AN INDEPENDENT REPUTATION SYSTEM FOR P2P NETWORKS

Chaiyasit Tayabovorn and Songrit Maneewongvatana
Department of Computer Engineering, King Mongkut's University of Technology Thonburi
126 Pracha-Uthit Road, Bangmod, Thungkru, Bangkok, 10140, Thailand

Keywords: Peer-to-peer, reputation, trust, decentralized.

Abstract: Reputation is a very sensitive and important component in Peer-to-Peer (P2P) network. In this type of network, a user who has a well reputation can persuade others to do transactions with her easily. Most of current P2P networks use centralized reputation systems which have some drawbacks such as scalability, single point of attack and monopoly of reputation information. Another alternative is to use P2P reputation system. But most of these P2P reputation systems do not work well in practice due to some strict assumptions. In this paper, we propose a novel P2P reputation system that separates the entire reputation community. This results in reputation system that is more flexible, scalable, and controllable. We also propose a reputation assessment algorithm to evaluate the resources that have never been rated before.

1 INTRODUCTION

The reputation system is mostly used in eCommerce system like eBay. Its main function is to gather all information that relates to an entity in order to assess that entity's reputation.

Reputation systems, regardless of their type - "*centralized*" or "*Peer-to-Peer*" (P2P), have some weaknesses. For *centralized reputation systems* (CRS), the first weakness is a single point of attack allowing attackers to hack the system easily. To protect all attacks aiming at a single place, security cost is prohibitively high. Also, CRS can cause a monopoly and a corruption by CRS's owner. As all *reputation information* (RI) control is in the hands of a single authority, RI can be faked easily. The validity of RI is hard to monitor. This problem often arises in online-shopping environment. The website's administrator allows buyers to leave rating on products offered on the website. However, the corporate has absolute control of the RI and may ask the administrator to fake new ratings or alter actual ratings to deceive new-buyers to trust products that have been poorly rated by previous buyers.

From weaknesses of CRS mentioned above, *P2P reputation system* (P2PRS) becomes an attractive alternative. The idea of P2PRS is to spread RI to all peers, in order to avoid a single point of attack and a monopoly. However, nowadays, no P2PRS has been deployed successfully in practice. The weakness of P2PRS is that it works in only theories and assumptions are too restrictive in practice. For example, the assumption that peers in P2PRS can locate each other without central entity's helps, but such peers are required to be online via only public IP addresses in order to see all each other. Another weakness of P2PRS is that it does not allow an entity's RI to be referred across communities. For example, a user has to recreate his own reputation every time when joining new community, even if in fact he should be able to refer his own existing reputation created in prior acceptable communities.

In this paper, we present a novel P2PRS that focuses on practicality issues. Our key concept is to create a P2PRS in form of a reputation community, which is separated from the traditional P2P general service communities. The separation between two types of communities will increase the reliability, flexibility, and scalability. In addition, we propose a novel algorithm that helps an entity's reputation assessment be possible even if the entity has never been rated before.

The rest of this paper is organized as follows. Firstly, we describe our P2PRS overview in Section 2. We explain our P2PRS's interaction protocol in Section 3. All discussions are covered in Section 4. Finally, we brief related work in Section 5.

Tayabovorn C. and Maneewongvatana S. (2007).
AN INDEPENDENT REPUTATION SYSTEM FOR P2P NETWORKS.
In *Proceedings of the Ninth International Conference on Enterprise Information Systems - SAIC*, pages 249-252
DOI: 10.5220/0002375502490252

2 SYSTEM OVERVIEW

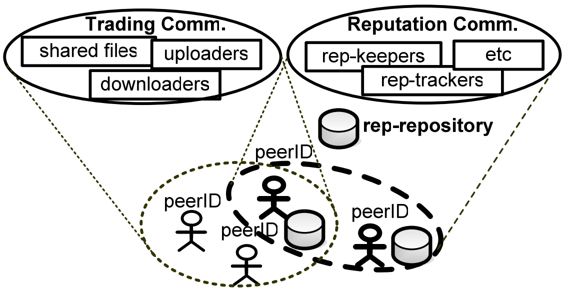

Figure 1: Our P2PRS's virtual network.

Figure 1 illustrates our P2PRS's virtual network that includes: "*trading*" and "*reputation*" communities. *Trading community* (TC) is a general service P2P network. One of common P2P services is filesharing (BitTorrent, 2007), which will be the service that is discussed throughout the paper. A separated community, *reputation community* (RC) is a P2P network that serves reputation related transaction. Both communities are separated apart but some peers can be part of both communities if it can comply with each community's roles.

In our P2PRS, the main service is to sharing *shared files*. The attributes of shared files are advertised via the *advertising file*. The advertising file contains important information for downloading files. In our system, we append information into the advertising file so that it provides the link to the repository of RI of the associated shared file.

In TC, roles of a peer can be uploader and downloader or both, which are similar to roles in general P2P filesharing networks. In RC, possible peer roles are *rep-keeper* and *rep-tracker*. Main function of a rep-keeper is to keep RI of resources (shared files in our case) in TC. Rep-tracker's functions are to locate the right rep-keeper and to perform bootstrapping services.

In order to persuade peers to join RC, credit system may be used. A peer is given higher credit if it provides more of reputation related service.

3 INTERACTION PROTOCOL

The interaction protocol of our P2PRS can be divided into the following consecutive steps.

3.1 Reputation Installation

This is an initial step to make shared files be able to support the reputation query, which are requested by peers of TC, which will be referred to as *clients*. Clients who want to share their own files must update the reputation field in the associated advertising file. This can be done by sending an initiated version of the advertising file to rep-tracker. Afterwards, rep-tracker updates the reputation field in the advertising file by generating a reputation ID <rid>, as shown in Figure 2. Rep-tracker also has to map the rep-keepers responsible for the <rid> by recoding such mapping information into the rep-tracker's database. Finally, the rep-tracker sends the updated advertising file to the requesting client.

```
<reputation>
  <tracker name="rep-tracker1">
    <rid>233dba2719a37</rid>
    <url>x.x.x.x:9701</url>
    <signature>
         ([19, 1E, F8, A2, 33, ...])
    </signature>
  </tracker>                   rep-tracker's
  <tracker name="...">           signature
    ...
```

Figure 2: An example of tracker elements attached.

3.2 Reputation Information Access

Once the RI is installed, other clients can access to RI repository held by the responsible rep-keepers. Clients can issue two commands: view and update RI of requested shared file. To locate the responsible rep-keepers, a client sends a request message to the rep-tracker specified in the advertising file. The request message indicates the id of RI of the shared file and the command from the client (view/update). After rep-tracker finishes parsing the request message, it replies the requesting clients with a *coupon* message, which contains information to locate the responsible rep-keepers.

Once the locations of the responsible rep-keepers are known, the client can access RI repository. This can be done by sending the copies of coupon messages to all rep-keepers listed in the coupon message. Involving rep-keepers look for the requested RI in their own RI repository. Any rep-keeper that has the requested RI will execute the command in the coupon message. Finally, each rep-keeper will send a result message to the requesting client. The result message would be either the requested RI or the updated result data, depending on the command in the requesting message.

3.3 Reputation Assessment

This last step relates to the assessment of the reputation of a shared file which a client needs. We called this step *reputation assessment* (RA). In this section, we propose a practical algorithm to assess the reputation. The often-seen problem of RA in practice is the lack of RI-like (rating) score which can be used in during a reputation assessment. To mitigate the problem, our RA algorithm relies on correlation of previously mutual rating between uploader and *assessor* (a client that wants to assess the reputation of a shared file). This method allows the assessor to assess the reputation of the uploaded file even if that file has never been rated before.

The assumption that our algorithm relies on is that most uploaders share their files when they are satisfied with those files and want other to download them. So, if such uploaders are allowed to rate those files according to their own perspective, they are very likely to rate their own files positively.

Based on this assumption, the main formula to assess the reputation is given in Equation (1). Positive rating is represented by value 1 in Equation (1). Negative rating is set to be -1.

$$rep = \left(\sum_{i}^{n} \left(1 \times corr(\mathrm{A}, \mathrm{U}_i) \right) \right) / n \qquad (1)$$

where *rep* is a reputation value; "corr(A,U_i) is a correlation function; A and U_i are set of the previously mutual rating given by an assessor and i^{th} uploader, respectively; *corr* function is shown in Equation (2).

$$corr(X,Y) = \frac{N\sum xy - (\sum x)(\sum y)}{\sqrt{\left(N\sum x^2 - (\sum x)^2\right) \times \left(N\sum y^2 - (\sum y)^2\right)}} \qquad (2)$$

where N is the number of mutual ratings between X and Y; x and y are individual rating to a shared file by X and Y respectively.

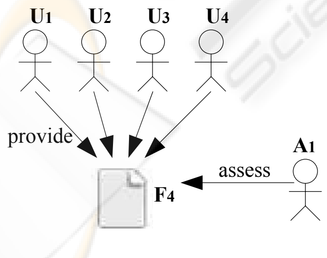

Figure 3: An example's reputation assessment.

We use Figure 3 as an example to demonstrate our algorithm. We assume that the uploaders $U_{1\text{-}4}$ provide the same file F_4, which is just released and has never been rated before. The assessor A_1 wants to assess reputation of F_4 but its reputation has not been assessed. In this case, A_1 computes the coefficients of correlation, *corr*(A_1,U_j) where $1 \leq j \leq 4$ and finds final *rep* in Equation (1). All coefficients are computed via the previously mutual ratings on $F_{1\text{-}3}$ given by $U_{1\text{-}4}$ and A_1. The previously mutual ratings are shown in Table 1.

Table 1: Previously mutual ratings given by $U_{1\text{-}4}$ and A_1.

FID	U_1	U_2	U_3	U_4	A_1
F1	0	1	1	N/A	0
F2	1	1	1	N/A	1
F3	0	-1	-1	N/A	-1
C	0.866	1	0.866	0	

Possible rating values are -1, 0, and 1 representing negative, neutral, and positive ratings respectively. We use Equation (2) to compute the coefficient of the correlation (***C***), whose value ranges between [-1, 1], where -1, 1, in turn stand for a reverse and a consistent relationship in prior rating. If there is no mutual rating between two peers, ***C*** is set to be 0, e.g. the case of [U_4, A_1]. For a peer whose ratings are all the same values, like all (1,-1 or 0), in using Equation (2), ***C*** is undefined even if the ratings of both peers completely agree. In this case, we set ***C*** heuristically to be 0.75.

Due to space limitation, we show only the computation for U_1 and A_1 as an example. Here are values to take in Equation (2) to compute the correlations: N=3; $\sum xy$=1; $\sum x$=1; $\sum y$=0; $\sum x^2$=1; $\sum y^2$=2.

$$0.866 = \frac{(3 \times 1) - (1 \times 0)}{\sqrt{\left((3 \times 1) - 1\right) \times \left((3 \times 2) - 0\right)}}$$

We substitute coefficients in Equation (1) with the values obtained from Equation (2):

$$0.683 = \left(0.866 + 1 + 0.866 + 0\right) / 4 .$$

The result *rep* computed from Equation (1) will have a value in the range of [-1, 1]. A positive value implies that the assessed file tends to satisfy the assessor, whereas a negative value implies that it tends to dissatisfy the assessor. In this example, *rep* is 0.683, which means that F_4 has a rather good reputation and tend to satisfy the client A_1.

4 DISCUSSIONS

In this section, we will discuss our P2PRS in aspects of the accessibility and the scalability.

4.1 Accessibility

We define *accessibility* to be the degree of how accessible RI is for the clients. Accessibility is a well gauge, which can affirm the efficiency of a P2PRS. In this paper, we measure accessibility by observing the probability that requested RIs are reachable by clients.

To support our idea, we implemented a simple simulating program imitating behaviour of peers in both communities. We evaluate RI's accessibility by looking at the number of hops needed to reach peers that are expected to contain the requested RI. We compare the accessibility to RIs of our P2PRS and a normal P2PRS. Normal P2PRS does not have a separate reputation community.

We set the following parameters in our simulation. The numbers of peers in TC and in RC are 3000 and 500, respectively. Each TC peer can request at most 100 files and can share at most 4 files. The total number of files in all of the peers in TC is 650. Probabilities in serving RI of a peer in TC and a peer in RC are 0.5 and 0.95, respectively. Note that peers in RC have a higher probability in serving RI since its main task is to provide RI.

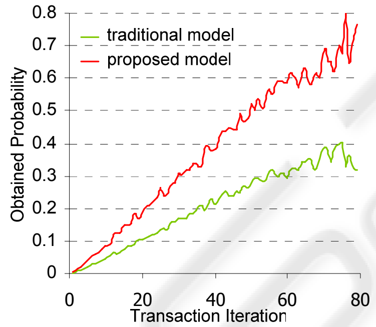

Figure 4: The result after the simulation of both models.

The result is shown in Figure 4. Y-axis is the average probability to obtain the requested RI successfully, whereas X-axis is the iterations to trigger all TC's peers to download, upload, and rate files according to the configuration given. The graph has two lines. The higher line is measured from our P2PRS while another lower is measured from the normal P2PRS. Both lines in graph indicate that our P2PRS has higher accessibility.

4.2 Scalability

Our P2PRS increases the load scalability by separating loads between the filesharing and the RI related tasks. This improves the efficiency of the total loads**.** Also, our P2PRS lets the RI of shared file be referred to from multiple RCs. The load of serving RI tasks can be distributed among RCs.

The geographic scalability of a P2P network mainly replies on helps of one or more local servers, e.g. Trackers in BitTorrent. Our P2PRS relies on rep-trackers which act as local servers called rep-tracker but we can prevent a single point of failure and monopoly by allowing a community to have more than one rep-tracker and allowing RI to be accessible across multiple TCs and RCs to makes RI more diverse.

Most of previous work focuses on pure P2P's principle, ignoring the administration concerns. Our P2PRS makes the administration scalable by separating RC and allows multiple rep-trackers in a RC. This makes it easy to update different strategies coping with new attacks, without impacting on file sharing loads or other main activities of TC..

5 RELATED WORK

EigenTrust Algorithm (Sepander, 2003) presented a method that each peer *i* is assigned an approximate unique global trust value; reflect the experience of all peers in the network with peer *i*. The prominent point of this paper is to account a global reputation value based on EigenTrust algorithm, using the Eigen vector matrix.

Managing Trust (Karl, 2001) proposed a method of preventing the malicious agents with simple method of data mining theory, using statistical data analysis of former transactions and placement reputation with a decentralized storage system called P-Grid.

REFERENCES

Sepandar, D. K., Mario, T. S., & Hector G. 2003. The EigenTrust algorithm for reputation management in P2P networks, *WWW2003*, 640-651.

Karl, A., & Zoran D., 2001. Managing trust in a Peer-to-Peer information system, *CIKM'01*, 310-317.

What is BitTorrent? Retrieved March 11, 2007, from http://www.bittorrent.org/introduction.html