Future Generation Computer Systems 27 (2011) 1135–1141

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

R^{2}Trust, a reputation and risk based trust management framework for large-scale, fully decentralized overlay networks

Chunqi Tian [a,b,*], Baijian Yang [c]

[a] The Key Laboratory of Embedded System and Service Computing, Ministry of Education, Tongji University, Shanghai, 200092, China
[b] State Key Laboratory of Networking and Switching, Beijing University of Posts and Telecommunications, Beijing, 100876, China
[c] Department of Technology, Ball State University, Muncie, IN 47306, USA

ARTICLE INFO

Article history:
Received 4 March 2010
Received in revised form
12 February 2011
Accepted 8 March 2011
Available online 23 March 2011

Keywords:
Peer-to-Peer
Trust
Reputation
Credibility

ABSTRACT

Peer-to-peer (P2P) networking is widely used to exchange, contribute, or obtain files from any participating user. Building trust relationships between peers in a large-scale distributed P2P file-sharing system is a fundamental and challenging research topic. However, it is difficult to build a good trust relationship with the traditional mechanism. Recommendation based trust model from social relationship can be adopted to resolve the problem. But it faces the challenges of subjectivity, and experiential referral weighting. This paper presents R^{2}Trust—a robust Reputation and Risk evaluation based Trust management model. Our novel framework uses both reputation and risk to evaluate the trustworthiness of a peer and it is applicable for unstructured P2P networks. The model will evaluate peer trust values from direct interactions and peers referrals. R^{2}Trust also distinguishes the credibility of peers. As a result, the aggregated trust value will filter out the noises and reflect more accurate trust values. The proposed R^{2}Trust can also defense against several malicious attacks, such as simple malicious attacks, collusive attacks, and strategic attacks. Our experimental results show that, compared to the existing trust models, our model is cleanly a winner when security is the major concern of a system.

1. Introduction

In a peer-to-peer (P2P) system, peers can freely join and leave the system and the group membership is very dynamic. Due to its openness and lack of validation, a P2P system is vulnerable to such a kind of attack where some peers maliciously poison the system with corrupted data or harmful services [1]. To protect the interest of participating peers, it is therefore very important to ensure the authenticity of shared resources. Trust and reputation management is introduced to the P2P systems as a solution to promote a healthy collaboration relationship among participants. Currently, most trust and reputation systems focus on evaluating the credibility of resource providers. One approach [2,3] adopts a global trust value for every participating peer. But calculating and maintaining global trust values may incur significant overheads in a large-scale P2P system and a single trust value by itself is not sufficient to identify the behaviors of a peer. Another approach [4–8] allows each peer to calculate and store its own trust values of all the peers it has conducted transactions with. Generally, when a peer requests a reputation value of an unknown peer from its friends/neighbors, it typically relies on a flooding-like method. Unfortunately, flooding is not a scalable solution for a large P2P system.

In this paper, we present a robust reputation and risk evaluation based trust management model for decentralized P2P systems. The major contributions of this paper are illustrated as follows.

First, in order to portray the unpredictable and uncertain behaviors from malicious peers, we introduce risk value into the computation of a peer's trust value. The risk value is used to measure various malicious behaviors, such as fluctuating behavior and misusage of trust. In our design, the weights of the reputation and risk are adjustable so that they can be configured and effectively applied to different environments with different requirements.

Second, although some existing approaches [8,9] consider the credibility of a recommender, they do not evaluate and update the recommender's credibility effectively in the presence of dishonest or unreliable referrals. The method we proposed for credibility computation and update is proven to be able to efficiently distinguish reliable peers from deceptive or unreliable peers. This is achieved by quantifying the credibility of a referral and minimizing the effect of ratings from denigrating or collusive peers.

Third, the ratings in most existing approaches are binary [2,4,8]. In the binary ratings, a peer rates the services from another peer as one of two values, commonly interpreted as either one

* Corresponding author at: The Key Laboratory of Embedded System and Service Computing, Ministry of Education, Tongji University, Shanghai, 200092, China. Tel.: +86 13585780926.
E-mail address: tianchunqi@163.com (C. Tian).

0167-739X/$ – see front matter
doi:10.1016/j.future.2011.03.006

(e.g., positive or satisfactory) or zero(e.g., negative, unsatisfactory). Binary ratings may not adequately represent a peer's experience of the quality of service with other peers, e.g., the quality of files the peer sends. Our approach considers quality of service as probabilistic ratings in the interval [0, 1] and focuses on how to aggregate these ratings.

The paper is organized as follows. In Section 2, we review some existing works. In Section 3, we describe our proposed R^2Trust model for peers' trust evaluation followed by presenting our simulation and experiment results in Section 4. We then conclude our work in Section 5.

2. Related works

EigenRep [2], is one of the initial proposals on reputation ranking aggregation. In EigenRep, a unique global trust value is assigned to a peer and is updated by normalizing and aggregating local trust values from other peers. The authors also proposed a distributed iterative algorithm to calculate and update a global trust vector at each node. The trust vector is then used by the node to isolate malicious users and reward the peers with good reputation. PowerTrust [3] dynamically selects a number of most reputable nodes as the power nodes. By using a look-ahead random walk strategy and leveraging the power nodes, PowerTrust improved the accuracy of global reputation and the speed of aggregation. However, establishing global trust values for every peer in a large-scale P2P system seems not necessary. In addition, it might not be feasible either.

XRep proposed by Cornelli et al. [4] employed a reputation-based approach to evaluate the reputation of peers through a distributed polling algorithm. A binary rating system was adopted by using the Gnutella query broadcasting method with TTL limit. However, neither formalized trust metrics nor experimental results were presented to validate their approach. The scheme [5] proposed by Tian et al. uses advanced Dempster–Shafer (D–S) Evidence Theory to model the trust relationship among peers in P2P networks. D–S evidence theory is a prevailing approach to tackle problems with uncertainty. But the traditional combination rule will produce unreasonable results when evidences are greatly in confliction with each other. In [5], authors proposed a new combination rule when building the trust model. The experiment results proved it was robust and scalable. In ARTrust [6], the trust model includes two parts: reputation evaluation and penalty evaluation. The penalty evaluation is further divided into conflicting value and misusage value. By utilizing multi-factor evaluation, ARTrust can improve trust management by separating the short-term behaviors from the long-term behaviors of the peers.

Xiong and Liu [8] presented a reputation-based trust supporting framework. They introduced three basic parameters and two adaptive parameters. They incorporated the concepts of a trust value and the similarity with oneself to compute credibility and satisfaction. In [9], trusted collaboration was proposed to help manage the trust and a trust value will not be directly used. Instead, its fairness was evaluated by considering the opinions of all the peers through a voting mechanism. This approach may also suffer from the scalability problem because it is complex to collect votes from a great number of peers, and it may negatively impact the performance due to network constraints of each participating peer. TrustWalker [10] is a random walk model that considers both trust values and recommendation value. It defined a confidence level of each recommendation as a reputation quantifier. SFTrust [11] distinguishes the trust value for providing services from providing feedbacks. It also designed and implemented a framework to store, compute and update trust values. In short, this type of approach trust evaluation is mainly derived from the direct transactions without factoring in the quality of the evaluation, the quantity of the transactions and the time of the transactions, etc.

Liang and Shi [12] proposed PET, a personalized trust model with reputation and risk evaluation for P2P resource sharing. In PET, risk factor is adopted to complement the reputation evaluation. Risk evaluation represents the influences of short-term behaviors while reputation is the accumulative assessment of long-term behaviors. The main contribution of PET was that it introduced risk factor when compute the trust value of a peer. Ignjatovic et al. [13] used a sequence of time-based experiences to evaluate how much a peer can trust other peers and their recommendations. A concept called 'weight of evidence' was introduced to adjust the referral of a peer based on its reputation. Therefore, the trust management system they proposed computes trustworthiness, reputation and weight of evidence to achieve more accurate results.

In [14], a peer maintains reputation and trust ratings for a selected number of peers. It will periodically advertise its local reputation ratings to others. Peers received the updates from others will then utilize a Bayesian approach to determine whether or not the second-hand reputation ratings should be accepted to modify their own local ratings. Ma et al. [15] proposed a probabilistic matrix factorization to improve the accuracy of a reputation network when the data is sparse. In addition to the data from users' records, it also used the ratings from social networks. In [16], a Bayesian learning was applied. In this framework, the first-hand information was exchanged more frequently while the second-hand information was merged with the reputation ratings. Wang and Vassileva [7] proposed another Bayesian network-based trust model to implement recommendations-based reputation rating. They differentiate two types of trust in their system. One is the trust of a host's capability to provide the service, and the other is the trust of a host's reliability to provide recommendations.

FuzzyTrust [17] is a trust model that uses fuzzy logic inferences to handle the situation when complete information was not available from the peers. It was tested over public domain using real-world transaction data from eBay. The results demonstrated that it was more effective than EigenRep. In [18], an adaptive reputation-based trust framework was proposed to minimize the risk from malicious peers and motivate selfish peers to provide more services. ARRep [19] aimed to deal with on–off, bad mouthing and collusive cheating attacks in P2P networks.

Wang et al. [20] proposed another social-network based reputation ranking algorithm. It is capable of inferring reputation ranks more accurately when the system is under front-peers attack. The scheme proposed in [21] uses local trust evaluation to warn other peers not to download a file from a suspicious peer. The scheme aimed at reducing the attack surface of malware and assumed no local file was infected. Although this type of trust management schemes considered malicious attacks, effective measures are still missing in their design and implementation.

In this paper, we propose a robust and efficient reputation mechanism in P2P systems. We also study possible attacks to reputation mechanisms in P2P systems. We construct a mathematic model of referral using credibility, and then adopt it to aggregate the referrals. Finally, we discuss such problems on security as denigration, collusion and oscillating behaviors of malicious peers and also address the solutions to these problems, consequently conduct many experiments to validate them.

3. R^2Trust model for P2P networks

In this section we will present a novel Reputation and Risk evaluation based trust model, R^2Trust. We are interested in applying R^2Trust in file-sharing type of services over unstructured P2P networks.

Table 1
Description of quality of service.

File quality	Description
G (Good)	The file is as good as expected.
C (Common)	The file is correct, but with some degradation.
I (Inauthentic)	The file is inauthentic.
M (Malicious)	The file is malicious (e.g. virus or Trojan Horse).

3.1. Trust value overview

Our trust metric is composed of two parts — trust value and risk value. The overall trust value of a peer is evaluated by subtracting the risk value from the trust value. For each candidate, the trust value is calculated from the direct trust value and the reputation value. The risk value is derived from existed negative experiences. Let TV_{ij} denote the overall trust value of provider j from the view point of peer i; T_{ij} and R_{ij} denote the trust value and risk value of peer j, respectively; and a, β denote the corresponding weight for T_{ij} and R_{ij}. Therefore the overall trust value for peer j at peer i is:

$$TV_{ij} = \alpha T_{ij} - \beta RV_{ij} \quad (0 \le a,\ \beta \le 1). \tag{1}$$

The values of a, β can be chosen based on how optimistic a peer is. If peer i is optimistic about provider j, it will choose a bigger α and a smaller β such that the overall trust value is less affected by the risk value. On the other hand, if peer i is pessimistic about provider j, it will choose a smaller α and a bigger β so that the overall trust value is more sensitive to the value of the risk.

The calculation of a trust value needs two parts of information: direct trust value and reputation value. Direct trust value can be obtained when a peer has direct transactions with a provider. In reality, a peer may not have dealt with a provider before so it will have to rely on other peers' recommendation, i.e. reputation value, to evaluate how much it can trust the 'new' service provider. Reputation value is measured by aggregating all the referrals from other peers. Let T_{ij} denote the trust value from node i to node j. It is defined in (2).

$$T_{ij} = \zeta DT_{ij} + (1 - \zeta) RE_{ij}, \tag{2}$$

where DT_{ij} is the direct trust value from peer i to peer j, RE_{ij} represents the reputation value of peer j, ζ is the confidence factor and $0 \le \zeta \le 1$.

Confidence factor ζ describes how confident peer i is regarding its direct trust value on peer j. Generally speaking, if peer i has had sufficient transactions with peer j, then i knows j well enough and does not need many feedbacks from other peers about the trustworthiness of j. Otherwise, peer i will weigh more on the recommendations coming from other peers. If peer i has performed h transactions with node j, then ζ can be defined as follows:

$$\zeta = \begin{cases} h/H_{LMT}, & h < H_{LMT} \\ 1, & else \end{cases} \tag{3}$$

where H_{LMT} is the threshold of direct transaction numbers.

In this paper, we use P2P file-sharing systems as an example to measure a peer's reputation. In Section 2, it is clear many previous works barely considered the quality of service providers in the evaluation. We believe the quality of service is a very important factor when in the process of trustworthiness evaluation and should be taking lightly. We first classify the services into four categories based on the quality of the files provided by the corresponding peers, as shown in Table 1. We formalize the quality set $Q = \{G, C, I, M\}$. Note that the coarse-grain classification introduced is flexible: more classes or subclasses can be introduced if it is necessary.

We then define a Map function $f(q)$ to quantify the quality of service. From Eq. (4), we can see that if the quality is rated either as Good or Common, the corresponding rating value will be a positive number. As a result, it will slightly boost one's direct trust value as further explained in Section 3.2. If the quality of file is Inauthentic or Malicious, then the value of the rating will be a negative number, which will then have a big impact to drop one's direct trust value.

$$f(q) = \begin{cases} v_1,\ q = G, & 0 < v_1 < 1 \\ v_2,\ q = C, & 0 < v_2 < v_1 \\ v_3,\ q = I, & -1 < v_3 < 0,\ |v_3| > v_1 \\ v_4,\ q = M, & -1 \le v_4 < 0,\ |v_3| < |v_4|. \end{cases} \tag{4}$$

3.2. Direct trust value

Assume $R_{ij}^{t_k}$ is a local rating of peer i for peer j during time period t_k $(1 \le k \le n)$ and their interaction number is N_{ij}, we define the local rating $R_{ij}^{t_k}$ peer i to peer j as follows:

$$R_{ij}^{t_k} = \sum_{N_{ij}} f(q)/N_{ij}, \quad q = G, C, I, M,\ N_{ij} \ne 0. \tag{5}$$

If $N_{ij} = 0$, we define $R_{ij}^{t_k} = 0$. We introduce the time factor when computing the reputation rating due to the fact that a typical P2P system turns to be very dynamic and the behaviors of a peer could change from time to time. It is our belief that most recent ratings may be more accurate to reflect a peer's reputation in the near future. Also, because the weight distribution is typically experiential and subjective in its nature, we define the decay function in such a way that its impact can be adjusted in a given system by assigning proper value to its parameter.

Decay function λ, which is a timing discount function, is described as $\lambda(k) = \lambda_k = \rho^{n-k}$, $0 < \rho < 1$, $1 \le k \le n$, where the function value λ_k is the decay factor for the kth time window.

Direct trust value DT_{ij} is computed directly from the peer i's historical ratings for peer j. If j has a number of direct interactions with i during period $[t_{start}, t_{end}] = [t_1, t_2, \ldots, t_n]$, We define DT_{ij} as follows:

$$DT_{ij} = \sum_{k=1}^{n} (\lambda_k \times R_{ij}^{t_k}) \Big/ \sum_{k=1}^{n} \lambda_k \tag{6}$$

where $\lambda_k = \rho^{n-k}$ is the decay factor of period t_k, and $0 < \lambda_k < \lambda_{k+1} \le 1$, $1 \le k < n$. We bring in a decay function into the equation so that the most recent ratings will carry more weight when compute the direct trust value. As a result, the direct trust value reflects most recent status of a service node and can effectively reduce the impact of on–off attack (also called strategic threat, see Section 4.1).

3.3. Reputation evaluation

In a fully distributed P2P system involving numerous nodes, it is often not possible for a peer to directly assess the trust value of another peer. Instead, a peer may need to resort to other peers in the system and rely on the collective opinions to do the evaluation. Therefore, many recommendation based trust schemes [4–9] have already been proposed. However, this also introduces new challenges, such as how to determine the accuracy of collected opinions and how to efficiently aggregate referrals from diverse recommenders with different trustworthiness.

Kamvar and Schlosser [2] improved the Gnutella protocol by adding a reputation polling phase prior to choosing a peer to download files. Zhou and Hwang [3] proposed the EigenRep scheme that uses a DHT to calculate and store the global trust value of each node based on Power iteration. In [13] authors employed an adaptive scheme to evaluate referral's reliability. This is because recommenders' reputations are different and the referrals from

peers with high reputation value are more trustworthy than those from low reputation peers. However, the referrals are treated equally and reliabilities of theirs are not considered in [8]. NICE, a scalable and efficient scheme proposed in [22], infers the reputation ranking for a particular node based on a chain of trust. [11,12] experientially assign different weights to referrals and then combine these information when calculating peer's reputation value. In order to neutralize skewed reputation reports, Jamali and Ester [10] proposed a weighted majority algorithm. Each client assigns weights to all the advisors and calculates a weighted sum of ratings provided by them. The weight of an advisor is tuned after each reputation prediction according to the quality of the information provided by that advisor. Marti and Garcia [9] proposes a trust system that collects the referrals of the first few peers joining networks.

We believe these methods may not be able to adapt well in a system that requires high precision or a system that a good number of malicious peers exist. A good P2P reputation system must effectively aggregate the overall recommendations. Moreover, it must also accurately filter out untrustworthy second opinions from any malicious peers trying to defame the reputations of some well behaved peers in the system. In this article, we present a quantified general model to address the issue. We use the term "*credibility*" to describe how much a peer can trust the recommendations from other peers (recommenders).

Let Cr_{im} denote the credibility of peer m from i's point of view. The Credibility defined in this article has two basic characteristics: dynamic and personal. It is dynamic because Cr_{im} may change over the time base on how reliable the information peer m provides to peer i. It is personal because Cr_{im} is only meaningful to peer i regarding the trustworthiness of peer m. So peer m might be well regarded as a peer with great credibility but if it has provided false referrals to peer i, then the value of Cr_{im} could be very low.

In R^{2}Trust, the credibility of a recommender is factored in to weigh its referral about other peers. Then all the referrals are aggregated to compute the reputation value a peer. The reputation value RE of peer j from the view point of i is illustrated in Eq. (7):

$$RE_{ij} = \sum_{m \in I(j)} (Cr_{im} \times DT_{mj}) \Big/ \sum_{m \in I(j)} Cr_{im} \tag{7}$$

where $I(j)$ represents the aggregate of peer j's recommenders, DT_{mj} is the direct trust value of recommender m for peer j, and Cr_{im} is the credibility of peer i for recommender m. It can be observed from Eq. (7) that the higher the credibility a recommender has the more weight its local reputation value carries. Note that the value of credibility will be updated periodically, as described in the next few paragraphs.

In R^{2}Trust, the credibility of a peer is used to weigh the feedback it reports. If a peer gives wrong feedback about other peers its credibility value is decreased and its subsequent reports will have less impact on the reputation of another peer. Similarly, if a peer's feedback is consistently good its credibility will go up. Credibility values are based on first-hand experience. Unlike ratings, they are not shared with other peers. Credibility values are also normalized so that they lie between 0 and 1.

To determine if an individual referral is consistent with the weighted average, we first define reputation differential as follows:

$$Diff_{im} = \sum_{m \in I(j)} |RE_{ij} - DT_{mj}| / |I(j)| \tag{8}$$

where RE_{ij} is the reputation of j from peer i's perspective, $I(j)$ is the set of j's referrals, and $|I(j)|$ is the cardinality of the set. Reputation differential is also known as the feedback similarity. We then further define the relative reputation difference RTD in Eq. (9).

$$RTD_{im} = Diff_{im} / STD_j \tag{9}$$

where RTD_{im} is the relative reputation difference of peer i for peer m, and STD_j is the standard deviation of all the referred local trust value on peer j. If the value of RTD_{im} is less than or equal to 1, then the referral from peer m is considered consistent with the overall referrals and peer m is therefore be rewarded by increasing the value of its credibility value. Otherwise if RTD_{im} is greater than 1, a penalty will be applied to peer m.

Typically, peers in a collusive group always give themselves good ratings while badmouthing peers outside its group. For the same node in a system, the reputation difference between the peers in a collusive group and the peers outside the group will be significant. Therefore, we can use RTD as a measure to identify peers that might be involved in collusive cheating attacks.

We then define credibility value in Eq. (10).

$$Cr_{im}^{k+1} = \begin{cases} Cr_{im}^{k} + \delta(1 - Cr_{im}^{k})(1 - RTD_{im}), \\ \quad 0 \le RTD_{im} \le 1,\ k > 0 \\ Cr_{im}^{k} - \gamma Cr_{im}^{k} \left(1 - \dfrac{1}{RTD_{im}}\right), \\ \quad RTD_{im} > 1,\ k > 0 \\ 0.5, \quad k = 0 \end{cases} \tag{10}$$

where $0 < \delta < \gamma < 1$, k is an integer, and Cr_{im}^{k} is the credibility value peer i has for peer m after the kth referral. Eq. (10) indicates, the new credibility Cr_{im}^{k+1} is the result of the most recent credibility Cr_{im}^{k} and the deviation. The change may be either an increment or a decrement, depends on the value of RTD_{im}. If $RTD_{im} < 1$, it is an increment. Otherwise it is a decrement.

In Eq. (10), the initial credibility value Cr_{im}^{0} is set to 0.5. It is believed [23] that partially trust a new peer can improve the P2P performance until the new peer is proven to be not trustworthy. Therefore, 0.5 is an appropriate initial credibility value for a new peer. After this new peer sent out more referrals, the credibility value of this peer will be updated accordingly. Eventually, the credibility value will truly reflect how credible its referrals are.

3.4. Risk value

The reputation value of a peer is measured in terms of its trustworthiness in the existing trust models. We know that reputation is an accumulative value for the past behaviors and reflects the overall evaluation of the responding peer. However, by itself it is not sensitive enough to perceive suddenly spoiled peers because it needs time to decrease the accumulative score. Risk evaluation can help to solve this problem.

Risk value is capable of capturing behaviors of malicious peers especially when they know the rules of the game and try to maintain certain amount of trust value before performing oscillating attacks. We introduce risk value to the trust computation in order to portray the unpredictable and uncertain behaviors of those malicious peers. In R^{2}Trust, the risk value is computed by applying the concept of information entropy because it has been proven to be applicable in dealing with uncertain problems.

As mentioned before, every peer has its own personalized view about the community. Therefore, to make the trust model more precise, we only use the interaction-derived information to calculate the risk value. In our model a peer locally maintains a list of the proportions of four types of files (G, C, I, M) it downloaded from its peers. Let r_q^k denote the ratio of q $(q = G, C, I, M)$ kind of transaction results over all results during the time t_k $(1 \le k \le n)$. Using the decay function, the ratio results of the total transactions during all the time can be described as

$$\left\{ \left(\sum_{\substack{k=1 \\ q=G}}^{n} \lambda_k r_q^k \right)(G), \left(\sum_{\substack{k=1 \\ q=C}}^{n} \lambda_k r_q^k \right)(C), \left(\sum_{\substack{k=1 \\ q=I}}^{n} 1/\lambda_k r_q^k \right)(I), \right.$$

$$\left(\sum_{\substack{k=1 \\ q=M}}^{n} 1/\lambda_k r_q^k \right) (M) \Bigg\}.$$

If we use p_q to denote the ratio of r_q^k to the sum of all kinds of ratio results, satisfying $0 \le p_q \le 1, \sum_{q=G,C,I,M} p_q = 1$, then $p_q (q = G, C, I, M)$ is respectively $(\sum_{\substack{k=1 \\ q=G}}^{n} \lambda_k r_q^k)(G)/S_{GCIM}$, $(\sum_{\substack{k=1 \\ q=G}}^{n} \lambda_k r_q^k)(C)/S_{GCIM}$, $(\sum_{\substack{k=1 \\ q=I}}^{n} 1/\lambda_k r_q^k)(I)/S_{GCIM}$, $(\sum_{\substack{k=1 \\ q=M}}^{n} 1/\lambda_k r_q^k)(M)/S_{GCIM}$, where $S_{GCIM} = (\sum_{\substack{k=1 \\ q=G,C}}^{n} \lambda_k r_q^k)(q) + (\sum_{\substack{k=1 \\ q=I,M}}^{n} 1/\lambda_k r_q^k)(q)$.

With the help of information entropy, we define the risk value as given below:

$$RV = f(q)H(p_q)/f(M), \quad q = I, M, \tag{11}$$

where $H(p_q) = -\sum_{q=I,M} p_q \log p_q$ is the entropy of $q (q = I, M)$. Because only I and M bring bad results, especially M produce the worst results, these transactions are taken into account when computing risk values.

Using risk evaluation, the risk-sensitive users can find the bad peers much earlier than only using the reputation value. Subsequent experiments prove that risk evaluation is essential to identify those bad peers.

4. Experiment results

In this section, we will present the results of our experiments that will show the effectiveness of our trust model. In our evaluation, we assess the performance of our scheme and compare it with PET [12] scheme and famous EigenRep [2]. We study scheme's performance under a variety of threat models. All three schemes are implemented based on QueryCycleSimulator [24,25]. There are 100 query cycles in one experiment and the results are averaged over 3 runs.

4.1. Simulation environment

The simulation is based on QueryCycleSimulator developed by P2P research group in Stanford University. In each query cycle, peer i in the network may be actively issuing a query, inactive, or even down and not responding to queries passing by. Upon issuing a query, a peer waits for incoming responses, selects a download source among those peers that responded and starts downloading the file until gets the authentic file or tries all the download sources. Then the query cycle finishes and the data is collected.

In simulation we assume that there are 1000 peers in the network. Among which, there are 100–500 malicious peers (providing inauthentic files in order to undermine the network performance) and the query message is flooded with TTL = 5. In the experiment, the chances of a peer is in the uptime stage and is uniformly distributed over [0%, 100%]. And the chance of issuing queries in the uptime is uniformly distributed over [0%, 50%]. Malicious peers are always up and always issue queries. In addition, different types of peers also vary in their behaviors when responding queries and providing files. For good peers, the probability of providing authentic files is 96%. Simple malicious peers will respond to all queries they have received and provide inauthentic files with a probability of 70%. Collusive peers provide with a probability of 100% malicious files to other peers.

The content distribution model is the same as described that in [24]. In this model each file is characterized by the content category c and the popularity rank r within this category and both follow the Zipf distribution. Files are distributed probabilistically to peers based on their popularities and the content categories that peers are interested in. Distributions used in the model are based on the measurement of real-world P2P networks. Normal peers issue queries in accordance with their interests while malicious peers issue queries randomly just to harm other peers or disturb the system. In our simulation environment, there are 10000 numbers of files and 100 content categories. Other parameters in the experiments are in Table 2.

Table 2
The parameters in the experiments.

Parameter	Value	Description
(α, β)	(0.7, 0.3) or (1, 0)	Weight ratio of trust value and risk value
ρ	0.8	Decay parameter
δ	0.4	Update factor in Eq. (10)
γ	0.8	Update factor in Eq. (10)
H_{LMT}	20	Threshold of direct transaction numbers

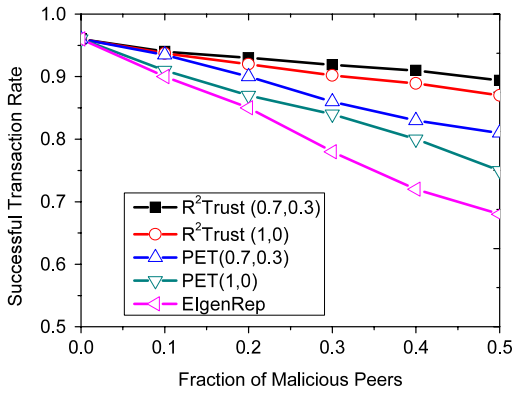

Fig. 1. STRs under SM.

As to PET, we assume that the weights of reputation and risk are 0.7 and 0.3 respectively, and the weight of direct trust is 0.5. The size of risk window is fixed to be 16. Other parameters are the same as R^2Trust. In our experiments, we consider different threat models, where a threat model describes the behaviors of a malicious peer in the network.

- Simple malicious peers. This type of malicious peers always provide an inauthentic file when selected as download source denoted as SM.
- Collusive peers. This type of malicious peers forms a malicious cycle by assigning a high trust value to other malicious peers in the network. Collusive peers provide inauthentic files to outsiders when selected as download source and provide denigrated ratings for those non-collusive peers. Collusive by assigning high trust value to each other, they hide their malicious intentions in an unstructured P2P system.
- Strategic peers. This type of malicious peers can build a good reputation and then abuse their credibility to mislead other nodes, but still allows them to maintain an acceptable reputation, also called on–off attack.

4.2. Experimental results

We compare the successful transaction rate (STR) of our scheme with PET and EigenRep under SM, Collusive and Strategic. The metrics, successful transaction rate, is the ratio of the number of successful transaction over overall transaction numbers. It is typically used to evaluate the efficiency of a trust model.

Simple malicious peers. The successful transaction rates of three models under SM are depicted in Fig. 1. When there is no malicious peers in the system, STRs of three schemes are all 96%. With the fraction of malicious peers increasing, STRs of all the three schemes decrease; the STR of our model decreases the least. Not taking into consideration under SM that malicious peers may provide

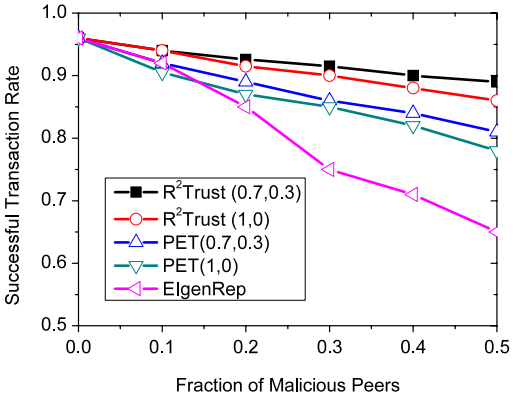

Fig. 2. STRs under Collusive.

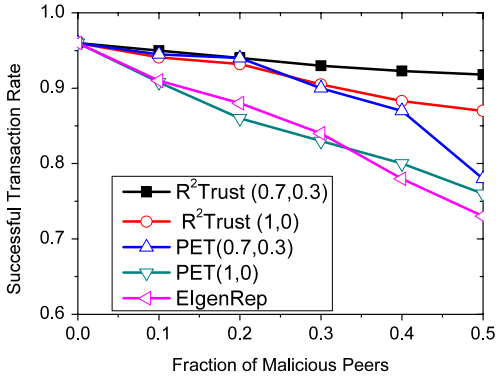

Fig. 3. STRs under strategy.

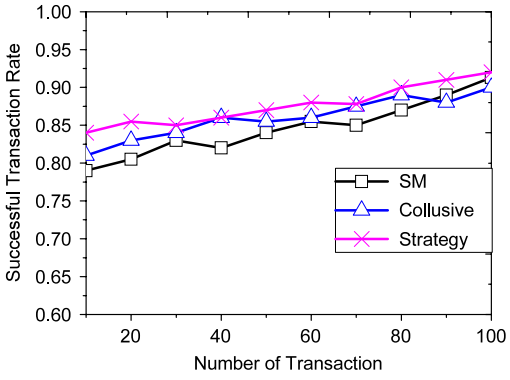

Fig. 4. STRs under number of transaction.

authentic files with certain probability, EigenRep cannot punish these peers and therefore the successful transaction rates fall more heavily. PET's STRs drop more quickly than R^{2}Trust because PET does not differentiate the transactions performed in different time periods, and the reliability of all the referrals. It does not punish the untruthful transactions and therefore the computed trust value of a peer is relatively coarse. As a result, it cannot tell whether the bad files are from bad peers or simply from good peers by errors. In comparison with PET scheme and EigenRep, the proposed R^{2}Trust remains a very high successful transaction rates: it remains 87% when the fraction of malicious peers is 50%.

Collusive peers. The successful transaction rates of three models under Collusive are shown in Fig. 2. Total number of malicious peers performing collusion was increased in steps of 10%. Each peer in the colluding group always provided inauthentic services to the consumers outward. They also boosted the trust value of their accomplices regardless of their behaviors, while downplaying the trust value of good providers. The successful transaction rates of EigenRep and PET, which do not take collusive attacks into account, descends evidently when malicious peers increase. Compared to both schemes, our trust model is designed to tackle collusive attacks, therefore it is proven robust against collusive attacks.

Strategic peers. We also compare the successful transaction rate of the system under the strategic peers with EigenRep and PET scheme. For ease of experiment implementation, we simplify the strategic peers, attack and created a representative case. In our simulation, we assume a peer with trust value less than 0.5 is untrustworthy, and a strategic peer provides true files with a probability of 20% when its trust value is beyond 0.6; and in 60% probability when its trust value is less than 0.6.

As seen in Fig. 3, the successful transaction rate of EigenRep and PET scheme are lower than that of R^{2}Trust, because both schemes cannot efficiently tackle this type of attack and cannot recognize malicious peers sensitively. However, the STR of R^{2}Trust is better than the former two trust mechanisms no matter what proportion malicious peers are. The reason is that the risk value depicts the dynamic behaviors of a peer and gives an explicit punishment to the peer whose performance drops either deliberately or unconsciously.

We compare the STRs under the scenario when the fraction of malicious peer is 25% of all the peers in the networks. In Fig. 4, STRs gradually increase when more transactions are conducted. This is because the credibility of a good peer is accumulating while the credibility of a bad peer is dropping.

4.3. *Implementation of* R^{2}Trust

The R^{2}Trust scheme can be implemented in distributed, unstructured P2P system. Typical issues in implementing R^{2}Trust in a decentralized P2P network include selection of the pre-trusted peers and decentralized trust data management, i.e., how to efficiently store and look up trust data that are needed to compute the trust value of a peer.

Like other reputation schemes, the pre-trusted peers are essential to R^{2}Trust. Generally, we have to take the extraneous factors into account to select the pre-trusted peers. For example, the founders of the P2P networks, which founded and initially used P2P networks, are commonly known to be trustworthy, and some peers that firstly join the system (usually known to be trusted by the founders). Then, the real social relationships of those peers may be used to recommend other trustworthy peers. Moreover, from a long-term viewpoint, as the P2P network evolves and grows, the P2P system can observe and identify some good peers (or malicious peers) with the help of trust mechanism.

In R^{2}Trust, a peer needs to locally store two kinds of trust information. One is the local trust information of the peers it has directly interacted with. The more transactions a system has, the more storage overhead it has. The other is the credibility information of every recommender. It needs to be updated whenever a new referral arrives. The overhead of updating credibility is related to how frequently transactions have occurred.

Each trust mechanism has its own trust data lookup scheme. Gnutella uses broadcast-based schemes and do not guarantee reliable content location. EigenTrust [2], PowerTrust [3] and PeerTrust [8], are all based on a distributed hash table (DHT) overlay network to store, search trust information. They use a DHT to deterministically map keys into points in a logical coordinate space and guarantee a definite answer to a query in a bounded number of network hops. Depending on the choice of a trust data lookup scheme, the implementation of the trust model may be somewhat different. The FPS trust data searching algorithm is presented in R^{2}Trust to guarantee reliable feedback location. FPS characterizes high search success rate and low overhead of the trust data management.

5. Conclusions

With the increasing popularity of self-organized communication systems and the mounting demand for more reliable and more secured computing environment, distributed trust and reputation management systems have received more and more attention. In this paper we present R^{2}Trust, a framework for trust management in distributed P2P networks. Different from the previous schemes, R^{2}Trust evaluates the overall trust value of a peer by its trust value and its risk value. Furthermore, we present the methods to quantifying and updating the credibility of a recommender. Our system is also application independent and can simultaneously serve an arbitrary number of P2P applications. The design of R^{2}Trust is also very resistant to malicious attacks, such as simple, collusion and behavior oscillating. The experiment results prove that the proposed R^{2}Trust performs very well even when the number of malicious peers in the system is under half. An extensive set of experiments has been performed in order to test the correctness of our model, showing how it accurately adjusts the global trust given to a domain to its real behavior, and how it quickly and effectively reacts against sudden behavioral fluctuations. The scheme continues to work well when the malicious nodes cheat in a probabilistic fashion instead of cheating all the time. From the theoretical study and experimental results, we believe R^{2}Trust can be efficiently applied to large-scale distributed P2P systems.

In the future, we would like to extend our work on R^{2}Trust along the following directions. First, we are going to investigate more threat models in P2P networks, such as intrusions, free riders and so on. Second, more work will be done to study how dynamic behaviors of peers impact the trustworthiness value and how unfair, misleading, fake ratings can be more accurately filtered out. Third, we will research what improvements can be made to enhance the robustness of R^{2}Trust to fight against malicious peers and behaviors.

Acknowledgment

This research is partially supported by the NSFC Grant 60903194, the National Research Foundation for the Doctoral Program of Ministry of Education of China under Grant 20080247 1060 and Open Project of State Key Laboratory of Networking and Switching (Beijing University of Posts and Telecommunications), Key Laboratory of Communication and Information System (Beijing Jiaotong University). The authors thank the anonymous reviewers for their suggestion on how to improve the articles. Their comments were of great help.

References

[1] Y.H. Liu, A two-hop solution to solving topology mismatch, IEEE Transactions on Parallel and Distributed Systems 19 (11) (2008) 1591–1600.
[2] S. Kamvar, M. Schlosser, The eigentrust algorithm for reputation management in P2P networks, in: Proceedings of the International Conference on WWW, Budapest, Hungary, 2003.
[3] R.F. Zhou, K. Hwang, PowerTrust: a robust and scalable reputation system for trusted peer-to-peer computing, IEEE Transactions on Parallel and Distributed Systems 18 (4) (2007).
[4] F. Cornelli, E. Damiani, D.C. Vimercati, et al. Choosing reputable servents in a P2P network, in: Proceedings of the 11th International Conference on WWW, Hawaii, USA, 2002.
[5] C.Q. Tian, S.H. Zou, W.D. Wang, S.D. Cheng, A new trust model based on adbanced D–S evidence theory for P2P networks, in: Proceedings of the 4th International Workshop on Formal Aspects in Security and Trust, in: LNCS, Ontario, Canda, 2006.
[6] C.Q. Tian, S.H. Zou, W.D. Wang, S.D. Cheng, An efficient attack-resistant trust model for P2P networks, International Journal of Computer Science and Network Security 6 (11) (2006).
[7] Y. Wang, J. Vassileva, Trust and reputation model in peer-to-peer networks, in: Proceedings of the IEEE International Conference on Peer-to-Peer Computing, P2P'03, Washington, DC, USA, 2003.
[8] L. Xiong, L. Liu, PeerTrust: supporting reputation-based trust for peer-to-peer electronic communities, IEEE Transactions on Knowledge and Data Engineering 16 (7) (2004) 845–857.
[9] S. Marti, H. Garcia, Limited reputation sharing in P2P system, in: Proceedings of the 9th ACM conference on Electronic commerce, New York, NY, USA, 2005.
[10] M. Jamali, M. Ester, Trustwalker: a random walk model for combining trust-based and item-based recommendation, in: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, USA, 2009.
[11] Y.C. Zhang, S.S. Chen, G. Yang, SFTrust: a double trust metric based trust model in unstructured P2P system, in: Proceedings of the 23rd IEEE International Parallel & Distributed Processing Symposium, IPDPS, Rome, Italy, 2009.
[12] Z.Q. Liang, W.S. Shi, PET: a personalized trust model with reputation and risk evaluation for P2P resource sharing, in: Proceedings of the 38th International Conference on System Science, Hawaii, USA, 2005.
[13] A. Ignjatovic, N. Foo, C.T. Lee, An analytic approach to reputation ranking of participants in online transactions, in: Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, California, USA, 2008.
[14] S. Buchegger, J. Boudec, Robust reputation system for P2P and mobile ad-hoc networks, in: Proceedings of the 2nd Workshop on Economics of Peer-to-Peer Systems, Berkeley, USA, 2004.
[15] H. Ma, H. Yang, M.R. Lyu, I. King, Sorec: social recommendation using probabilistic matrix factorization, in: Proceeding of the 17th ACM Conference on Information and Knowledge Management, CIKM, New York, USA, 2008.
[16] K. Walsh, E. Sirer, Experience with an object reputation system for peer-to-peer file-sharing, in: Proceeding of International Symposium on Networked Systems Design & Implementation, San Jose, USA, 2006.
[17] S.S. Song, K. Hwang, R.F. Zhou, Trusted P2P transactions with fuzzy reputation aggregation, IEEE Internet Computing (2005) 18–28.
[18] W. Sears, Z. Yu, Y. Guan, A adaptive reputation based trust framework for peer-to-peer applications, in: Proceedings of IEEE International Symposium on Network Computing and Applications, Washington, DC, USA, 2005.
[19] M. Wang, F. Tao, Y. Zhang, G. Li, An adaptive and robust reputation mechanism for P2P network, in: Proceedings of the IEEE International Conference on Communications, Cape Town, South Africa, 2010.
[20] Y.F. Wang, A. Nakao, Poisonedwater: an improved approach for accurate reputation ranking in P2P networks, Future Generation Computer System 26 (8) (2010).
[21] X. Dong, W. Yu, Y. Pan, A dynamic trust management scheme to mitigate malware proliferation in P2P network, in: Proceedings of the IEEE International Conference on Communications, Beijing, China, 2008.
[22] S. Lee, R. Sherwood, B. Bhattacharjee, Cooperative peer groups in NICE, in: Proceeding of the IEEE Infocomm, California, USA, 2003.
[23] E. Friedman, P. Resnick, The social cost of cheap pseudonyms, Journal of Economics and Management Strategy 10 (2) (2001).
[24] http://p2p.stanford.edu/www/demos.htm.
[25] M. Schlosser, T. Condie, S. Kamvar, Simulating a file-sharing P2P network, in: Proceedings of the 1st Workshop on Semantics in P2P and Grid Computing, California, USA, 2003.

Chunqi Tian was born in 1975. He received his B.S. and M.S. degrees in communication and information systems from Xi'an JiaoTong University and Xidian University, Xi'an, China, in 1998 and 2004, respectively, and the Ph.D. degree in Computer Science from Beijing University of Posts and Telecommunications (BUPT) in 2007. He is currently with Tongji University, Shanghai, China, as a Assistant Professor. His research interests include IP QoS, Peer-to-Peer and network security.

Baijian Yang is an assistant professor in the Computer Technology program at Ball State University. He became a Microsoft Certified Systems Engineer (MCSE) in 1998 and was one of the core software designers/developers for etang.com. He received his Ph.D. in Computer Science from Michigan State University in 2002. He is now engaged in research and development in the area of wireless networks and distributed systems.