Peer-to-peer reputation systems

Vilen Looga
Helsinki University of Technology
vlooga@cc.hut.fi

Abstract

A peer-to-peer (P2P) reputation system is a mechanism to rate a participant of a P2P sharing community. Trust relationships can be created between participants based on their global ratings, which are created by aggregating personalized ratings. Reputation systems have to be able to operate even when most of the users are malicious. Out of many existing reputation systems, this paper describes Eigentrust, Fuzzy reputation, H-Trust and R2P. Among those four, the last one is a complete P2P system rather than a standalone reputation system, like the others. Brief description of each of those protocols is given. Also test results for each algorithm are presented, if possible. Finally, the paper discusses the feasibility of aforementioned systems by comparing them to protocols, like Bittorrent, that do not posses any reputation mechanisms at all.

KEYWORDS: Peer-to-peer, trust, reputation systems

1 Introduction

Different peer-to-peer (P2P) applications and protocols have become very popular among Internet users. They allow users to share their resources, like files, bandwidth, CPU time or storage space with each other. Ideally, each user would not only use someone else's resources, for example to obtain a file, but also would offer own resources so that others could obtain something that they want. However, one of the biggest problems in P2P networks are users who either do not offer their fair share of resources to the others or otherwise behave against the interests of a sharing community. This paper describes several systems designed to overcome the problem of selfish user behavior. The so-called reputation systems take from each user the information that gives a rating to other peers, with whom the user has communicated. Then, the same information from all peers on the network is aggregated into a global score that describes a peer. Ideally, a global rating of a user should be objective and represent the collective opinion of the whole community. Based on that kind of global knowledge, the users are able to choose with whom they exchange resources, thus creating trust relationships between each other. A typical reputation system can be seen in figure 1.

To understand better why a reputation systems are necessary, we have to start with an assumption that all the peers in a sharing community are selfish[6]. This is problematic, since it is in the interest of a sharing community that everyone participating would provide more resources than they consume. Therefore, there should be some sort of mechanisms that would award good behavior by giving some sort of incentives. Also, those mechanisms should punish bad behavior, for example when the user only consumes resources and does not give anything back. Other types of bad behavior are uploading wrong files instead of the ones that a user requests (this puts an unnecessary load on bandwidth of other users), consuming other users CPU cycles with useless calculations in case of a distributed computing network and etc.

A reputation system has to also be able to withstand an attack from several malicious users that may have conspired to take advantage of the network by giving each other high ratings and thus letting the others participants to believe that they are trustworthy. If this kind of attack is successful, the malicious users can take advantage of their high reputation and start consuming resources of other users in an amount disproportional to their own contribution. Therefore it is crucial that a reputation system is able to deal with malicious behavior of different magnitudes[4]. Fortunately, most (if not all) reputation systems are specifically designed to withstand attacks and still provide reliable ratings even if most of the users participating in the network are malicious.

The remainder of the paper is organized as follows. Section 2 gives a general overview of how reputation systems work and how they are classified into four different categories. There is an example given for each reputation system category. Section 3 discusses feasibility of those systems compared to Bittorrent, todays most popular P2P protocol without reputation systems, and is followed by conclusions.

2 Reputation systems

This section describes some of the proposed reputation systems for P2P networks. Each subsection describes one reputation system, the main idea (algorithm) behind it. Also, where possible, results of comparative performance measurements are given. There is no formal taxonomy for reputation systems yet, however it is possible to categorize them based on some of their features. First of all, we can categorize systems by the way a user obtains ratings about the other participants. The simplest way to obtain a rating is by using a personalized trust system, in which case every user keeps its own database of peers. This means that the same peer can have different ratings in different databases. Each time the user interacts with another peer, the rating of a particular peer is updated based on what happened during the interaction. For example, if the other peer uploaded successfully data that the user requested, the other peers rating in users

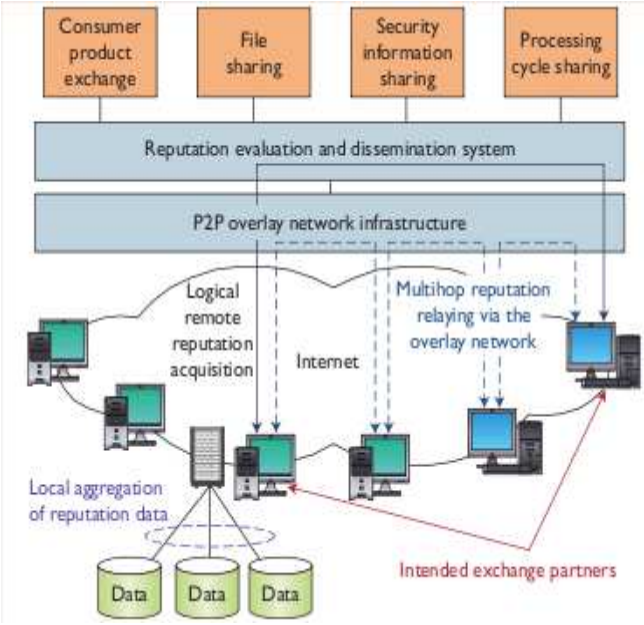

Figure 1: Typical reputation aggregation system. Image from[5]

database improves. This kind of approach is completely decentralized and does not require aggregation and therefore does not add additional load to the network. However it takes a lot of interactions before a meaningful understanding and an objective rating of other peer's behavior can be achieved. For a single user it might be too time consuming to get a sensible database with objective peer ratings, especially if there are many participants. Also, due to not knowing other peers rating beforehand, the user must interact with many peers before the user is able to find the ones he can trust.

This brings us to other alternative, namely global ratings. Those systems allow the user to obtain rating of another peer prior to having any contact with it. Global reputation systems aggregate personalized ratings from peers databases and based on that information create global rating for each participant. The advantage of such a system is that the user can get a objective rating of a peer beforehand. That means that the user can decide whether interaction with that particular peer is desired before connecting to the other peer. Obviously, collecting data for global ratings adds additional load on the network traffic and requires more CPU cycles.

Going further, another way to categorize reputation systems that aggregate ratings is whether they use either full or selective aggregation. In case of full aggregation all the personalized ratings from each individual user are used in calculating a global rating. This method allows to calculate a very precise rating at the cost of network and CPU load. Obviously, as the network grows larger the amount of calculations necessary to perform increases. To counter the problem of network overload, the system can choose to aggregate ratings only from a subset of users, for example based on users rating. This is known as selective aggregation. In this way the accuracy of ratings can suffer, but number of aggregations decreases as does the network load.

2.1 Eigentrust algorithm

This is an algorithm[3]. used to calculate global reputation of peers by using data aggregated from all participants and calculating a global trust vector. This vector determines precisely a users reputation, since it uses full aggregation. The paper presents three versions of the algorithm: basic, distributed and secure.

The basic algorithm does not scale on distributed systems and it does not solve issues arising in P2P networks like inactive peers or malicious collectives, where peers give each other high ratings and use that to take advantage of the system. Therefore, this version of the algorithm is a little interest to us.

The distributed version of the algorithm takes into consideration the distributed nature of P2P systems by engaging all the peers in the network to compute the global trust vector and keeping the message and CPU cycle overhead as low as possible for every individual peer. However, since each peer calculates its own trust value, malicious users could report false values to increase their rating. To tackle that problem another version of the algorithm is presented.

The secure version of the algorithm, which is also distributed by its nature, takes two precautions against malicious users. First, the trust value of a peer is never calculated or stored at that same peer. This means that the peer has no access to its of rating and therefore is not able to manipulate it. Secondly, to recognize malicious users that are reporting wrong trust values, other peers are also involved in computing the values. Multiple peers are given the task to calculate the rating for the same peer. Results from different peers are then compared to each other. The malicious users reported value gets discarded if it differs too much from values reported by other peers.

An implementation of EigenTrust algorithm was compared in a test against a non-trust system on a simulated network. The tests showed that even when the number of malicious peers rose up to 70 percent (X-axis in the figure) the EigenTrust algorithm was able to keep the number of unauthentic downloads around 10 percent (Fig 2). The result was the same in both cases when the malicious peers formed a collective or acted individually.

2.2 Fuzzy reputation aggregation

FuzzyTrust[5] global reputation system with selective aggregation is based on fuzzy logic approach. This system tries to overcome uncertainties introduced by selective aggregation by using fuzzy logics ability to handle such situations. Mainly, FuzzyTrust uses fuzzy logic inference rules to calculate local trust values and then aggregating them to form a global rating. Examples of inference rules are: if transaction is new and has high volume, its aggregation weight is larger; if peer's reputation is good and the volume is low then the aggregation weight is medium; if peer's reputation is bad then the aggregation weight is small etc. The system uses distribute hash tables (DHT) to send reputation infor-

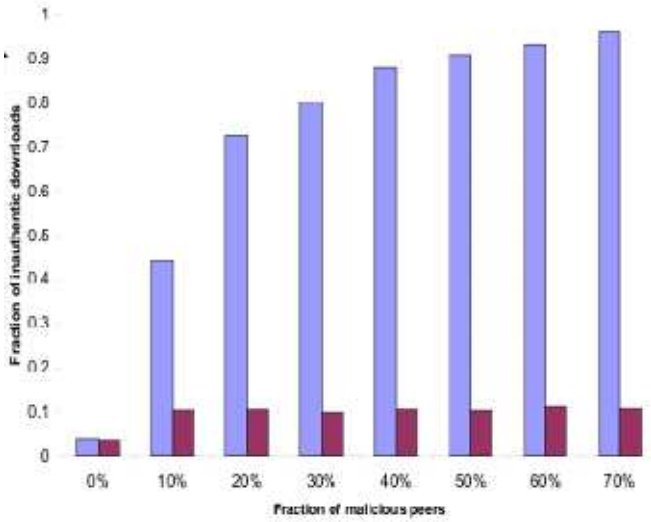

Figure 2: Effect of Eigentrust algorithm. Image from[3]

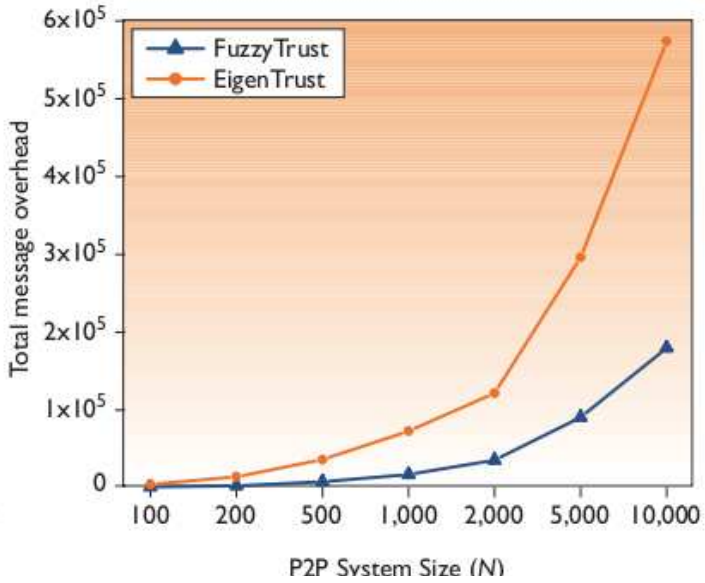

Figure 3: Fuzzytrust vs EigenTrust message overhead. Image from[5]

mation among peers. The system has two stages also called inference steps: local-score computation and global reputation aggregation. In local-score computation, which is performed at the user, the fuzzy inference mechanism tracks adaptively several variables that are considered for calculating a rating. The mechanism is able to adapt in case some uncertainties arise. During the second stage, the global reputation aggregation, the system collect ratings from different peers by applying fuzzy inference mechanisms to different input variables, such as peers reputation, date and volume of transaction etc.

The effectiveness of the system was tested using eBays transaction data. The system was configured so that it prefers super-users, who do a lot of transaction, while transactions from smaller users have less weight in ratio calculation. Secondly, the system prefers to evaluate bigger transactions more often than smaller ones. The same tests were run with EigenTrust algorithm. The results showed that FuzzyTrust was more effective on detecting malicious users and had smaller message overhead (Fig. 3) in global reputation aggregation process. Also, the tests compared convergence times, meaning the time it takes to calculate a ratio of a user, for both EigenTrust and FuzzyTrust. In this case, FuzzyTryust performed somewhat faster.

2.3 H-Trust

Group trust management system H-Trust[8] is based on the h-index aggregation algorithm. It is a personalized trust rating system that uses selective aggregation.

The designers of H-Trust divide the reputation aggregation into five phases.

- *Trust recording phase*. Past transactions with other peers are recorded into Local Service History Table (LSHT) where they are rated based on their importance and quality. Also more recent activities have bigger impact. In this stage there is no rating given to peers yet. Information in LSHT helps against malicious users who try to increase their rating by providing good quality but low priority transactions and then, by exploiting their good rating, provide high priority services with bad quality.

- *Local trust evaluation phase*. In this phase individual user may apply own algorithms to rate peers in local database (Local Trust Rating Table - LTRT) based on the information from LSHT. If no information on a particular peer is available, the system continues with trust query.

- *Trust query phase*. To get information about an unknown peer the user starts querying information from other peers. Since the response might come from a malicious peer, the user must keep a local credibility rating table (LCRT) where information about credibility of peers is kept. Credibility of another peer is based on the quality of previous trust query responses from that peer. The user will evaluate responses about an unknown peer based on the credibility of responder. Obviously, the higher the credibility of a particular peer is the more weight is given to his response. To keep network message load low, the system uses a query threshold, so only a specific number of responses will be considered. For example, if the threshold is 50 and 100 responses arrive, then the user sorts them by credibility and considers only first 50 of them, others are discarded.

- *Spatial-temporal update phase*. Now the user is ready to do transactions with the new peer. After each transaction the user updates his local trust table based on the service quality provided by the new peer. Also, the local credibility table is updated to reflect the quality of trust query responses given by another peers. Peers who

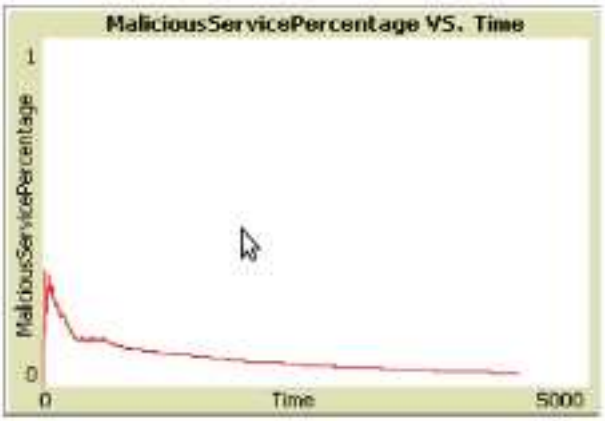

Figure 4: Malicious service decrease over time in a 100 node network. Image from[8]

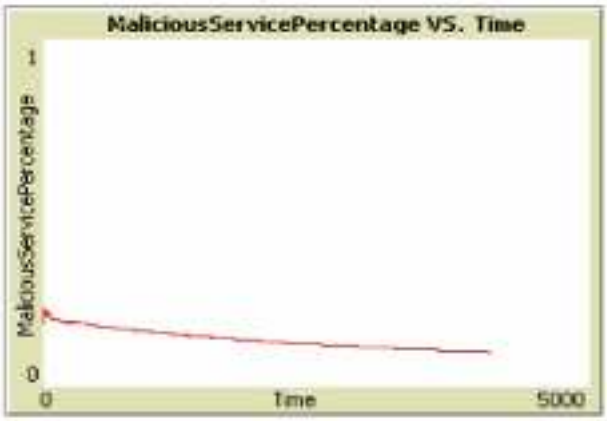

Figure 5: Malicious service decrease over time in a 500 node network. Image from[8]

gave correct response get higher ratings in the LCRT.

- *Group reputation evaluation phase*. The system allows peers to form groups that can provide some sort of service. A user calculates a local rating for a group as follows: some number of peers from the group must have a rating bigger or equal to a threshold set by user; the rest of the group must have a trust in the user bigger or equal to the same threshold. As with peers, among groups that offer the same services, the one with highest rating is selected.

For this system tests were performed where networks with 100 and 500 nodes were affected with 20 percent of malicious users. In both cases most of malicious users were identified after a few aggregations (figures 4 and 5) and the number of malicious services dropped over time. As we mentioned in the taxonomy of reputation systems, personalized reputation systems, such as this one, might encounter problems when the number of participating nodes grows too large. However, the test results show that although it is a personalized system it is able to scale well on a large number of nodes. Unfortunately the systems was not tested to withstand a bigger attack. To do that the percentage of malicious user should have been increased step by step, to see how the system performs with different amount of malicious users. Some of the systems described in this paper even try to simulate situations where most of the users are malicious. Although this situation might be not very likely, it provides information on scalability and robustness. In this case, full potential of H-Trust is not clear.

2.4 R2P

The R2P system[7] is more complex and differs substantially from other systems presented in this paper. As a matter of fact, R2P can be considered as a complete P2P system architecture that employs several other features besides a global rating, like user authentication, role and reputation based access policies and a central portal to control the whole P2P network. This kind of a system architecture is included in this paper because it can give a better understanding of the context in which practical applications of reputation algorithms are used.

The architecture is presented in Fig. 6.

R2P employs two access control policies (ACP) to manage users access to the resources. The first one is a role based ACP, which implements user access by using Public Key Infrastructure (PKI) in the form of X509 certificate. Each certificate is signed by certificate authority and allows a peer to use it for identification to the central server. Based on his role, that is described in the certificate, the user is granted access to the resources available in the network. The second one is a reputation based ACP. This control policy evaluates user's contribution to the network by combining two factors: the amount of data the user has uploaded and the feedback that data recipients give after data transfer. The feedback is then uploaded to the central server where it is processed using a global reputations system called PowerTrust[1], which is an algorithm that uses selective aggregation.

The R2P paper provides results of measurements done on a test network, however those are done to compare R2P with other complete P2P architectures, such as BitComet and eMule. Reputation system comparisons are not presented, therefore it is not possible to compare it to other systems presented in this paper. Comparison with other systems could be considered useless since there is no way that malicious users could strive in an R2P network. One could argue that because each user is identified with a digital signature, there is no need to consider malicious behavior. If any peer would show signs of harmful behavior, he would be identified and revoked of his rights to access resources of the network.

3 Discussion

In this paper we presented several reputation based systems and analyzed the test results presented in the respective papers. To go further, this section will try to evaluate the success of reputation systems in the real world. One could argue that a new design can be considered successful only if it is implemented in an actual product and widely adopted by user. This is where the reputation systems seem to fail, since none of the algorithms mentioned in this paper is used in any

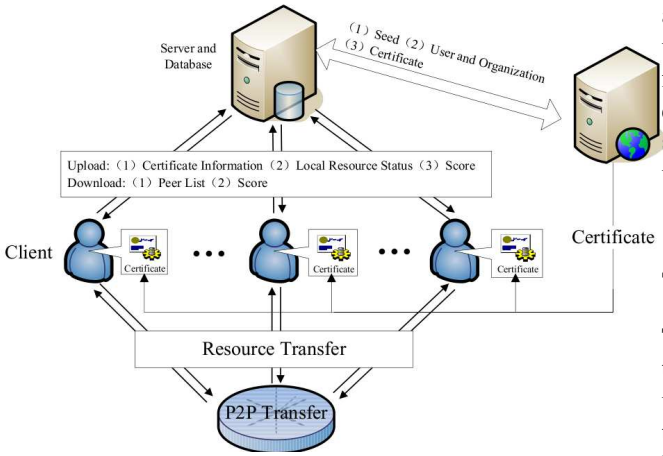

Figure 6: R2P architecture. Image from[7]

of today's most popular P2P software. Those reputation systems have not been tested on a large scale at all. As a matter of fact, it seems that today's P2P networks prefer to use alternative ways to regulate their user communities. Protocols that did have built-in rating systems, like eMule or Kazaa are fading away.

Take for example the most popular P2P system [2] nowadays, Bittorrent, that does not use any reputation system, at least not as a part of the protocol. The tracker server, a crucial part of the Bittorrent protocol, collects data of how much data each user has uploaded and downloaded. However, the tracker does not use any reputation systems. It should be noted that some websites do require registration and login, before a user can download a torrent file or use the tracker provided by the website. The purpose of that is mainly to monitor and manage the users by means of some reputation system employed by the website. Usually, there are some simple rules that state how much the user has to share to continue being a member of the community. Therefore, the user may be a subject to some rules enforced by the reputation system of the website, although he is using a protocol without a built-in reputation system. Nevertheless, this raises the question, if we really need an reputation system built into the protocol, as proposed by the reputation systems described in this paper.

Judging by the popularity of the Bittorrent protocol, it seems that although lacking any rules to award good behavior and punish malicious users, it is still able to provide satisfactory level of quality and incentives to most of the users participating in the community. It seems that each sharing community, that uses Bittorrent as a sharing protocol, is able to decide themselves what kind of reputation system, if any, they need. Different Bittorrent communities seem to be able to achieve the goal of a system where not only regular users, who mostly consume the available resources, but also the super-users, who mostly provide their resources to others, are interested in participating.

Looking back into history of P2P software, we can see how popular P2P clients of previous generations faded away as did the reputation systems used in them. However, reputation systems can still have a place in niche architectures, for example the R2P system presented in this paper. However for now it seems that wide scale adoption of reputation systems, for example in a form of a popular P2P software, is unlikely.

4 Conclusion

This paper described the motivation behind creating reputation systems. As we pointed out the main reason to use a reputation system is to minimize the effect of malicious users on the network, whereas the goal to provide incentives for good behavior is also achieved. Further on, we gave a short taxonomy of reputation systems and based on that introduced several systems that use various methods to rate peers. Those ratings are used to create trust between peers. Unfortunately, due to different testing methods we could compare only two of the presented systems with each other. However the test results of each individual system show that it is able to deal with malicious behavior quite well. Finally, we discussed the adoption of reputation systems in real world and, although they are not used in popular P2P clients, they have their place in specialized solutions such as R2P.

References

[1] Powertrust: A robust and scalable reputation system for trusted peer-to-peer computing. *IEEE Trans. Parallel Distrib. Syst.*, 18(4):460–473, 2007. Member-Zhou,, Runfang and Fellow-Hwang,, Kai.

[2] ipoque. Internet study 2007. http://www.ipoque.com/resources/internet-studies/internet-study-2007. Referenced at 11.02.2009.

[3] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina. The eigentrust algorithm for reputation management in p2p networks. Working Paper 2002-56, Stanford InfoLab, 2002.

[4] S. Marti and H. Garcia-Molina. Taxonomy of trust: Categorizing p2p reputation systems. Working Paper 2005-11, Stanford InfoLab, 2005.

[5] S. Song, K. Hwang, R. Zhou, and Y.-K. Kwok. Trusted p2p transactions with fuzzy reputation aggregation. *Internet Computing, IEEE*, 9(6):24–34, Nov.-Dec. 2005.

[6] W. Wang and B. Li. To play or to control: A game-based control-theoretic approach to peer-to-peer incentive engineering. In *In International Workshop on Quality of Service*, pages 174–192. ACM Springer-Verlag, 2003.

[7] Y. Xia, G. Song, Y. Zheng, and M. Zhu. R2p: A peer-to-peer transfer system based on role and reputation. *Knowledge Discovery and Data Mining, 2008. WKDD 2008. First International Workshop on*, pages 136–141, Jan. 2008.

[8] H. Zhao and X. Li. H-trust: A robust and lightweight group reputation system for peer-to-peer desktop grid. *Distributed Computing Systems Workshops, 2008. ICDCS '08. 28th International Conference on*, pages 235–240, June 2008.