33, Maui, Hawaii (HICSS)*, January 2000.
2. Karl Aberer and Zoran Despotovic. Managing Trust in a Peer-2-Peer Information System. In *CIKM*, pages 310–317, 2001.
3. E. Adar and B. Huberman. Free riding on gnutella, 2000.

4. A. Chavez and P. Maes. Kasbah: An agent marketplace for buying and selling goods. In *Proceedings of the First International Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology (PAAM'96)*, pages 75–90, London, UK, 1996. Practical Application Company.
5. S. Snell C.M. Grinstead. Introduction to probability.
6. J. Crowcroft, R. Gibbens, F. Kelly, and S. Ostring. Modelling incentives for collaboration in mobile ad hoc networks. In *Proc. of WiOpt'03*, 2003.
7. Chrysanthos Dellarocas. Mechanisms for coping with unfair ratings and discriminatory behavior in online reputation reporting systems. In *ICIS*, pages 520–525, 2000.
8. Boris Dragovic, Steven Hand, Tim Harris, Evangelos Kotsovinos, and Andrew Twigg. Managing trust and reputation in the XenoServer Open Platform. In *Proceedings of the 1st International Conference on Trust Management*, May 2003.
9. Boris Dragovic, Evangelos Kotsovinos, Steven Hand, and Peter Pietzuch. XenoTrust: Event-based distributed trust management. In *Second IEEE International Workshop on Trust and Privacy in Digital Business*, September 2003.
10. Global Grid Forum, Distributed Resource Management Application API Working Group. Distributed resource management application api specification 1.0, September 2003. Available from http://www.drmaa.org/.
11. Global Grid Forum, Grid Economic Services Architecture Working Group. Grid economic services, June 2003. Available from http://www.doc.ic.ac.uk/ sjn5/GGF/gesa-wg.html.
12. Steven Hand, Timothy L Harris, Evangelos Kotsovinos, and Ian Pratt. Controlling the XenoServer Open Platform. In *Proceedings of the 6th International Conference on Open Architectures and Network Programming (OPENARCH)*, April 2003.
13. Ramayya Krishnan, Michael D. Smith, and Rahul Telang. The economics of peer-to-peer networks. Draft technical document, Carnegie Mellon University, 2002.
14. Seungjoon Lee, Rob Sherwood, and Bobby Bhattacharjee. Cooperative Peer Groups in NICE. In *Infocom*, 2003.
15. L. Mui, M. Mohtashemi, and A. Halberstadt. A Computational Model for Trust and Reputation. In *Proceedings of the 35th Hawaii International Conference on System Sciences*, 2002.
16. Larry Peterson, David Culler, Tom Anderson, and Timothy Roscoe. A blueprint for introducing disruptive technology into the internet. In *Proceedings of the 1st Workshop on Hot Topics in Networks (HotNets-I)*, Princeton, NJ, USA, October 2002.
17. Paul Resnick and Richard Zeckhauser. Trust among strangers in internet transactions: Empirical analysis of ebay's reputation system. In *The Economics of the Internet and E-Commerce*, volume 11 of *Advances in Applied Microeconomics*. Elsevier Science, 2002.