SECURITY AND COMMUNICATION NETWORKS
Security Comm. Networks 2011; **4**:61–69
Published online 21 December 2009 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/sec.165

SPECIAL ISSUE PAPER

FineTrust: a fine-grained trust model for peer-to-peer networks

Yizhi Ren[1,2]*, Mingchu Li[1] and Kouichi Sakurai[2]

[1] School of Software, Dalian University of Technology, Dalian 116621, China
[2] Department of Computer Science and Communication Engineering, Kyushu University, Hakozaki, Fukuoka 812-81, Japan

ABSTRACT

Trust research is a key issue in peer-to-peer (P2P) networks. Reputation-based trust models as one of the good solutions to resolve the trust problems in P2P network are received more and more attention in recent years. One of the fundamental challenges is to capture the evolving nature of a trust relationship between peers and reflect the varied bias or preference of peers in a distributed and open environment. In this paper, we present a fine-grained trust computation model for P2P networks. Our model defines the service as a fined-grained quality-of-service (QoS) (*N*-dimensional vector), and in order to accurate the recommendation trust computing, several concepts are introduced to reflect the recommenders' current status, history behavior, and the gap between these two behaviors. Also, we firstly introduce the Gauss-bar function to measure the preference similarity between peers. All these will result in a flexible model which represents trust in a manner more close to human intuitions and satisfies the diverse QoS requirements of peers in P2P networks. The extensive simulations have confirmed the efficiency of our model. Copyright © 2009 John Wiley & Sons, Ltd.

KEYWORDS

granularity; reputation; trust model; similarity measure; P2P

***Correspondence**

Yizhi Ren, School of Software, Dalian University of Technology, Dalian 116621, China.
E-mail: ren@itslab.csce.kyushu-u.ac.jp

1. BACKGROUND AND MOTIVATION

Peer-to-peer (P2P) networks define a class of systems and applications in which autonomous peers pool their computing resources and collaborate to perform a special task. Peers have an identical functionality and play the role in both servers and clients. Unfortunately, the lack of a centralized trusted entity capable of monitoring user behavior and enforcing rules complicates the design of mechanisms for detecting and preventing malicious behavior in autonomous environments. Current research about trust model varies on the trust description, evaluation, reasoning, computing, and so on. Unfortunately, few of them so far have well resolved the following issues: (1) context-based trust: the trust relationship varies based on the different scenarios and applications; (2) social-based trust: the trust relationship is affected by the social properties of participants, etc. When facing the above two problems, the main drawback of related work is that they classify a service as either good or bad without any interim state. Such methods limit their potential in many applications [1]. For example, when peer provides a service to others, it is not easy to say that the service is good or not, because someone considers the time factor of service, someone cares about the security, etc. If only using the binary method, we are hard to differentiate the service provider and satisfy the quality-of-service (QoS) demand of various service requesters. Toward to resolving partial problems in the above two issues, Zhang and Fang [1] present a fine-grained QoS differentiation method for satisfying the diverse QoS need of individuals. For example, if QoS is defined as the time taken to finish a unit computation task, they classify the QoS as four levels: $t > 40$ (*very unsatisfied*), $30 < t \leq 40$ (*medium*), $20 < t \leq 30$ (*satisfied*), and $t \leq 20$ (*very satisfied*), while only two levels: $t \leq 40$ (*unsatisfied*) and $t > 40$ (*satisfied*) in the traditional models. Our work is a little similar to Zhang's work. However, in real world, a service may include many facets, such as time, congestion, security, and so on. It is not accurate to evaluate a service only from one facet. In this paper, we focus on the multidimensional of service and define a service as

N-dimensional vector. For example, a service is denoted as a three-dimensional vector (*time*, *security level*, and *activity*), here *time* denotes the minutes taken in the service, *security level* denotes the security of service, and *activity* denotes the involved operations or actions of service. By this, it provides a potential efficient way to measure preference of diverse individuals, and also help to design a flexible trust model to satisfy the various QoS requirements of individuals in P2P applications.

The remainder of the paper is organized as follows: Section 2 provides related works about trust model in P2P networks; Section 3 shows the overview of FineTrust; in Section 4, we will focus on the computing of recommendation trust; extensive simulation and basic analyses about this model will be given in Section 5; finally, we will make conclusions and provide future works in Section 6.

2. PREVIOUS WORKS

Trust can be analyzed from different perspectives and applications. This makes the classification of trust a difficult task. We try to classify dimensions considering the special kind of models, and the related research in the trust model can be classified into two basic directions.

2.1. Mathematical-based trust model

The simplest mathematical-based trust model is the eBay [2], where the user can give three possible values: positive (1), negative (−1), or neutral (0). The reputation value is computed as the sum of those ratings over the last 6 months. Its reputation value is computed as sum of positive ratings and negative ratings. It is too simple, and neither considers more complex information (e.g., context) nor provides explicit mechanisms to deal with malicious users. All this kind of models considers reputation as a global property and uses a single value that is not dependent on the context. Other popular methods in this direction are probability theory-based approaches. For example, Beth *et al*. [3] classifies the experience into positive type and negative type, and successful transactions will lead to positive experience increasing, while unsuccessful transactions will lead to contrary result. The direct trust is defined as the possibility of target entity successful transaction. Bayesian theory-based model use binary event (*satisfaction*, *dissatisfaction*), and trust can be expressed as expectation of probability of Beta probability density function [4,5]. Wang proposes a Bayesian network-based trust model and a method for building reputation based on recommendations in P2P networks. Bayesian networks provide a flexible method to present differentiated trust and combine different aspects of trust. Also, the logic-based method received more and more attention in recent years. For example, JØsang's subjective logic [6], Yu and Singh's Dempster–Shafer evidence theory [7], Manchala [8] use belief theory to compute trust value. In Yu's model, they use Dempster–Shafer theory of evidence as the underlying computational framework. In this case, they use Dempster's rule of combination to aggregate the information from different witnesses. This model does not combine direct information with witness information. If direct information is available, that is the only source considered to determine the trust of the target agent. Only when direct information is not available, the model appeals to witness information.

2.2. Social cognitive-based trust model

Considering the social ties and properties of peers in the network, people think that the related research should have involved more filed knowledge, such as economics, artificial intelligence, social sciences, and so on. For example, Falcone and coworkers [9,10] proposed a cognitive trust model. The basis of their model is the strong relationship between trust and delegation. They claim that trust is the mental background of delegation. In other words, the decision that takes an agent to delegate a task to agent is based on a specific set of beliefs and goals and this mental state is so-called trust. Therefore, only an agent with goals and beliefs can trust [11] takes into account the social dimension of agents and a hierarchical ontology structure [12] represent the reputation into three dimensions: individual dimension, social dimension, and the ontological dimension, and presented a reputation-based model that takes into account the social dimension of peers. Hales and coworkers [13–15] try to find what extent can social structure affecting the peers' behavior, feelings, and so on, and design a simple scenario to verify the truth behind the emerging social phenomena in P2P networks.

As a whole, recent trust research has made some progress [16–20]. The main drawback of the above researches is that they all classify a service as either good or bad without any interim state. So, how to reconcile the different notions of trust or reputation that exist in diverse fields for different scenarios [18]? How to reflect the context-dependent quantity of the trust and reputation? To address above problems, formal models for context-dependent reputation have been proposed by Wang and Vassileva [21] and Sabater and Sierra [11,22]. Context-dependent reputation systems might help mitigate cyber crimes involving self-rating on small value items among a cartel of users for gaining reputation points, which adding the capability to computational trust models to deal with several contexts has a cost in terms of complexity and adding some side effects that are not always necessary or desirable. In this paper, we propose a fine-grained trust model to provide an effective way to evaluate trust relationships among peers, and our works mainly focus on the two aspects: (1) the effect of service granularity to QoS and (2) how to find good recommenders [23]?

Security Comm. Networks 2011; **4**:61–69 © 2009 John Wiley & Sons, Ltd.
DOI: 10.1002/sec

3. FINETRUST: OVERVIEW

In P2P network, when peer *A* send a service request to another peer *B*. Then, *B* decides whether or not provides the service to *A* depending on its reputation information. If the answer is yes, then they will interact with each other, or else, they will do nothing. The above simple example is the whole process about the reputation-based trust computing in reputation system. Usually, the above process consists of two phases.

3.1. Evaluation phase

When receiving the service request, the service provider will accumulate the requester's history information (e.g., the direct interaction information with itself, recommendation information about the request from others, or both). After that, the provider using these information makes a final trust decision based on certain principles.

3.2. Interaction phase

In this phase, participants interact with each other, after that, they will evaluate the interaction based on the QoS or their bias, then preserve and update the related results in their local database, or submit the results to somewhere.

The task in the *evaluation phase* is to provide some mechanisms protecting users from potential risk interactions. If we can discover or predict the dishonest peers before the *interaction phase* happened, we can reduce the unnecessary cost of users. And the main works in our paper will focus on related issues in the *evaluation phase*.

In the P2P system, when a peer sends a service request to the potential service provider in the network, there are roughly two types of relationship between the two persons: (1) they are familiar with each other (they have no interactive history) or, (2) they are strangers. Unfortunately, the participants may freely join and leave the system, or frequently change their identities, and so on. The case (1) is rarely happened under the open environment. Even when the case (1) happened, peers may also not make an accurate decision because of the few interactions or the time decay factor.

Figure 1 shows two types of trust between *A* and *B*. In (a), because of the directive interaction between *A* and *B*, *A* can make a trust decision based on the direct information to *B* when *B* sending a service request. However, in (b), *A* and *B* are totally stranger with each other or *A* only with a few direct information about *B*, it is difficult to response to *B*'s request if no other information provided. Basically, *A* needs some recommendation about *B* from others, then make the final trust decision.

According to the descriptions in Figure 1, let DT be the *direct trust* between peers which reflecting the trust degree based on the direct interactions. Generally speaking, *direct trust* is more credible. In this paper, we compute the direct

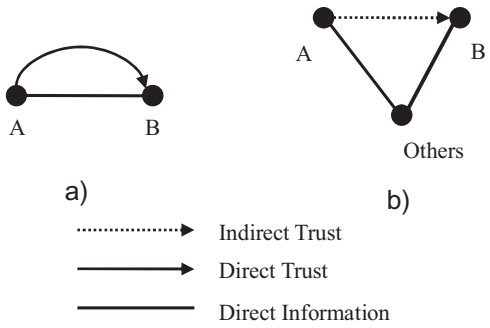

Figure 1. Two types of trust between *A* and *B*. In (a), *A* has the *direct trust* relationship with *B* when *A* has the direct information about *B*. In (2), A has the *indirect trust* (or *recommendation trust*) with *B* when *A* analyzes the recommendation about *B* from others.

trust based on satisfaction of the fine-grained QoS and the importance of the service. Let *service* be *N*-dimensional vector $(p_1, p_2, \ldots, p_N)$, here p_i denotes the *i*th property of service, and let *service satisfaction* of service be SerS $(x_1, x_2, \ldots, x_N)$, here x_i $(i \in [1,N])$ be the satisfaction value of property p_i. So, the *direct trust* between two participants *i* and *j* can be computed as following:

$$\mathrm{DT}_i^j = \frac{\sum_{s=1}^{\mathrm{INT}} \mathrm{SerS}^{(s)} \times w^{(s)}}{\mathrm{INT}} \tag{1}$$

Here $\mathrm{SerS}^{(s)}$ denotes *s*th service satisfaction, $w^{(s)}$ is weighted value of the *s*th service, and INT denotes the interaction times between *i* and *j*.

As the description in the above, it is hard for peers to make an efficient trust decision with their own knowledge and scope in an open environment (like P2P network). So, the common way is that they need the help from others. It refers to the process of trust evaluation based on the recommendation, which called as *recommendation trust* evaluation. Let RT_i^j be the *recommendation trust* between peers *i* and *j* which deducing the trust relationship from the third parties. We will give the details about the evaluation of recommendation trust in Section 4.

According to the *direct trust* and *recommendation trust*, we can easily compute the trust relationship between peers *i* and *j* as following:

$$T_i^j = \beta \mathrm{DT}_i^j + (1 - \beta)\mathrm{RT}_i^j \tag{2}$$

Here $\beta \in [0,1]$ which reflects the degree of confidence between in direct experience and in indirect experience.

As the descriptions in the above, one of the key challenges in open reputation-based system is how to compute the *recommendation trust*. We will give the further information about *recommendation trust* in the next section.

4. RECOMMENDATION INFORMATION EVALUATION

Before evaluating of recommendation information, we should consider the following two fundamental issues: (1) Which recommenders and what kind of information you want to learn? (2) With what probability you can get the information and how trustful of this information? These two issues refer to vulnerabilities when computing *recommendation trust*. One of the detrimental vulnerabilities is that a malicious peer may strategically alter its behavior in a way that benefits itself such as starting to behave maliciously after it attains a high reputation, or malicious peers submit dishonest feedback and collude with each other to boost their own reputation, or bad-mouth other peers, or malicious peers can flood numerous fake feedbacks through fake transactions in a transaction-based feedback system.

4.1. Recommendation source considering

From a sociological term, people are prone to be generous and helpful to the known ones, while always be constantly alert to strangers. In order to capture this feature, we simply classify recommenders into friends, acquaintances, strangers, and dishonest person. For each peer, it keeps three types of peer list: *Friend list*, *acquaintance list*, and *defector list*. *Friend list* is a set of the most trusted peers, *acquaintance list* is a set of peers who have the interaction history and medium trust relationship, and *defector list* is a set of malicious peers. In the real applications, peers always update their view about the world. So, it is common that a acquaintance becomes to be a friend, while friend to be a defector. Similarity, in order to refine the recommenders, peers will update their recommendation lists. For each peer, its *friend list*, *acquaintance list*, and *defector list* are always co-evolve on time. Figure 2 gives an example of the process of peer *i* updating its view toward peer *j*.

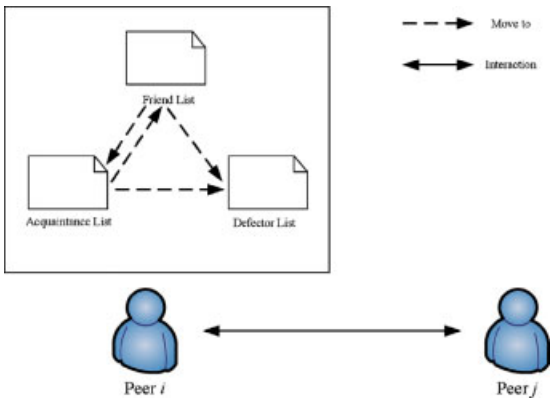

Figure 2. The peer *i* updating the view about *j*.

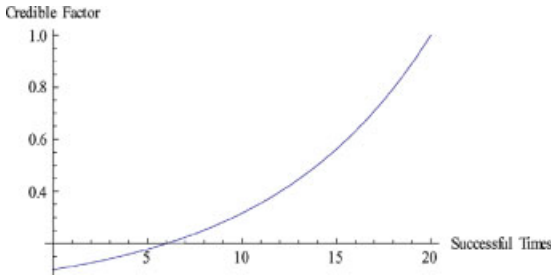

Figure 3. The dynamics of *credible factor* of peer *i* about *j*, with $\tau^{(i)} = 20$.

As in Figure 2, peers *i* will add *j* into its *acquaintance list* after the first interaction, and if they are lucky enough, they may re-interact in the future. If *j* is trustful and honest enough, it will be added into *i*'s *friend list*, otherwise, *j* is a defector to *i*.

In our paper, we only use the recommendation information from *acquaintances* or *friends*. Here, we define the *credible factor* (cf) to differentiate the creditability of recommendation from acquaintance and friend. For peer *i*, we can compute *credible factor* about *j* as follows:

$$\mathrm{cf}_i^j = \begin{cases} e^{(\tau^{(i)} - \mathrm{SINT}) \ln(0.1)/\tau^{(i)}} & \text{if } \tau^{(i)} > \mathrm{SINT} \\ 1 & \text{otherwise} \end{cases} \quad (3)$$

Here $\tau^{(i)}$ is critical value of successful interactions between friend and acquaintance, and SINT is the total number of successful interaction between peers *i* and *j*.

As in Figure 3, if SINT < 20, cf_i^j increases slowly in the initial interactions, and rapidly climb to 1 after 15 times of successful interactions. According to formula (4), cf will be constant to 1 when SINT > 20. So, when the successful time exceeds the critical value $\tau^{(i)}$, *j* will be added into *i*'s friend list ($Friends^{(i)}$). There are two reasons that we introduce the cf here: firstly, we use the $\tau^{(i)}$ to reflect the different attitudes toward the world that the bigger of $\tau^{(i)}$ is, the more passive of peer is; secondly, cf can incentive recommenders to offer the honest information in a sense.

4.2. Refining the recommendation information

In this section, we will introduce some parameters to make our model more flexible in dynamic environment. Let IT be *instantaneous trust* which reflects the current behavior and system status from the view of observer. *Instantaneous trust* is monitored by the agents of peers which easily affected by their emotion or bias acting or observing peers. So, the same transaction is observed by different agents may have varied results. In practical application, we can monitor IT by online, offline monitoring mechanisms, or combining

Security Comm. Networks 2011; **4**:61–69 © 2009 John Wiley & Sons, Ltd.
DOI: 10.1002/sec

the both mechanisms. For example, we can use offline mechanism to record systems' events as well as offline mechanism to analyze these events, and it is out of range of this paper, we will not give the details here.

Let ξ be the *recommendation error* which defined as the distance between direct trust and current trust status, and the *recommendation error* of peer i thinking about j in the time t can be computed as follows:

$$\xi(t)_i^j = \left|\mathrm{DT}_i^j - \mathrm{IT}(t)_i^j\right| \tag{4}$$

Here DT_i^j denotes the direct trust from i to j, and $\mathrm{IT}(t)_i^j$ is instantaneous trust from i to j in time t. *Recommendation error* reflects gap between peer's current behavior and long-term behavior. The bigger $\xi(t)_i^j$ is, and the more suspicious peer's behavior. Also, peer can set the recommendation error tolerance κ. If *recommendation error* exceeds κ, the recommender will be thought as dishonest. Usually, people are tend to wish punishing less familiar one in the human society, in this paper, peers can reset the threshold of recommendation error κ as follows:

$$\kappa = \kappa \frac{\gamma}{\gamma + (1 - \mathrm{cf}_i^j)} \tag{5}$$

Here cf_i^j is the *credible factor* between i and j, and γ is a constant which is used to control the degree of punishment, when κ becomes small enough, any mirror *recommendation error* will lead peers to be filtered by the system. Relatively, when recommendation error happens, we punish acquaintances more seriously than friends.

Let RA be *recommendation accuracy* which reflects recommendation precision. For peer i, the *recommendation accuracy* of peer j in time t can be computed as follows:

$$\mathrm{RA}(t)_i^j = \begin{cases} (1 - \xi(t)_i^j)\Delta_i^j & \xi(t)_i^j < \kappa \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

Here $\xi(t)_i^j$ denotes peer j's recommendation error to peer i in time t, and Δ_i^j is *preference similarity* between peers i and j. The details of Δ_i^j will be given in the next section.

4.3. Preference similarity measure

As we know, people may have various views about the same thing because of preference. For example, even recommenders provide the right information they preserved, peers may also doubt it. So, it is a natural way to introduce some methods to measure the preference distance among peers. Here, we introduce Gauss-bar function to measure the preference between peers. Let $\mathrm{Rset}^{(i)}$ ($\mathrm{Rset}^{(i)}$) be the set of peers who can provide recommendation information to peer i (j). For each peer $k \in \mathrm{Rset}^{(i)} \cap \mathrm{Rset}^{(i)}$,

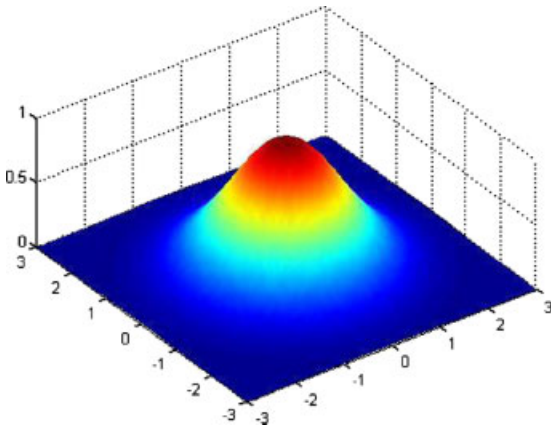

Figure 4. The preference similarity measure between peers i and j, with matrix $w=1$, $c=(0,0)$, and $\sigma=(1,1)$.

the preference similarity between i and j is

$$\Delta_i^j = \sum_{s=1}^{N} w_{ks} \exp\left[-\frac{1}{2}\left(\frac{x_{ks} - c_{ks}}{\sigma_{ks}}\right)^2\right] \tag{7}$$

Here, coefficient w_{ks} is weighed value about peer i to k, and $\sum_{s=1}^{N} w_{ks} = 1$. $c_{ks} = (c_{ks}^1, c_{ks}^2, \ldots, c_{ks}^N)$ is the kth center of peer i to k, and σ_{ks} is the kth base spreads of the blob.

As in Figure 4, apparently, when service satisfaction serS_j^k is close to the center c_{ks}, it means that the preference between i and j to k is very similar.

4.4. Recommendation trust computing

According to the above results, we can compute *recommendation trust* between peer i and j's as follows:

$$\mathrm{RT}_i^j = \frac{\sum_{k \in Friends^{(i)}} \mathrm{Info}_k^j \times \mathrm{RA}(t)_i^k + \sum_{k \in Acquaintance^{(i)}} \mathrm{Info}_k^j \times \mathrm{cf}_i^k \times \mathrm{RA}(t)_i^k}{Friends^{(i)} \cup Acquaintance^{(i)}} \tag{8}$$

Here Info_k^j is the recommendation trust degree from k about j, cf_i^k is credible factor of i to k, friend factor peer i to peer k, $Friends^{(i)}$ and $Acquaintance^{(i)}$ are friend and acquaintance list, respectively, and $\mathrm{RA}(t)_i^k$ is the recommendation accuracy that i thought about k. As in formula (8), we differentiate the recommendation from *acquaintance* and *friend* by the *credible factor* (cf).

4.5. Related algorithm

The details of related algorithm are shown in the following:

We note the following:

Algorithm 1 In each step, a pair of peers (i and j) are randomly chosen, and one as service provider (i), and the other as service recipient (j). For peer i, it performs the algorithm.

```
1. If j is stranger = FALSE
2.    Compute j's direct trust DT_i^j using formula (1);
3.       Set β = 0.8;
4. Else // no direct trust
5.       Set β = 0.2;
6. End if
7. Send recommendation requests to its Acquaintance List and Friend List;
8. If Recommendation Reply = TRUE
9.    For each replier r
10.      Analyzing instantaneous trust ( IT_i^r );
11.      Compute recommendation error ξ(t)_i^r using formula (4);
12.      End if
13.      For each peer k ∈ (Rset^(r) ∩ Rset^(i))
14.         Assign weighted value w_k to k;
15.         Assign i's view c_k to k;
16.         Get r's view x_k about k;
17.      End For
18.      Compute preference similarity Δ_i^r using formula (7);
19.      Compute recommendation accuracy RA(t)_i^r using formula (6);
20.   End For
21.   Compute indirect trust RT_i^j using formula (8);
22.   Compute T_i^j using formula (2);
23.   If T_i^j < 0.5
24.      i refuses to interact with j;
25.      RETURN;
26.   End If
27. Else
28.   i interacts with j;
29.    INT++
30.   Store j's service satisfaction SerS;
31.   If successful interaction = TRUE
32.        SINT++;
33.   Else
34.        With certain probability moving j into Defector List ;
35.   End if
36.   Update credible factor cf_i^j using formula (3);
37.   If cf_i^j = 1
38.     Add j into Friend List;
39.     Else If 0 < cf_i^j < 1
40.           Add j into Acquaintance List;
41.        End if
42.   End if
43.   Update recommendation error tolerance κ using formula (5);
44. End if
End
```

(1) If i and j are strangers, that means $\mathrm{DT}_i^j = 0$, and we set $\beta = 0.2$. Or else, $\beta = 0.8$.
(2) If i has no idea (neither direct information nor recommendation information) about j, i always provides the service to j.

5. EXPERIMENTAL EVALUATIONS

5.1. Simulation setting

We develop a simple scenario of simulation environment in C++. The related parameters used in the simulation are listed in Table I.

Table I. Description of related parameters.

Parameters	Description
M	Total number of nodes
τ	Threshold of successful times to be friend
κ	Threshold of recommendation error
γ	Degree of punishment
β	Weighted value of direct trust
N	# Dimensional of service vector
pe	Probability of adding peers into defector list when cheating happens

We use $M = 100$ nodes in the network. For each peer, we randomly generate a set of N numbers normalized by their sum as its preference center (C). In each step, we randomly generate a value between 0 and 1 as the *instantaneous trust* of the chosen service recipient.

5.2. Performance of our model

5.2.1. Impact of the number dimensional *N*.

Here, we define two metrics: *rate of successful interaction* (RSI) and *average degree of friend* (ADF). RSI is defined as fraction of a honest peers chosen in each step. If a peer is added into friend list of another peer, the *degree of friend* (DF) of that peer increases to 1. So, Average Degree of Friend is defined as the average DF of all peers in the network.

Figure 5 shows the rate of successful interactions established among peers over the simulation time. The curve with $N = 1$ shows the result of one-dimensional service vector using in References [4,5]. The curves $N = 2$ and 4 are the results employing our fine-grained QoS measure. As seen in the figure, the proposed method can help to find the potential cooperators as time passes. After 800 steps, the RSI of $N = 4$ had increased from 0.56 to 0.91.

Figure 6 shows the average degree of friends over the time steps. The metric of ADF reflects the cooperation degree of the system. If the system consists of many cooperators, it will help to increase its ADF. The curve

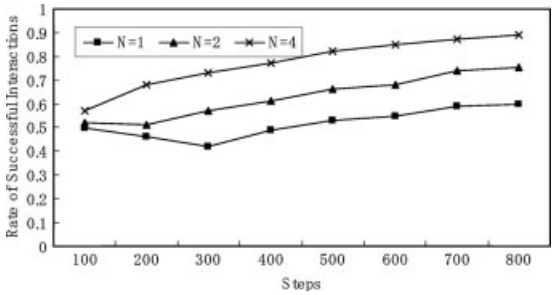

Figure 5. The rate of successful interactions, with $\tau = 5$, $\kappa = 0.5$, $\gamma = 2$, $\beta \in (0.8, 0.2)$, and $\mathrm{pe} = 0.05$.

Security Comm. Networks 2011; **4**:61–69 © 2009 John Wiley & Sons, Ltd.
DOI: 10.1002/sec

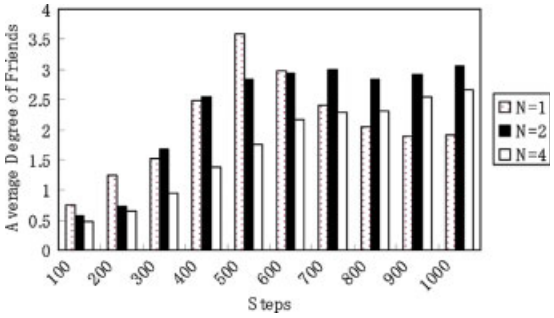

Figure 6. Average degree of friends, with $\tau = 5$, $\kappa = 0.5$, $\gamma = 2$, $\beta \in (0.8, 0.2)$, and pe = 0.05.

with $N = 1$ shows good performance in the initial time steps, it dramatically decreases after around 500 steps. The reason of the phenomena may be that all peers try to cooperate when facing totally strangers, while they keep cautious attitude after familiar with the world. Interestingly, we can see that the curve with $N = 2$ shows good result after hundreds of steps, while curve with $N = 3$ increase slowly. The reason behind that may be that the evaluation process of service will be more complex when N becomes bigger, and it is not easy of peer becoming a friend of others.

5.2.2. Impact of credible factor.

In this section, we will check the impact of the *credible factor* on the recommenders finding. We introduce two metrics: *rate of successful recommendations* (RSR) and *rate of valid recommendation* (RVR). The *rate of successful recommendations* (RSR) is defined as the fraction of recommendation from the peer whose recommendation accuracy is bigger than κ, and the *rate of valid recommendation* (RVR) defined as the fraction of recommendation from honest recommenders.

Figure 7 shows the rate of successful recommendations over the time steps. If no considering of the credible factor and with $N = 1$ [4,5], they always accept all recommenders, no filtering mechanism. However, the credible factor shows its good capacity in refining the accuracy of potential recommenders.

Figure 8 shows the rate of valid recommendations across steps. If we combine the credible factor with a fine-grained QoS measure, the proposed method shows a good performance in the validation of recommendation after 600 steps.

Also, we have checked the effect of *preference similarity* in the above simulations, we think that the *preference similarity* measure discovers the recommenders with the similar preference or bias, while *credible factor* provides a kind of filtering mechanism to reject the dishonest ones, also, it incents others good recommending if we combine it with some priorities, e.g., the download speed.

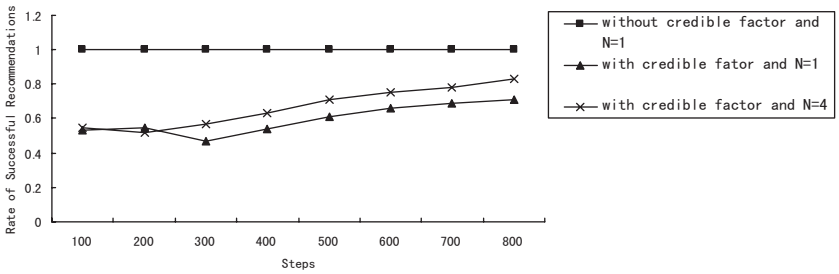

Figure 7. The rate of successful recommendations, with $\tau = 5$, $\kappa = 0.5$, $\gamma = 2$, $\beta \in (0.8, 0.2)$, and pe = 0.05.

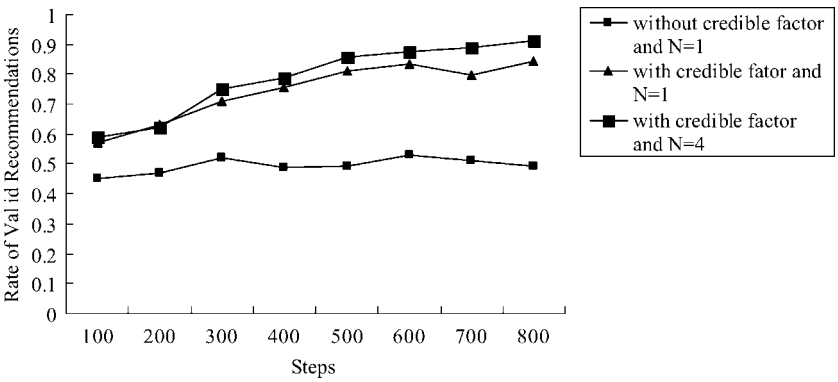

Figure 8. The rate of valid recommendations, with $\tau = 5$, $\kappa = 0.5$, $\gamma = 2$, $\beta \in (0.8, 0.2)$, and pe = 0.05.

6. CONCLUSIONS AND FUTURE WORKS

We propose a fine-grained trust computation model. The main contribution of the paper includes: (1) we define a fine-grained QoS, and use the Gauss-bar function to measure the preference similarity of peers; (2) inspired by social network, we use the *credible factor* to refine the recommendation source; and (3) also, in order to accurate the related recommendation information, we introduce some parameters to reflect the recommenders' behavior, potential recommendation risk, and so on. In the next work, we will consider the effect of time factor, e.g., the decay feature of trust relationship. Also, we will test the effect of other parameters in our simulation (e.g., ADF). In the initial experiment, we find the small world phenomena of *degree of friends* (DF). So, what the essence behind the phenomena, how is going on, and so on.

ACKNOWLEDGEMENTS

The authors like to appreciate the referees for their valuable comments and suggestions. This first author of the research is supported by the governmental scholarship from China Scholarship Council, National Nature Science Foundation of China Nos: 60673046 and 90715037, National 973 Plan under grant No.: 2007CB714205, University Doctor Subject Fund of Education Ministry of China under grant No.: 200801410028, and Natural Science Foundation Project of Chongqing, CSTC under grant No.: 2007BA2024.

REFERENCES

1. Zhang Y, Fang Y. A fine-grained reputation system for reliable service selection in peer-to-peer networks. *IEEE Transactions on Parallel and Distributed Systems* June 2007; **18**(8): 1134–1145.
2. Resnick P, Zeckhauser R, Friedman E, Kuwabara K. Reputation systems. *Communications of the ACM* 2000; **43**(12): 45–48.
3. Beth T, Malte B, Birgit K. Valuation of trust in open networks. *Proceedings of the Conference on Computer Security*. Springer-Verlag: New York, 1994; 3–18.
4. JØsang A, Ismail R. The beta reputation system. *Proceedings of the 15th Bled Electronic Commerce Conference*. June 2002.
5. Mui L, Mohtashemi M, Halberstadt A. A computational model of trust and reputation. *Proceedings of the 35th Hawaii International Conference on System Sciences*. January 2002; 2431–2439.
6. JØsang A. A logic for uncertain probabilities. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 2001; **9**(3): 279–311.
7. Yu B, Singh M. An evidential model of distributed reputation management. *Proceedings of the 1st International Joint Conference on Autonomous Agents and Multi-agent Systems*. 2002; 294–301.
8. Manchala D. E-commerce trust metrics and models. *IEEE Internet Computing* 2000; **4**(2): 36–44.
9. Castelfranchi C, Falcone R. Social trust: a cognitive approach. In Castelfranchi C, Tan Y, (eds). Trust and Deception in Virtual Societies. Boston MA: Kluwer Academic Publishers 2001; 55–90.
10. Castelfranchi C, Falcone R, Pezzulo G. Integrating trustfulness and decision using fuzzy cognitive maps. *Trust Management* 2003; LNCS 2692, 195–210.
11. Sabater J, Sierra C. Reputation and social network analysis in multi-agent systems. *First International Joint Conference on Autonomous Agents and Multi-agent Systems*. 2002.
12. Halberstadt, A, Mui, L. Group and reputation modeling in multi-agent systems. *Proceedings of the Goddard/JPL Workshop on Radical Agents Concepts*. NASA, Goddard Space Flight Center, 2001.
13. Edmonds B, Norling E, Hales D. Towards the emergence of social structure. *Computational and Mathematical Organization Theory* 2009; **15**(2): 78–94.
14. Hales D, Arteconi S, Marcozzi A, Chao I. Towards a group selection design pattern. Technical Report UBLCS-2007-25, University of Bologna, November 2007.
15. Marcozzi A, Hales D. Emergent social rationality in a peer-to-peer system. *Advances in Complex Systems (ACS)* 2008; **11**(4): 581–595.
16. Artz D, Gil Y. A survey of trust in computer science and the semantic web. *Journal of Web Semantics: Science, Services and Agents on the World Wide Web* 2007; **5**(2): P58–P71.
17. Mui L. *Computational Models of Trust and Reputation: Agents, Evolutionary Games, and Social Networks*, PhD Thesis, Massachusetts Institute of Technology, December 2002.
18. Jøsang A, Ismail R, Boyd C. A survey of trust and reputation systems for online service provision. *Decision Support Systems* 2007; **43**(2): 618–644.
19. Despotovic Z, Aberer K. P2P reputation management: probabilistic estimation vs. social networks. *Computer Networks* 2006; **50**: 285–500.
20. Aberer K, Despotovic Z, Galuba W, Kellerer W. The complex facets of reputation and trust

invited paper. *Proceedings of the 9th Fuzzy Days, International Conference on Computational Intelligence*. Dortmund, Germany, September 2006; 18–20.

21. Wang Y, Vassileva J. Trust and reputation model in peer-to-peer networks. *Proceedings of the Third IEEE International Conference on Peer-to-Peer Computing*. 2003; 150–157.
22. Sabater J, Sierra C. REGRET: a reputation model for gregarious societies. *4th Workshop on Deception, Fraud and Trust in Agent Societies*. 2001.
23. Martinovic I, Leng C, Zdarsky F, Mauthe A, Steinmetz R, Schmitt J. Self-protection in P2P networks: choosing the right neighbourhood. *Proceedings of the International Workshop on Self-Organizing Systems (IWSOS 2006)*. Passau, Germany. September 2006; 23–33.