Scalable Feedback Aggregating (SFA) Overlay for Large-Scale P2P Trust Management

Xiaoyong Li, Feng Zhou, and Xudong Yang

Abstract—In Peer-to-Peer (P2P) trust management, feedback provides an efficient and effective way to build a reputation-based trust relationship among peers. There is no doubt that the scalability of a feedback aggregating overlay is the most fundamental requirement for large-scale P2P computing. However, most previous works either paid little attention to the scalability of feedback aggregating overlay or relied on the flooding-based strategy to collect feedback, which greatly affects the system scalability. In this paper, we proposed a *scalable feedback aggregating* (SFA) overlay for large-scale P2P trust evaluation. First, the local trust rating method is defined based on the time attenuation function, which can satisfy the two dynamic properties of trust. The SFA overlay is then proposed from a scalable perspective. Not only can the SFA overlay strengthen the scalability of the feedback aggregation mechanism for large-scale P2P applications, but it can also reduce networking risk and improve system efficiency. More importantly, based on the SFA overlay, an adaptive trustworthiness computing method can be defined. This method surpasses the limitations of traditional weighting methods for trust factors, in which weights are assigned subjectively. Finally, the authors design the key techniques and security mechanism to be simple in implementation for the easy incorporation of the mechanism into the existing P2P overlay network. Through theoretical and experimental analysis, the SFA-based trust model shows remarkable enhancement in scalability for large-scale P2P computing, as well as has greater adaptability and accuracy in handling various dynamic behaviors of peers.

Index Terms—Large-scale P2P computing, trust management, feedback aggregating overlay, scalability, adaptability.

1 Introduction

P2P computing is a distributed application architecture that partitions tasks or work loads between peers. Peers make a portion of their resources, such as processing power, disk storage, or network bandwidth, directly available to other network participants, without the need for central coordination by servers or stable hosts. Peers are both suppliers and consumers of resources, in contrast to the traditional client-server model where only servers supply and clients consume. Some of the benefits of P2P computing are scalability, resource aggregation, and interoperability without any administration cost or centralized infrastructure support.

To encourage resource sharing and combat malicious peer behaviors, trust management is essential for peers to evaluate the trustworthiness of others and to interact selectively with the more trusted ones [1], [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13]. Without an efficient trust management mechanism, peers will have little incentive to contribute their computing or bandwidth resources. Peers may hesitate to interact with unknown peers due to the concern of receiving corrupted or poisoned files or being exploited by Liu and Shi [18]. To promote availability and alleviate the worries of a large number of users, we must design the P2P ecosystem to be secure, trustworthy, and dependable. Trust management is especially necessary in commercial P2P applications, such as P2P file-sharing [16], trusted content delivery [17], pay-per-transaction [19], etc.

• *The authors are with the Beijing Key Laboratory of Intelligent Telecommunications Software and Multimedia, Beijing University of Posts and Telecommunications, Beijing 100876, P.R. China. E-mail: lxyxjtu@163.com, {zfeng, xdyang}@bupt.edu.cn.*

Manuscript received 16 Feb. 2011; revised 12 Sept. 2011; accepted 28 Dec. 2011; published online 11 Jan. 2012.
Recommended for acceptance by R. Baldoni.
For information on obtaining reprints of this article, please send e-mail to: tpds@computer.org, and reference IEEECS Log Number TPDS-2011-02-0089.
Digital Object Identifier no. 10.1109/TPDS.2012.26.

1.1 Motivation

In literatures, many state-of-the-art trust models have been proposed, such as [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13]. Some of them are very creative and elaborate, but most of them still have two key limitations that need to be addressed.

1.1.1 Limited Work is Advanced in Scalable Feedback Aggregation (SFA) Overlay

In P2P trustworthiness computing, the system collects locally generated peer feedback and aggregates them to yield the global trustworthiness. For large-scale P2P networks, the key to the success of reputation-based trust management is the feedback aggregating mechanism [14]. There is no doubt that the scalability of a reputation-based trust system is the first requirement for large-scale P2P computing. That is, the trust management system should be scalable to serve for a large number of peers in terms of accuracy, convergence speed, and extra overhead per peer. However, most previous studies either paid little attention to the distribution of peer feedback or relied on flooding-based strategy to collect feedback (see Section 3), which greatly affects the system scalability and feedback availability.

1.1.2 Current Studies Lack Adaptability in Global Trust Aggregation Mechanism

In trust models, avoiding the effect of individual subjective judgment and favoritism as well as confirming the evaluation subjectively and reasonably are very important to the objectivity and unbiasedness of trust evaluation. In most current studies, two trust factors, namely, self-experience

trust (local trust) and rating trust (feedback trust) are necessary to calculate global trustworthiness [20], [28]. Such as in [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], and [13], they define global trust degree (GTD) T as follows:

$$T = W_1 \times D + W_2 \times R, \quad (1)$$

where D is the *local trust degree* (LTD), R is the *feedback trust degree* (FTD), and W_1 is the weight of D (correspondingly, W_2 is the weight of R). LTD D and FTD R can be computed through different mathematical methods proposed by these trust models. However, which values of W_1 and W_2 are reasonable? Most previous works assign theirs values using subjective or manual methods. However, these methods do not reflect the complexity and adaptability of the trust evaluation process. Furthermore, they may lead to misinformation and preclude an accurate evaluation of trustworthiness.

1.2 Main Ideas and Contributions

In the scalable feedback aggregating overlay proposed in the current paper, if the interaction of two peers' occurs, the two peers will become a pair of buddies. Thus, once a peer joins the P2P system and obtains service from others, he/she must become a buddy of the peers who provides him/her with service. Thus, we consider each peer in the SFA overlay to hold some buddies for voting. Based on this understanding, the main idea of the SFA overlay is to restrict the polling of peers' advice to a limited number of "buddies," achieving faster convergence and lower storage complexity with regard to traditional "open poll" approaches. The SFA overlay regards scalability as the first requirement of a trust system; i.e., our trust mechanism is scalable to serve a large number of peers in terms of convergence speed, extra overhead per peer, and so on.

Moreover, the proposed trust evaluation is based on two-methods: LTD and FTD. The value of LTD is evaluated based on direct experiences between two peers. In the human cognitive process, people are more willing to believe their own direct experiences. Based on this understanding, we suggest an adaptive trust evaluation method (known as *self-feedback mechanism*) to integrate peers' direct experience into the overall trust evaluation. The mechanism can give the self-experience trust more weight. Thus, in all kinds of trust sources, the weight for the self-experience trust is the largest one, which is in line with the cognitive pattern of human beings on trust. Additionally, using this trust evaluation mechanism, our trust system can partly mend the accuracy of the feedbacks evaluation with a small number of buddies.

Overall, the main innovations of this paper go beyond existing approaches in the following ways.

1.2.1 A Scalable Feedback Aggregating Overlay

Based on a scalable perspective, we present the SFA overlay. Feedback is searched using the SFA instead of the polling-based methods used in previous works. Not only can this strengthen the scalability of feedback aggregation mechanism for large-scale P2P applications, but it can also reduce risk and improve system efficiency.

1.2.2 An Innovative Local Trust Rating Method Based on Time Attenuation Function

The dynamic property of trust creates the greatest challenge in measuring trustworthiness. In this paper, we propose an innovative computing method for LTD based on the time attenuation function, which can satisfy two social properties of trust: the dynamics and time-based attenuation.

1.2.3 An Adaptive Weight Allocation Method for GTD Calculation

Many previous studies have used subjective means to assign various weights for trust factors. The adaptability of these models has limitations. Based on the human cognitive process, we use a novel self-feedback mechanism to integrate peers' local trust scores into the overall trust evaluation. This mechanism can overcome the effect of the rigidity of assigned weights on the overall trust perception and evaluation.

1.2.4 A Risk-Probabilistic-Based Method to Combat Raters' Misbehavior

In P2P networks, some raters may be misbehaving peers (e.g., colluding cheaters). They may exaggerate the positive or negative ratings, or offer testimonies that are outright false. Focusing on this issue and based on the risk-probabilistic model in economics, we propose a feasible method to combat raters' misbehavior or at least make raters' misbehavior costly.

1.2.5 A Pretty Good Privacy (PGP)-Based Signature Mechanism to Support Identity-related Issues

Identity support is probably the most crucial element in a security and trusted service system. However, P2P systems lack infrastructure services to support public key cryptographic mechanisms that rely on a trusted CA. We adopt a PGP-based signature mechanism to support identity-related issues in a fully self-organized manner, which can better meet the identity-related issues in the SFA overlay.

We also design the key techniques to be simple in implementation over pure P2P networks, so that the mechanism can be incorporated into the existing P2P overlay network. The remaining parts of this paper are organized as follows. Section 2 gives an overview of the related works. Section 3 analyzes the existing discovery strategy for feedback and Section 4 outlines the details of the SFA overlay. Section 5 discusses the overlay implementation principle over pure P2P networks. The security mechanisms are presented in Section 6 to combat malicious peers. The simulation results are presented in Section 7. Section 8 concludes the paper and suggests future directions for improvement.

2 Related Work

Damiani et al. proposed a reputation-based approach for P2P file sharing systems (called P2PRep) [3], [4], [6], [7]. In P2PRep, a peer polls other peers by broadcasting a request for the opinion of the selected peer. The behavior of the proposed system is described through a comparison with probabilistic approaches. Damiani et al. presented a similar approach called XRep, which considers the reputations of

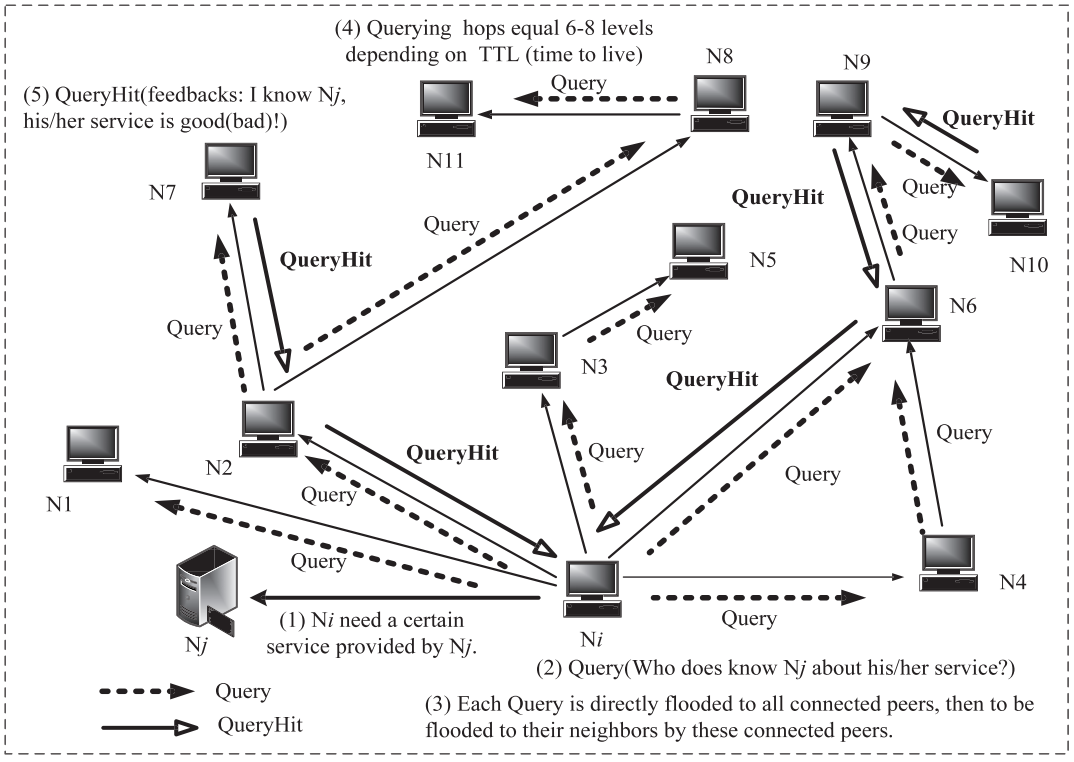


Fig. 1. Example of a flooding-based searching algorithm for feedback relaying via the unstructured P2P overlay.

both peers and resources [5]. P2PRep and XRep do not give any metrics to quantify the credibility of the voters. Moreover, they only can find trustworthy peers within a given horizon. Our approach involves an adaptive process of trusted neighbor selection, which may help detect malicious peers originally beyond the horizon.

Xiong and Liu developed the Peer-Trust model based on a weighted sum of five peer feedback factors [8]. Peer-Trust is fully distributed, uses overlay for trust propagation as well as public-key infrastructure for securing remote scores, and protects peers from some malicious abuses. The limitation of this approach is that the scalability and the convergence in large-scale P2P systems is not considered. The five factors used in their trust model must be retrieved with a heavy overhead.

In [9] and [10], Shi and Liang proposed the Trust-Ware system, a trusted middle-ware for P2P Web server sharing. Their approach consists of two models: the *Multiple Currency-based Economic* model (M-CUBE) and the *Personalized Trust model* (PET). The M-CUBE model provides a general and flexible substrate to support high-level P2P resource management services. The PET model derives peer trustworthiness from long-term reputation evaluation and short-term risk evaluation. However, this method does not take into account the weight assigning problem for the trust evaluation factors, which lacks adaptability in global trust calculation.

In the University of Southern California, Zhou and Hwang developed a robust and scalable P2P reputation system, Power-Trust [14], to leverage the power-law feedback characteristics. The Power-Trust system dynamically selects a small number of the most reputable power nodes using a distributed ranking mechanism. By using a look-ahead random walk strategy and lever-aging the power nodes, it significantly improves global reputation accuracy and aggregation speed. Power-Trust is adaptable to dynamics in peer joining and leaving and is robust to disturbance by malicious peers. However, their method is fully based on reputation strategies to evaluation the global trustworthiness and without considering the local-trust factor in overall trust evaluation.

3 Existing Discovery Strategy for Feedback

Flooding (or polling) algorithm is the key technology for feedback discovery over unstructured P2P systems based on Gnutella [23]. Fig. 1 shows an example of this algorithm, where N_i needs to discovery feedback from others to generate a global trustworthiness for N_j. To search the feedback, N_i broadcasts a Query message to each node to which it is directly linked. As the message is broadcasted through the P2P overlay network, and each node is not directly connected with N_i, other nodes have to forward this message via intermediaries. Feedback raters (FR), N_7 and N_{10}, receive the Query and have in their repository feedback requested, and answer with a QueryHit unicast packet containing a Result Set plus their IP address and feedback. When N_i receives these feedback, it will generate a global reputation value for N_j according to the related evaluation algorithms.

Query messages are broadcasted to each neighbor in the network until TTL equals zero. Thus, each message passes n peers such as (2) for a given TTL

$$n = \sum_{i=1}^{t} U_i (U_i - 1)^{i-1}, \tag{2}$$

where U is the number of connections to peer i, and t is TTL value. Based on (2), for a large-scale P2P network, this method of searching feedback produces too much search

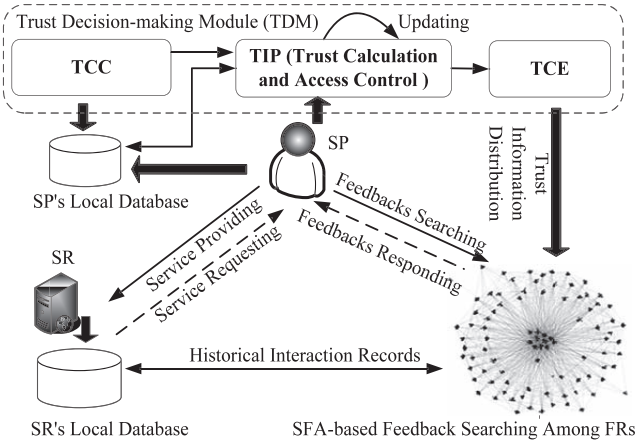


Fig. 2. Global trust service architecture.

traffic and consumes too much time and storage cost. For a reputation system, aggregating speed and cost are fundamental demands for scalability. Clearly, the flooding-based search is unsuitable for large-scale P2P applications. Therefore, we need more efficient ideas to solve this problem.

4 Construction of the SFA Overlay

4.1 Global Trust Service Architecture

Using $\Omega = \{N_1, N_2, \ldots, N_n\}$ to denote the interacting peers in the P2P system. Based on the roles in the P2P system, peers can be divided into three types: *service provider* (SP), *service requester* (SR), and *feedback rater*. Whenever a peer N_i, has to decide whether to trust another peer N_j, trust information (feedback) about N_j has to be collected. The fundamental role of our trust system is to monitor the behaviors of other entities and collect, aggregate, and distribute feedback.

The architecture of the trust management system we have developed is depicted in Fig. 2. *Trust decision-making module* (TDM) is deployed at each peer by a special software agent. It comprises three core components: *trust context collector* (TCC), *trust information processor* (TIP), and *trust context emitter* (TCE). TCC is used to gather feedback from FRs. TIP uses the related algorithms to calculate an aggregated representation of a peer's trustworthiness. TCE makes the results available to other requesting peers.

In Fig. 2, two pieces of basic information, self-experience trust (local trust) and rating trust (feedback trust), are necessary to calculate global trustworthiness. Local trust represent the peer's history; the trust decision-making framework keeps it in the SP's local database. Feedback comes from other peers' recommendations, and TDM propagates feedback by TCE. By making the trust scores publicly available, peers are able to make informed decisions about which peers to trust.

Definition 1. *Using $T(N_i, N_j)$ to denote GTD, we use the following equation to calculate $T(N_i, N_j)$:*

$$T(N_i, N_j) = W_1 \times D_L(N_i, N_j) + W_2 \times R_U(N_i, N_j), \quad (3)$$

where N_i is an SP, and N_j is an SR. $D_L(N_i, N_j)$ represent the value of LTD at time-stamp L. $R_U(N_i, N_j)$ is FTD and U is the set of FRs toward peer N_j. W_1 and W_2 are the weights of LTD and FTD, respectively. In most previous work, two kinds of subjective methods were assigned the values to W_1 and W_2 [22], namely, expert opinion, and average weight approach. However, each of the two methods lacks dynamic adaptability. Once the value of each weight is identified, dynamically adjusting the value is difficult. Therefore, assigning adaptive values to W_1 and W_2 is one of the key tasks in this work.

4.2 LTD Computing Based on Attenuation Function

LTD is given by the knowledge of the peer's nature or past interactions in the physical world, without requesting information from a *trusted third party* (TTP). LTD is generated every time when an interaction takes place. it has two fundamental properties [21].

Property 1. *The dynamic of LTD. The value of LTD changes over time due to newer interactions.*

Trust is one of the most fuzzy, dynamic and complex concepts in both social and business relationships. The dynamic of LTD creates the greatest challenge in evaluation trustworthiness [20]. To develop an LTD evaluation method, we first need to consider the dynamic of LTD.

Property 2. *The time-based attenuation of LTD. The value of LTD decreases with the passage of time.*

Property 2 shows that when no experience with an SR is made in a long time, the old trust relationship may no longer be valid. This usually means that the trustworthiness decreases over time.

The main design goal for LTD is to meet the requirements of Properties 1 and 2. Supposing that N_i has rated the quality of service of the latest h interactions with N_j as a time-based series probabilistic ratings

$$S(N_i, N_j) = \left\{\tau^{(1)}_{N_i,N_j}, \tau^{(2)}_{N_i,N_j}, \ldots, \tau^{(m)}_{N_i,N_j}, \ldots, \tau^{(L)}_{N_i,N_j}\right\}, \quad (4)$$

where $\tau^{(m)}_{N_i,N_j} \in [0, 1]$, and L is known as the history evidence window, which refers to the largest number of history records considered by the trust model. After an interaction (called an experience), N_i will give a score for N_j in accordance to its performance. For the sake of risk reduction, the quantization function of $\tau^{(m)}_{N_i,N_j}$ is defined as the following risk probabilistic model:

$$\tau^{(m)}_{N_i,N_j} = \frac{\sum \alpha^+(m)}{\sum \alpha^+(m) + \sum \alpha^-(m)}, \quad (5)$$

where $\sum \alpha^+(m)$ is the total number of positive behaviors during the past m interactions, and $\sum \alpha^-(m)$ is the total number of negative behaviors. Using (5), risk is implicitly considered within the trust evaluation. Thus, from the past behaviors, we can calculate the trust probabilities. Examples are listed in Table 1.

Definition 2. *Based on the risk probabilistic model (5), LTD of peer N_i on N_j can be computed as follows:*

$$D_L(N_i, N_j) = \frac{\sum_{m=1}^{L} \left(\tau^{(m)}_{N_i,N_j} * \gamma(m)\right)}{\sum_{m=1}^{L} \gamma(m)}, \quad (6)$$

where $\gamma(m) \in [0, 1]$ ((7)) is the time-based attenuation function and determines the weights given to the past

TABLE 1
Trust Probabilities According to the Peer Behaviors

Time-stamp	$\sum \alpha^+(m)$	$\sum \alpha^-(m)$	$\tau^{(m)}$
1	5	0	1.00000
2	5	1	0.83333
3	6	2	0.75000
4	11	3	0.78600
5	11	8	0.57900

observations. According to Property 2, old knowledge has less infection, whereas new knowledge has more contribution to trust decision-making. Thus, we define the computing expression of the time-based attenuation function $\gamma(m)$ as the following recursive expression:

$$\begin{cases} \gamma(m) = 1, & m = L, \\ \gamma(m-1) = \gamma(m) - (1-\mu)^L, & m < L, \end{cases} \quad (7)$$

where $\mu \in (0,1)$ is an adjustable positive constant in the system and can be tuned accordingly. Attenuation function $\gamma(m)$ reflects the time-based attenuation of LTD. Notice that instead of the above function, we could have used any other function that has the property of slowly approaching "0" with the decrease in argument.

Fig. 3 shows the curve of attenuation function $\gamma(m)$ with different adjustable constant μ, where $L \in [1,5]$. The value of μ represents the degree of attenuation, and a lower μ discounts older observations faster. According to Property 2, LTD should be more sensitive to new evidence and experience a long time ago should have less contribution on trust decision-making. Considering the results reported in Fig. 3, we suggest that the value of μ should be configured between 0.5 and 0.2.

From (5) to (7), our computing model on LTD (trustworthiness) develops over time. Based on experience, only positive experiences can lead to an increase in trustworthiness. Thus, two dynamic properties of LTD have been perfectly satisfied. As an example, supposing $L = 5$ and $\mu = 0.4$. Peer N_0 rates the quality of service of the latest L interactions with N_1 as a series: $S(N_0, N_1) = \{0.6, 0.7, 0.6, 0.8, 0.75\}$. According to (7), $\gamma(5) = 1$, $\gamma(4) = \gamma(5) - (1-0.6)^5 = 0.922$, $\gamma(3) = 0.84$, $\gamma(2) = 0.777$ and $\gamma(1) = 0.69$. Finally, using (6), we can obtain $D_5(N_0, N_1) = \sum_{m=1}^{5}(\tau_{N_0,N_1}^{(m)} * \gamma(m)) / \sum_{m=1}^{5}\gamma(m) = 0.696$.

4.3 Construction of the SFA Overlay

In the SFA overlay, each peer has a set of buddies, a subset of which is identified as its directly connected neighbors. A peer's trustworthy buddies are those acquaintances who have direct interactions with the peer. A peer can maintain a data item in its local database for each trustworthy buddy. To construct the SFA overlay on top of purely P2P overlay networks, we apply a dynamic data table (see examples in Section 5) in each peer's local database. Thus, each peer needs to maintain a buddy table. If peer N_j has direct interactions with N_i, their interactive information (including LTD $D_L(N_i, N_j)$) will be recorded in N_i's local database. N_j then becomes a buddy of N_i. As previously mentioned, the main contribution of the SFA overlay is to restrict the polling of peers' advice to a limited number of "buddies,"

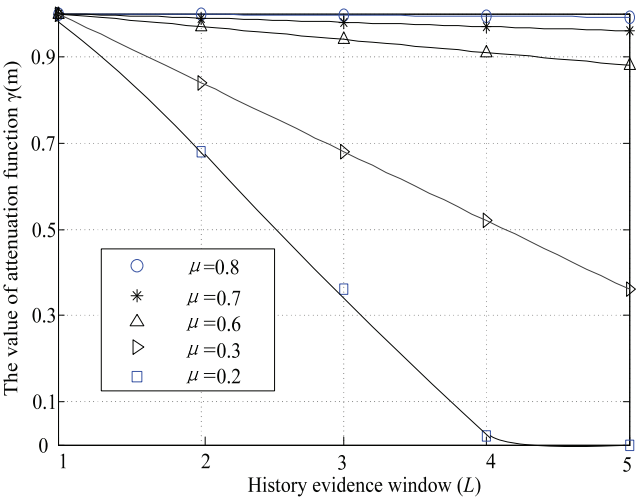


Fig. 3. Value of $\gamma(m)$ with different μ.

achieving faster convergence and lower storage complexity with regard to traditional "open poll" approaches.

Definition 3. *A service peer N_i is used as the root (source node), and the N_i's trustworthy buddies are used as the successor nodes, which also have their own trustworthy buddies (successor nodes) and so on. Thus, any peer can be connected to its trustworthy buddies, and we can obtain a multilevel tree-based network topology. We use LTD ($D_L(N_i, N_j)$) to weight the linked edges between N_i and its buddy N_j. Thus, we can construct a weighted directed tree called SFA overlay. We use a 3-tuple to represent the SFA framework*

$$sfa\_overlay(N_i) = (V_n, G, A), \quad (8)$$

where $V_n = \{N_1, N_2, \ldots, N_n\}$ is the set of peers in the SFA overlay, and n is the total number of peers. $G = \{g_1, g_2, \ldots, g_k\}$ shows the direct trust relationship between father nodes and their trustworthy buddies. $\forall g_i \in G$ and $g_i = <N_i, N_j>$, indicating that N_i and N_j are the start-point and end-point of the directed edge g_i, respectively. In the logic sketch map, we use the line with an arrow to show g_i. $A = \{a_{N_i,N_j}\}$ is the weight vector for the directed edge $<N_i, N_j>$; its value is equal to the LTD between the two peers (N_i, N_j). Thus, a_{N_i,N_j} can be defined as the following matrix:

$$a_{N_i,N_j} = \begin{cases} D_L(N_i, N_j), & <N_i, N_j> \in G, \\ 0, & others. \end{cases} \quad (9)$$

A trusted path over $sfa\_overlay(N_0)$ consists of a sequence of vertices $<N_0, N_1, N_2, \ldots, N_h>$, such that $<N_x, N_{x+1}> \in G$ (or $\{N_x, N_{x+1}\} \in G$) for each x from 0 to $h-1$ is a trusted buddy in $sfa\_overlay(N_0)$. The weight of a trusted edge over the overlay is often considered for its length. For any internal node, N_x, has an associated length $level(N_x)$, which is the sum of the edge lengths on the hops from root to N_x. We stipulate that $level(N_i) = 0$. Thus, for the root's trustworthy buddies, such as N_2 in the example in Fig. 4, its $level(N_2) = 1$. For trustworthy buddies in the second layer, such as N_9 in Fig. 4, its $level(N_9) = 2$, and so on.

Fig. 4 shows an example of the SFA-based searching process for feedback. The solid lines with solid arrows refer to the edges of the overlay involved in forwarding Query

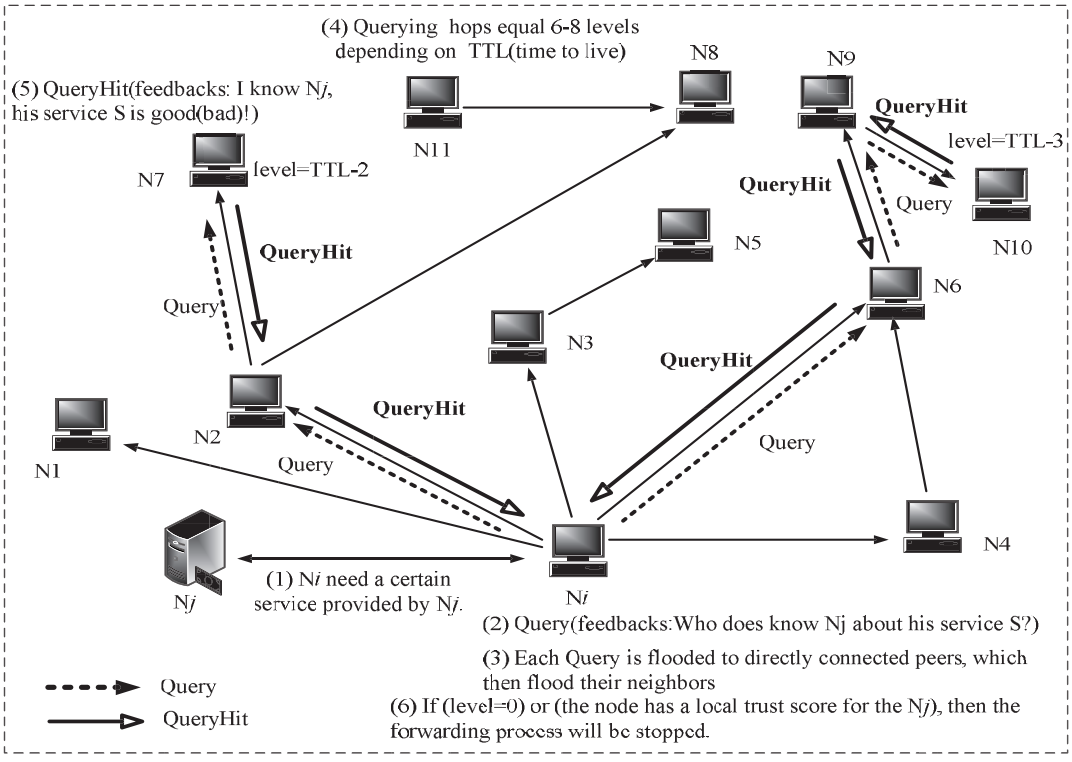


Fig. 4. Example of the SRA-based searching overlay for feedback relaying via the unstructured P2P overlay network.

messages. Different from the flooding strategy, N_i only sends the Query message to its two trusted buddies, N_2 and N_6. Other peers use the same method as N_i, and only send the Query messages to their trusted buddies. Some FRs on the SFA overlay receive the Query message, and they are willing to offer the requested feedback to N_i. They will send back a QueryHit message containing the feedback for N_j. In Fig. 4, there are two trusted paths: $<N_i, N_6, N_9, N_{10}>$ and $<N_i, N_2, N_7>$. Clearly, in this way, the total number of participating nodes and edges is significantly reduced.

4.4 Key Properties Analysis of the SFA Overlay

If N_i trusts N_j, and N_j trusts N_k under a particular context, then this chain relationship creates a new trust relationship between N_i and N_k. However, the trustworthiness between N_i and N_k is not equal to the trustworthiness between N_i and N_j; rather, it is attenuated. Thus, the partial transitivity is one of the most fundamental properties of a feedback-based trust relationship. Following, we will introduce two essential properties to explain the feasibility of the SFA overlay.

Property 3. *The partial transitivity of the SFA overlay. Supposing that N_i trusts N_j, and N_j trusts N_k, then N_i can partly trust N_k through N_js recommendation.*

The partial transitivity of the SFA overlay is a very important concept in the large-scale P2P network environment where peers often want to identify the quality of service through a transitive introduction. However, assuming that the partial transitivity is an explicit phenomenon in trust is illogical. It must be clarified that derived trust and the trust from which it is derived should be considered within the same context. Understanding that this derived trust may be explicit is important, but generally, quantifying it accurately is very difficult. In Section 4.5 (11), a novel LTD-based weighted mechanism is used to incorporate the partial transitivity of the SFA overlay into the GTD evaluation.

Property 4. *The distance-based attenuation of the SFA overlay. If N_i collects feedback about N_j from other peers in the SFA overlay, the feedback collected from peers closer to N_i should be counted with more weight compared with the values collected from distant peers.*

The distance-based attenuation can ensure the fast convergence of trust computing process over the SFA overlay. Let us use an universal example to illustrate this property. In Fig. 5, N_i is the root node, and $\{N_j, N_k, N_m \in \Omega\}$ are N_is successor nodes (children nodes). According to the definition of the SFA overlay, $level(N_j) = 1$, $level(N_k) = 2$ and $level(N_m) = 3$. According to the human habits, trust degree in the first hand is more trustworthy than that of the second hand. Thus, based on the trust path $<N_i, N_j, N_k, N_m>$, we can get the following trust relationships: $R(N_i, N_j) \geq R(N_i, N_k) \geq R(N_i, N_m)$, indicating that the SFA overlay meets the distance-based attenuation.

Hence, the partial transitivity and distance-based attenuation should be the basic properties of the SFA overlay. The computing model of FTD should meet Properties 3 and 4. In

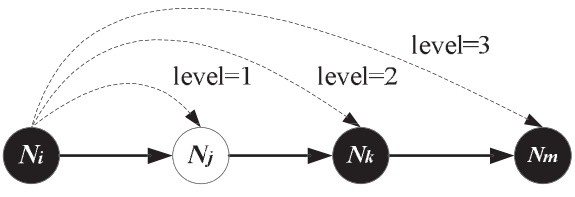


Fig. 5. A trusted path in N_is SFA-based overlay.

the following, we will discuss the global reputation aggregation based on Properties 3 and 4.

4.5 FTD Aggregation and GTD Calculation

Liang and Shi [28] proved that the simple averaging algorithm is good enough considering the simplicity of the algorithm design and the low cost in the system running. Thus, we define FTD as the follows.

Definition 4. *Supposing that $U = \{U_1, U_2, \ldots, U_m\}$ are a group of FRs toward peer N_j, and the testimony $D_L(U_k, N_j)$ is U_ks LTD for peer N_j. Then, the value of FTD can be evaluated by the following equation:*

$$R_U(N_i, N_j) = \frac{\sum_{k=1}^{m}(\varpi(U_k) * D_L(U_k, N_j))}{\sum_{k=1}^{m}\varpi(U_k)}, \tag{10}$$

where $\varpi(U_k)$ is the weight for the feedback provided by an FR U_k. According to Properties 3 and 4, the value of $\varpi(U_k)$ should reflect the attenuation and partial transitivity of the SFA overlay. If $level(U_k)$ is large, implying that U_k is far from the root of the SFA overlay and the feedback from U_k has less influence, then U_ks feedback should be given a little weight. Supposing that N_i is the root of the SFA overlay, and U_ks trusted path from N_i is $<N_i, N_{i+1}, \ldots, N_{i+level(U_k)-1}, U_k>$. Then, the computing function of $\varpi(U_k)$ is defined as follows:

$$\varpi(U_k) = \begin{cases} \prod_{r=0}^{level(U_k)} D_L(N_{i+r}, N_{i+r+1}), & level(U_k) > 0 \\ 1, & level(U_k) = 0 \end{cases} \tag{11}$$

In (11), if $level(U_k) = 0$ and $\varpi(U_k) = 1$, which means the FR is the service provider itself. From the view of the SFA overlay, when $\varpi(U_k) = 1$, which implies a self-trust relationship of the root node (**Note:** in the SFA overlay, the self-trust relationship also called *self-feedback*). SPs' *self-feedback* can be used to integrate peers' LTD into the overall trust evaluation.

As previously mentioned, avoiding the effect of individual subjective judgment and favoritism as well as confirming the evaluation subjectively and reasonably are very important to the objectivity and unbiasedness of trust evaluation. However, most previous works have used expert opinion in weighing trust sources, which lacks adaptability. So, avoiding the effect of individual favoritism on the weight allocation of trust sources is a key task of this paper. In the human cognitive process, people is more willing to believe his own experience. Based on this understanding, an adaptive weights allocation mechanism (*self-feedback mechanism*) is defined to calculate the values for W_1 and W_2:

$$W_1 = \frac{1}{1 + (\sum_{k=1}^{m}\varpi(U_k))/m}, W_2 = \frac{(\sum_{k=1}^{m}\varpi(U_k))/m}{1 + (\sum_{k=1}^{m}\varpi(U_k))/m}. \tag{12}$$

Obviously, in (12), the value of $(\sum_{k=1}^{m}\varpi(U_k))/m$ is always lesser than or equal to the value of self-feedback (its value is defined as 1 by (11)), thus W_1 is always greater than or equal to W_2. According to this reasoning, our trust system can give LTD more weight. Thus, in all kinds of trust sources, the weight for LTD is the largest one, which is in line with the cognitive pattern of human beings on trust.

Finally, GTD in the proposed trust system is calculated by the following equation:

$$\begin{aligned} T(N_i, N_j) &= W_1 * D_L(N_i, N_j) + W_2 * R_U(N_i, N_j) \\ &= \frac{D_L(N_i, N_j)}{1 + (\sum_{k=1}^{m}\varpi(U_k))/m} \\ &\quad + \frac{(\sum_{k=1}^{m}\varpi(U_k)/m) * R_U(N_i, N_j)}{1 + (\sum_{k=1}^{m}\varpi(U_k))/m}. \end{aligned} \tag{13}$$

5 Implementation in Pure P2P Networks

5.1 Implementation Mechanism

Our overlay can be constructed in a Gnutella-based P2P network, with an architecture similar to that of P2Prep [4], which is a pure P2P network for file exchange, and, more precisely, in the Gnutella architecture. The reason for focusing on a pure P2P network is that it is closest to the ideal structure of the peer-to-peer (P2P) spirit, where all participants have a uniform role.

In the SFA overlay, each peer should hold a set of buddies, a subset of which is identified as its neighbors. In order to construct the SFA overlay on top of purely unstructured P2P networks, a data table (called *buddy table*) is applied in every peer's local database. Each peer (as an SP) maintains a buddy table (see examples in Fig. 6). If another peer (as an SR) has direct interactions with it, then the LTD of the peer (SR) will be recorded in this table and the peer (SR) becomes a buddy of the SP. To manage online status of a peer, an item is added in each peer's buddy table. This item is called "*online mark*," when a peer (SR) joins the SFA overlay, the SP will set the SR's "*online mark*" as "*yes*." While a peer (SR) leaves the SFA overlay, the SP will set the SR's "*online mark*" as "*no*." Thus, the main items of the buddy table include the buddy's *ID*, *LTD*, *online mark* and *other items* (such as GTD, the total number of interactions, etc).

As shown in Fig. 6, N_i has four neighbors, but only N_2 and N_6 are its trustworthy buddies. According to a peer's buddy table, a high LTD score implies that this buddy can provide high quality service and trustworthy feedback to its father node. In Fig. 6, N_2 and N_6 are N_is trustworthy buddies; i.e., N_2 and N_6 have a direct interaction history with N_i. When N_j requests a certain service from N_i, N_i needs to collect feedback from the SFA overlay and aggregate these feedback into an FTD value for N_j. Thus, N_i only sends Query messages to N_2 and N_6 rather than to its four neighbors N_1, N_4, N_2, and N_6. Other peers work the same as N_j. As shown in Fig. 6, if $level = 1$, then $\varpi(N_2) = 0.7$ and $\varpi(N_7) = 0.7$ implying the first-level weight for the successors. Whereas if $level = 2$, then $\varpi(N_9) = 0.7 \times 0.5 = 0.35$. As for $level = 3$, it implies the third-level buddies. For N_{10}, its weight is $\varpi(N_{10}) = 0.8 \times 0.7 \times 0.5 = 0.28$. In the example presented here, supposing that LTD $D_L(N_7, N_j) = 0.7$, $D_L(N_{10}, N_j) = 0.8$ and N_is LTD on N_j is $D_L(N_i, N_j) = 0.6$, then GTD $T(N_i, N_j) = 0.6552$.

5.2 Convergence Control of the Feedback Discovery

According to Property 4 and (11), with the increasing of $level(U_k)$, $\varpi(U_k)$ quickly approaches zero. To acquire fast *convergence* in feedback discovering, we only need to search a lesser number of hops (levels) in the SFA overlay. Based on this understanding on the SFA overlay, we adopt two new parameters, *quality-factor* and *distance-factor*, to control automatically the peer scale involved in the feedback

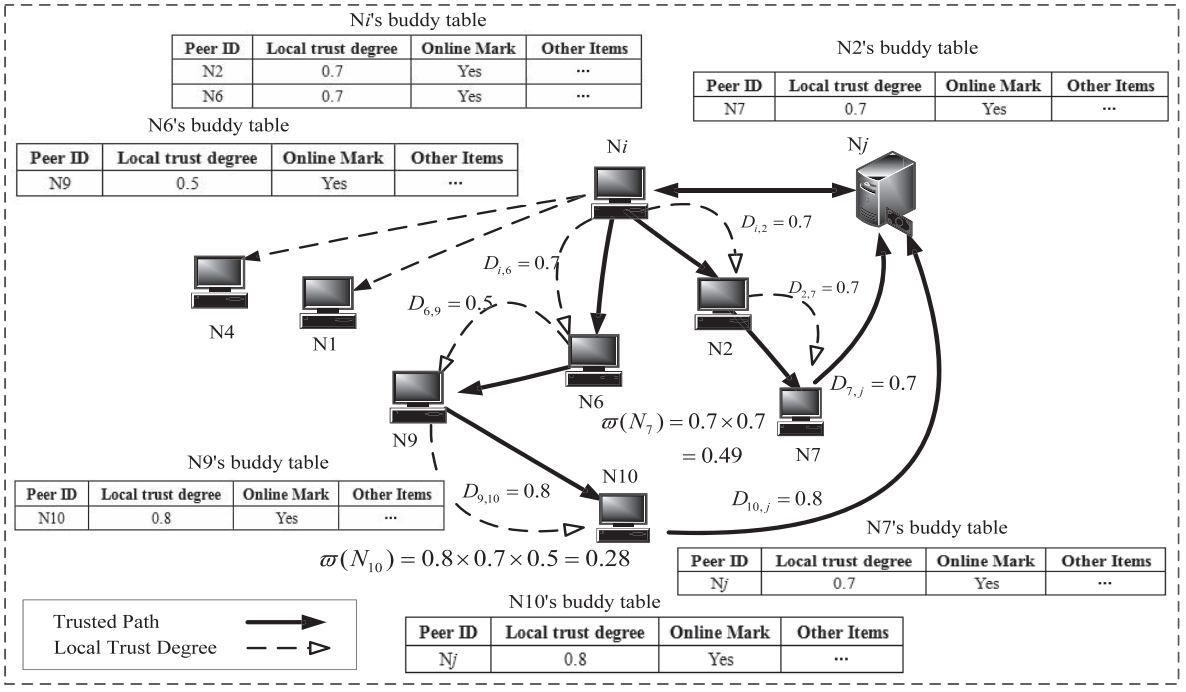


Fig. 6. Implementation mechanism of the SFA overlay in a pure P2P overlay network.

discovery and to save the time and space overhead of the feedback discovery as much as possible.

We use $\eta \in [0, 1]$ to represent *quality-factor*. In feedback discovering, not all FRs are necessary. Only the FRs coming from the SFA overlay whose $\varpi(U_k)$ is greater than the minimum positive number η contribute to the final result. In the SFA overlay, to improve security and reduce risk, when a peer's $\varpi(U_k)$ is lower than η, all feedback from itself and its buddies will be discarded.

We use $\lambda \geq 1$ to represent *distance-factor*. In feedback discovering, λ is used to control the searching depth (hops) in the SFA overlay. When $level(U_k) \leq \lambda$, the intermediate peer U_k will propagate the Query message to all its buddies. If $level(U_k) > \lambda$, U_k will stop propagating the Query message to its buddies.

To sum up the above arguments, in the SFA overlay, each peer may be specified through the threshold η and λ. These two thresholds can be used to control the convergence of feedback discovery according to the system's requirements of scalability and security.

6 Security Enforcement Mechanisms

6.1 Combating FRs' Misbehavior

In an entirely anonymous P2P network environment, FRs may not always reveal their true ratings about other peers. Sometimes, FRs may be colluding cheaters with malicious recommendation behaviors. Collusion occurs when multiple peers or a peer with multiple identities conspire to influence their own or other peers' reputation, certification processes, and so on. The significant effect of these FRs' misbehavior on the trust management of the P2P systems is that these malicious behavior can undermine the reliability of the following factors.

6.1.1 Reliability of the P2P Reputation System

A peer may conspire with associates to influence his/her own or other peers reputation by providing unfair deal evaluations, leading other peers to make incorrect trust decisions that result in unsatisfactory transactions. In a colluding group, peers can give good feedback to everyone in the group and bad feedback to other peers. One example of a colluding group is that of a single physical user generating multiple IDs, such that at least one of his/her IDs obtains a higher rating.

6.1.2 Reliability of P2P Identity Management

P2P systems are distributed systems without a centralized control or organization. Thus, they lack infrastructure services to support public key cryptographic mechanisms that rely on a trusted CA. A peer may create multiple identities to provide bogus support for certificates. Multiple peers may collude to subvert collaborative decision-making for P2P identity management, or a malicious FR (MFR) may discard or alter a vote that is sent through this node without being detected by the requesting peers.

Focusing on the first issue proposed here, we use a risk-probabilistic-based method to counter FRs' misbehavior or at least make their misbehavior costly. As regards the second issue, we present a scheme for identity support using PGP certificates in a fully self-organized manner, which will be introduced in Section 6.2.

In the trust system proposed here, according to (10) and (11), a requesting peer calculates the weight for the FR who provides her/his feedback to the requester. This weight estimates how credible the given feedback is. The basic *idea* of our resistance mechanism is to revise the weights after an unsuccessful recommendation so that the relative weight assigned to the successful FR is increased and the relative weight assigned to the unsuccessful FR is decreased. Based

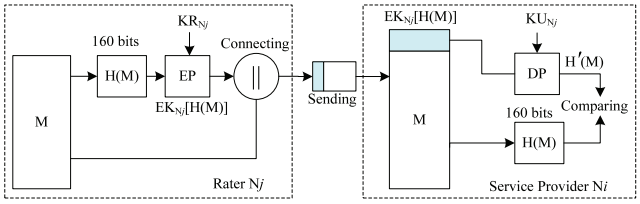


Fig. 7. Use of a PGP-based signature mechanism to support identity-related issues between FRs and SPs.

on the risk-probabilistic model in economics [24], [25], the following function can be used to revise the weight of an FR and partly resist its malicious recommendation behaviors

$$\varpi'(U_k) = (1 - |D_L(N_k, N_j) - \tau_{N_i,N_j}^{(L+1)}|) * \varpi(U_k), \qquad (14)$$

where $D_L(N_k, N_j)$ is the feedback provided by the peer N_k, which is equal to N_ks LTD on peer N_j, and $\tau_{N_i,N_j}^{(L+1)}$ is the probabilistic rating introduced in (4).

Let $\theta = |D_L(N_k, N_j) - \tau_{N_i,N_j}^{(L+1)}|$. Then, $\varpi'(U_k) = (1 - \theta) * \varpi(U_k)$. Thus, in the resistance mechanism proposed here, the value of θ is used to measure how far the feedback provided by N_k is from the real probabilistic rating $\tau_{N_i,N_j}^{(L+1)}$ at time-stamp $(L + 1)$. If the value of θ is large, it is likely that N_k is lying. By the same token, in (14), the larger the value of θ is, the greater likelihood the peer N_k is a malicious FR. For example, if $\theta = 0.8$, the previous value of $\varpi(U_k)$ is 0.9, then the new value of $\varpi'(U_k)$ is decreased to 0.18. In a practical P2P environment, we can set a threshold for θ (such as 0.5). Once the value of θ is detected to be larger than this threshold, the trust calculation module in the service provider will discard the feedback which provided by the suspected malicious node, and then start a re-evaluation of GTD.

6.2 Identity-Related Issues Support

Identity support is probably the most crucial element in a security and trust service for a P2P application system. A robust means of identification will not only protect peers from attacks aimed at identity concealment, but will also let other elements of a security and trust service to function properly and effectively. It provides a kind of security assurance for peers to communicate, collaborate, carry out transactions, and establish trust relationships with known other peers [13], [26].

However, P2P systems lack infrastructure services to support public key cryptographic mechanisms that rely on a CA. They also cannot support self-organized substitutes that require one ore more peers to be the certification authority for other peers, as participation by those peers on a regular basis cannot be guaranteed in such a dynamic network community. To address this issue, we adopt a PGP-based signature mechanism to support identity-related issues in a fully self-organized manner, which can better address the identity-related issues in the SFA overlay. PGP does not use a CA. Instead, every entity certifies the binding of IDs and public keys for other entities. In the more recent PGP specification [27], trust signatures can be used to support the creation of certificate authorities. Fig. 7 illustrates how to use a PGP-based signature mechanism to support identity-related issues between FRs and SPs.

TABLE 2
The Parameters and Their Possible Values

	Description	Possible Values
I_{SP}	Interval of dynamics	3, 12, 30
N_P	Number of peers	10^5
N_B	% of bad SPs	20%, 80%
N_D	% of dynamic SPs	20%, 80%
N_H	% of honest raters	20%, 80%
Q_R	Quality as FR	HFR, MFR, EFR, CFR
Q_S	Quality as SP	Bad, Dynamic, Good
t	2000	Steps of simulation runs
λ	Distance Factor	3, 6, 9
η	Quality Factor	0.50, 0.70
L	history evidence window	10
μ	adjustable positive constant	0.6

Supposing a peer N_j (as an FR), through a PGP-based signature mechanism, wants to send an encrypted message to peer N_i (as an SP). The PGP-based signature process is as follows:

Step 1. Using message M as the input, N_j adopts the *Secure Hash Algorithm* (SHA) to generate a 160-bit hash code $H(M)$.

Step 2. Using the RSA encryption algorithm and its private key KP_{N_j}, N_j encrypts (EP) the hash code $H(M)$ and thus obtain an encrypted hash code $EK_{N_j}[H(M)]$.

Step 3. Connecting the original message M, N_j sends the signature message $\{EK_{N_j}[H(M)] \parallel M\}$ to N_i.

Step 4. Once the signature message is received, N_i uses N_js public key KU_{N_j} to decrypt (DP) the encrypted hash code $EK_{N_j}[H(M)]$ and obtain a decryption message $H'(M)$. At the same time, N_i adopts SHA to generate a 160-bit hash code $H(M)$.

Step 5. If $H'(M) = H(M)$, N_i will accept message M. Otherwise, M will be discarded by N_i. Through this certification process, N_js identity is confirmed accordingly.

Using this PGP-based technology, each peer creates its own private-public key pairs locally, and the format of the certificates is embedded in the form of PGP certificates [27]. Using a signature message as the identity credential in a PGP certificate, this mechanism enables participating peers to establish their online identity in a fully self-organizing manner. It also enables the certification process without relying on any trusted certification authority. For a systematic overview of PGP-based methods, see examples in [26] and [27].

7 Evaluation and Comparison

Referring to [28], we implemented a simulator to test the feasibility of the proposed model (SFA-T model) based on NetLogo [29], a very popular multiagent-based simulation tool implemented based on JAVA in the AI community. It can easily model the parallel and independent peers to simulate interactions among peers. For comparison purposes, we also added two other notable trust mechanisms, P2PRep (Rep-T model) [4] and Power-Trust (Power-T model) [14], in the simulator.

7.1 Simulation Setup

Simulation parameters and default values used in the experiments are summarized in Table 2.

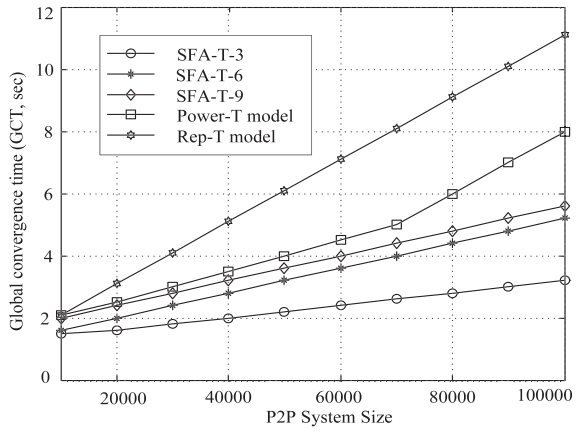


Fig. 8. GCT comparison when $\eta = 0.5$.

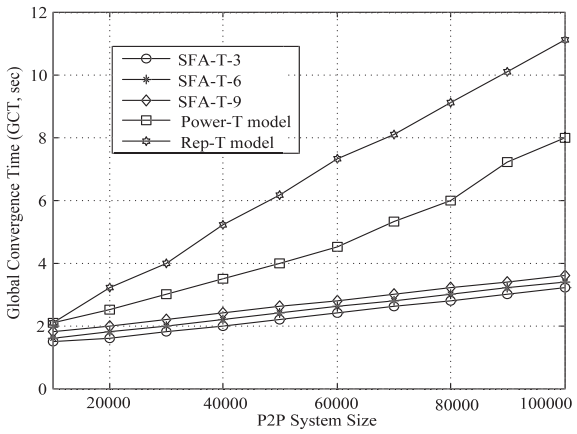


Fig. 9. GCT comparison when $\eta = 0.7$.

In our simulations, types of peers in a simulator is set similarly to [28]. Each peer carries two roles in the system, i.e., as an SP and as an FR (rater). The behaviors of a peer as an FR can be one of four types: honest FR (HFR), malicious FR, exaggeration FR (EFR), and collusive FR (CFR). HFR always gives correct feedback. MFR always gives the opposite opinion $(1 - T)$ to others. EFR exaggerates their ratings through an exaggerating factor, which is 0.5 in our simulation. For this type of FR, the feedback $T + e * (T - 0.5) = T + 0.5 * (T - 0.5)$ will be sent out. For the collusive peers, CFR sends out 1 for the peers in the collusive group and 0 for the entities outside the group. The quality of an SP can be Good, Bad, or Dynamic. Good SPs always provide good services. Bad SPs provide one of the three bad services randomly: Low-grade, No-response, and Byzantine [9]. Dynamic SPs provide the service with one of the two kinds of qualities, Good or Bad.

Dynamic SPs change their qualities in a specific interval (I_{SP}). Decreasing I_{SP} can make the SPs change their qualities faster. The total number of peers is N_P. N_B and N_D are the percentage of bad and dynamic SPs, respectively. Thus, there will be $N_P * N_D$ dynamic SPs in the whole network and $N_P * N_B$ bad SPs. The rest are good SPs, i.e., $N_P * (1 - N_D - N_B)$ good SPs. N_H is the percentage of honest FRs. If $N_H \neq 0$, then there are $N_P * N_H$ honest FRs; and the remaining $N_P * (1 - N_H)$ FRs are all bad raters. Instead of using the physical running time, we use the notion of time-step, which is introduced in Netlogo, to calculate the simulation time. Within each time-step, a peer will finish all the activity including service request, service providing, trustworthiness value update, and rating dissemination.

7.2 Evaluation of Scalability

We use two performance metrics to illuminate our model's scalability.

1. *Global convergence time* (GCT), is the total time of feedback aggregation. GCT is a useful to evaluate the scalability of the whole system. Most previous works focused on the system behavior when the system is stable. Here we argue that the convergence time is an important metric to measure how fast the system can reach a stable point, especially in a dynamic environment.

2. *Average messaging overhead* (AMO), is the average storage overhead of all kinds of data tables and messages in feedback aggregation. In some special environments, such as wireless networks, there is a strict limitation of storage. AMO of the feedback system will be one of the considerations for the system developer. The AMO consists of two parts: message overhead and data table overhead. Message overhead includes Query message overage (QO_t) and QueryHit message overage (QH_t). Data table overhead includes trustworthy buddy table (TBT_t), rating queue (RQ_t), and blacklist (B_t).

Suppose $QO_t + QH_t + TBT_t + RQ_t + B_t$ (bytes) is the total storage overhead consumed by the system at t, and N is the total number of peers, then AMO is defined as

$$AMO = \frac{QO_t + QH_t + TBT_t + RQ_t + B_t}{N}. \quad (15)$$

In the first group simulations, we suppose that most SPs and FRs are good in the community, where 80 percent SPs are good and 80 percent FRs are honest. The remaining 20 percent SPs are fixed malicious SPs, and 20 percent FRs are malicious. As we want to study the affect of the overlay in the feedback aggregation, setting up a community close to the real environment is natural. In Fig. 8, symbol Power-T denotes the Power-Trust model, Rep-T denotes the P2PRep-Trust, and SFA-T denotes our model. SFA-T-3 denotes SFA-T model in the condition of DF $\lambda = 3$, whereas SFA-T-6 and SFA-T-9 denote $\lambda = 6$ and $\lambda = 9$, respectively. In the simulations, to acquire a more accurate performance evaluation of scalability, we set the max-searching-depth to $\lambda = 9$. In what follows, we will observe the computing outcomes under different combinations of λ and η.

Figs. 8 and 9 show the compared outcomes of GCT computed by the three mechanisms under different network scales. In Fig. 8, when $\eta = 0.5$, the SFA-T model clearly outperforms other algorithms from the viewpoint of GCT. In the three conditions with different λ values, the GCT of the SFA-T model is only the half that of Rep-T. From the viewpoint of the SFA-T model itself, SFA-T-9 is slightly worse than SFA-T-3 and SFA-T-6. This indicates that with the significant increase of λ, GCT has no apparent increase. With the rapid increase in the network scale, GCT increases regularity in the SFA-T model. This shows that the SFA-T model has better scalability than the other models.

Fig. 9 shows the compared outcomes when $\eta = 0.7$. the SFA-T-6 almost has the same GCT as SFA-T-9, indicating that

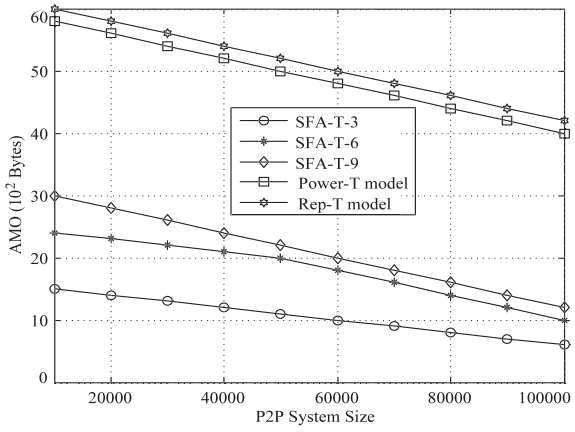


Fig. 10. AMO comparison when $\eta = 0.5$.

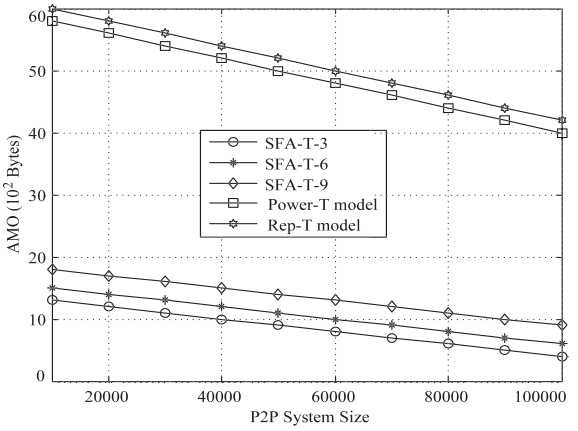


Fig. 11. AMO comparison when $\eta = 0.7$.

SFA-T-3 and SFA-T-9 almost have the same SFA search size. In Fig. 9, SFA-T-6 and SFA-T-9 have a smaller GCT, indicating that the system has faster convergence rate. The reason is the number of participating peers reduced. A higher value of η implies that the system has higher security requirements. If a peer's LTD is low, its father peer in the SFA overlay will not forwarding Query message to it. Thus, participating peers will decrease as well as the system's GCT.

Figs. 10 and 11 plot the AMO per individual peer computed by the three mechanisms under different network scales. In the two figures, the storage cost of Pow-T and Rep-T is much higher than that of SFA-T under the condition of three λ value. This indicates that using the SFA-T system, peers do experience a noticeably lower storage overhead. In other words, our model is scalable in handling an even larger number of P2P services. In Fig. 10, for 10^5 peers, our system needs under 10×10^2 bytes of storage overhead, whereas the Pow-T system needs 48×10^2 bytes, and Rep-T needs 50×10^2 bytes to perform the same task. In Fig. 11, SFA-T-6 and SFA-T-9 have a low and similar AMO. The reason for this is the same as that in Fig. 10. Also in Figs. 10 and 11, with the increase in peer size, the AMO gradually decreases. This justly indicates the P2P system's virtue, the larger the system's size, the stronger the system's processing capability is. That is, the storage information is evenly distributed in all P2P nodes: the larger the system's nodes, the lesser is the AMO.

7.3 Evaluation of Dynamic Adaptability

In general, the dynamics is caused by three factors: 1) *peers' dynamics*: all peers in the network can randomly join or leave the networks; 2) *SP's dynamics*: any SP can dynamically change its identity between good service and bad service; and 3) *service dynamics*, in a busy P2P system, there are more service requests than an idle system. Correspondingly, in our simulation, we use three parameters to reflect the dynamism of a P2P system [28].

7.3.1 Service Requesting Frequency (SRF $\in [0, 1]$)

For each peer in the simulator, after a random time ($\lceil radom(D+1) \rceil <$ SRF, $D \in [1, N]$), it will send out a service request to an SP. The bigger the SRF value is, indicating that the system is a busy system.

7.3.2 Service Dynamic Factor (SDF $\in [0, 1]$)

After a random time ($\lceil radom(Z+1) \rceil <$ SDF, $Z \in [1, N]$), an SP oscillates providing good and bad services.

7.3.3 Dynamic Peers Percentage (DPP $\in [0, 1]$)

If there are DPP $\times N_P$ peers that are unstable, they can leave or join the system at any moment.

We use the *successful service percent* (SSP) to show the system's dynamic adaptability

$$SSP = \frac{S_g(t)}{S_g(t) + S_b(t)} * 100\%, \tag{16}$$

where $S_g(t)$ is the total number of good service provided by the trust system at time-step t, and $S_b(t)$ is the total number of bad service at t.

A highly SSP indicates a system with robust dynamic adaptability. In the experiments, we configured all kinds of FRs percent as HFR 80 percent, MFR 10 percent, EFR 5 percent, and CFR 5 percent. This is in accordance with a practical P2P system, where majority of the peers are honest ($\mathrm{HFR} = 80\%$), and only a small portion of the peers is malicious (20 percent). According to the four conditions, we discuss the following related problems:

1. Idle and stable environment, where $\mathrm{SRF} = 0.2$, $\mathrm{SDF} = 0.2, \mathrm{DPP} = 0.2$.
2. Busy but stable environment, where $\mathrm{SRF} = 0.8$, $\mathrm{SDF} = 0.2, \mathrm{DPP} = 0.2$.
3. Dynamic but idle environment, where $\mathrm{SRF} = 0.2$, $\mathrm{SDF} = 0.8, \mathrm{DPP} = 0.8$.
4. Highly busy and dynamic environment, where $\mathrm{SRF} = 0.8, \mathrm{SDF} = 0.8, \mathrm{DPP} = 0.8$.

First, let us look into the case of an idle and stable environment. In Table 3A, the three mechanisms have a corresponding good robustness in providing good services; the SSPs are all beyond 96 percent. However, the SSP of SFA-T and Power-T is slightly higher than that of Rep-T. The average SSP of SFA-T and Power-T is beyond 96.4 percent, and the average SSP of Rep-T is 95.2 percent. Table 3B shows the outcome computed by the three mechanisms under a busy but stable environment, where the three dynamic parameters are $\mathrm{SRF} = 0.8, \mathrm{SDF} = 0.2$, and $\mathrm{DPP} = 0.2$. All three mechanisms also have accuracy, but Rep-T has a

TABLE 3
Evaluation of Dynamic Adaptability

A. SSP comparison in an idle and stable environment.						
Time-stamps	400	800	1200	1600	2000	**Average**
Rep-T	94.92	95.12	95.32	95.34	95.32	95.20
Power-T	96.44	96.44	96.51	96.45	96.47	96.46
SFA-T	97.12	97.13	97.21	97.42	97.44	97.26
B. SSP comparison in a busy but stable environment.						
Time-stamps	400	800	1200	1600	2000	**Average**
Rep-T	81.99	82.13	82.15	82.12	82.12	82.10
Power-T	90.23	90.36	90.30	90.42	90.40	90.34
SFA-T	96.23	96.37	96.32	96.34	96.27	96.31
C. SSP comparison in a dynamic but idle environment.						
Time-stamps	400	800	1200	1600	2000	**Average**
Rep-T	78.44	78.92	79.35	79.35	79.23	79.05
Power-T	87.87	87.90	87.92	87.92	87.90	87.90
SFA-T	95.73	95.92	95.87	95.89	95.90	95.86
D. SSP comparison in a busy and dynamic environment.						
Time-stamps	400	800	1200	1600	2000	**Average**
Rep-T	72.00	72.21	72.14	72.21	72.25	72.16
Power-T	82.23	82.22	82.24	82.23	82.12	82.21
SFA-T	90.17	90.23	90.10	90.24	90.24	90.20

13.1 percent drop in SSP. Thus, SFA-T and Pow-T are the robust trust mechanism which can obtain a better SSP in a busy but stable environment.

To study the robustness of these systems in a highly dynamic environment, SDF and DPP are set to 0.8 in this group of simulations. Thus, we can see the obvious effect brought by the highly dynamic environment. In Table 3C, in a dynamic but idle environment, SFA-T is still the best system to obtain the best SSP, which has approximately 95.86 percent of SSP. However, compared with that of the SFA-T, the robustness significantly decreases for Power-T and Rep-T. Especially for Rep-T, its SSP sharply drops down to 80 percent. When SRF is set to 0.8, the network is show to be not only a busy system but also a highly dynamic system. Table 3D shows the computing outcomes under this highly busy and dynamic environment. SSP of all three systems has a notable decrease. However, compared with Power-T and Rep-T, SFA-T is still the best system to obtain the best robustness under a highly busy and dynamic environment.

7.4 Evaluation of Malicious Peer Behaviors

We evaluate the effectiveness of the SFA-T system against various malicious peer behaviors. The experiment was performed under both settings of independent malicious peers and collusive malicious peers. We compute the *root-mean-square* (RMS) of the aggregated global trust of all peers. A lower RMS error indicates higher accuracy. The RMS error is defined as follows [14], [15]:

$$RMS\ error = \sqrt{\frac{\sum((T(N_i) - T'(N_i))/T(N_i))^2}{N_P}}, \qquad (17)$$

where $T(N_i)$ and $T'(N_i)$ are the actual and evaluated global trust degree of peer N_i, respectively.

We model two types of malicious behaviors. One type reports dishonest GTD (e.g., malicious behaviors caused by MFR and EFR), and the other type collaborates abusers with each other to boost up their own ratings (e.g., malicious behaviors caused by CFR).

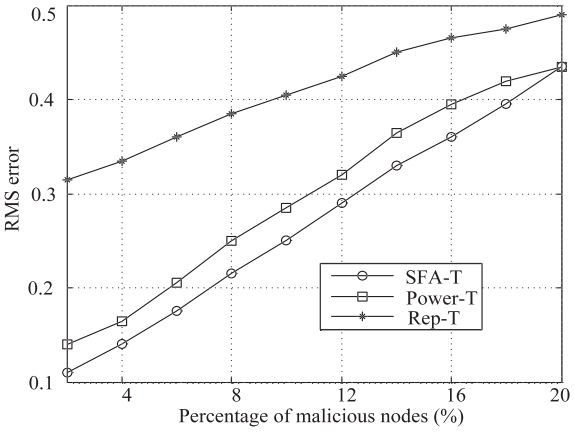


Fig. 12. RMS error comparison with different numbers of independent malicious peers.

Fig. 12 shows the RMS error incurred by malicious peers reporting false LTD independently. The probability of a node being malicious is modeled by the inverse of its GTD T. Thus, it always gives the opposite opinion $(1 - T)$ to others, as a node providing corrupted services is highly likely to issue dishonest ratings. With 8 percent malicious peers, the SFA-T system has 4 percent less RMS error than Power-T and has 16 percent less RMS error than Rep-T. As the percentage increases, the RMS error gap is closing up between the three systems. In Fig. 12, SFA-T is the most effective among the three trust systems in resisting independent malicious peers.

In Fig. 13, we model the collusive peers working collaboratively to abuse the system. We report the RMS aggregation errors under different collusion group sizes (the number of malicious peers in a group). The malicious peers will send out 1 for the peers in the collusive group and 0 for the entities outside the group. In this simulation, the percentage of malicious FRs is set to 10 percent. This percentage of malicious FRs indicates that the community is a relatively good community (i.e., with fewer malicious FRs). In Fig. 13, the SFA-T shows its robustness against

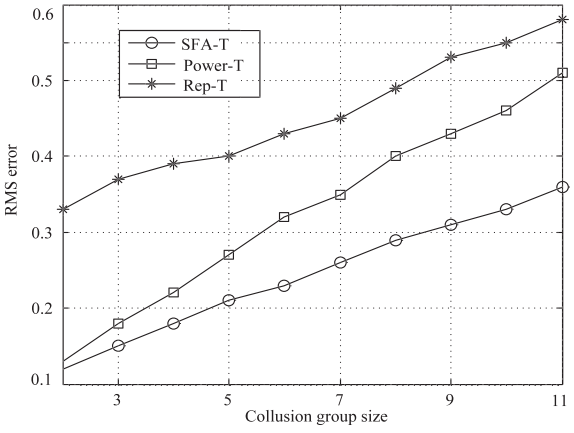


Fig. 13. RMS error comparison with different numbers of collusive malicious peers.

collusive peer groups of various sizes. The Rep-T system is less resistant to abuses by large collusive peer groups. The RMS error of Power-T system falls somewhere in between.

8 Conclusion and Future Works

In this paper, through a scalable perspective, the SFA overlay is presented, which not only can significantly enhance the scalability of the trust system, but can also reduce the risk and improve system efficiency. Meanwhile, based on the SFA overlay, an adaptive GTD computing method can be defined. This method surpasses the limitations of the existing approaches, where weights of the trust factors are assigned subjectively.

We have shown that our system yields very good results in many typical cases and that the proposed mechanism is robust against various complicated environments. However, there are still many open issues and improvements we can apply to the current model.

1. First, we are interested in combining trust management with intrusion detection to address the concerns of sudden and malicious attacks [30]. Implementing and evaluating our proposed model on various P2P systems, such as distributed file sharing and P2P grid computing, is another direction for future research.
2. Another open problem is how to predict accurately the trust value of newly joined peers with only few feedback reports and how to motivate more users to submit their feedback to the trust evaluation engine. The selection of an optimal configuration for many design parameters of our proposed solutions is also an important issue to be studied future.
3. Last, we intend to deploy related algorithms in a practical decentralized setting to observe the effectiveness of proposed trust techniques where many registry peers exchange among each other information about users, and services' quality data.

Acknowledgments

This work was supported by the National Nature Science Foundation of China (No. 61003281); the National Basic Research Program of China (973 Program) (No. 2012CB821200 (2012CB821206)); the Beijing Natural Science Foundation (No. 4122051); Fok Ying Tung Education Foundation (No. 132032) and the Chinese Universities Scientific Fund (No. BUPT 2012RC0210). The authors would like to appreciate the associate editor and the anonymous reviewers for their insightful suggestions to improve the quality of this paper.

References

[1] H. Li and M. Singhal, "Trust Management in Distributed Systems," *Computer,* vol. 40, no. 2, pp. 45-53, 2007.

[2] X. Li and X. Gui, "Research on Dynamic Trust Model in Large-Scale Distributed Environment," *J. Software,* vol. 18, no. 6, pp. 1510-1521, 2007.

[3] F. Cornelli, E. Damiani, S.D.C. di Vimercati, S. Paraboschi, and P. Samarati, "Choosing Reputable Servents in a P2P Network," *Proc. 11th Int'l World Wide Web Conf.,* pp. 376-386, 2002.

[4] E. Damiani, S.D.C. di Vimercati, S. Paraboschi, and P. Samarati, "Managing and Sharing Servants' Reputations in P2P Systems," *IEEE Trans. Knowledge and Data Eng.,* vol. 15, no. 4, pp. 840-854, July/Aug. 2003.

[5] E. Damiani, S.D.C. di Vimercati, S. Paraboschi, P. Samarati, and F. Violante, "A Reputation-Based Approach for Choosing Reliable Resources in Peer-to-Peer Networks," *Proc. Ninth ACM Conf. Computer and Comm. Security,* pp. 207-216, 2002.

[6] R. Aringhieri, E. Damiani, S.D.C. di Vimercati, S. Paraboschi, and P. Samarati, "Fuzzy Techniques for Trust and Reputation Management in Anonymous Peer-to-Peer Systems," *J. Am. Soc. for Information Science and Technology,* vol. 57, no. 4, pp. 528-537, 2006.

[7] E. Damiani, S.D.C. di Vimercati, P. Samarati, and M. Viviani, "A WOWA-Based Aggregation Technique on Trust Values Connected to Metadata," *Electronic Notes in Theoretical Computer Science,* vol. 157, no. 3, pp. 131-142, 2006.

[8] L. Xiong and L. Liu, "Peer-Trust: Supporting Reputation-Based Trust in Peer-to-Peer Communities," *IEEE Trans Data and Knowledge Eng.,* vol. 16, no. 7, pp. 843-857, July 2004.

[9] Z. Liang and W. Shi, "TRECON: A Trust-Based Economic Framework for Efficient Internet Routing," *IEEE Trans. Systems, Man, and Cybernetics, Part A: Systems and Humans,* vol. 40, no. 1, pp. 52-67, Jan. 2010.

[10] Z. Liang and W. Shi, "Enforcing Cooperative Resource Sharing in Untrusted Peer-to-Peer Environments," *J. Mobile Networks and Applications,* vol. 10, no. 6, pp. 771-783, 2005.

[11] B. Yu, M.P. Singh, and K. Sycara, "Developing Trust in Large-Scale Peer-to-Peer Systems," *Proc. IEEE First Symp. Multi-Peer Security and Survivability,* pp. 1-10, 2004.

[12] A.A. Selcuk, E. Uzun, and M.R. Pariente, "A Reputation-Based Trust Management System for P2P Networks," *Int'l J. Network Security,* vol. 6, no. 2, pp. 227-237, 2008.

[13] P. Dewan and P. Dasgupta, "P2P Reputation Management Using Distributed Identities and Decentralized Recommendation Chains," *IEEE Trans. Knowledge and Data Eng.,* vol. 22, no. 7, pp. 1000-1013, July 2010.

[14] R. Zhou and K. Hwang, "Power-Trust: A Robust and Scalable Reputation System for Trusted Peer-to-Peer Computing," *IEEE Trans. Parallel and Distributed Systems,* vol. 18, no. 5, pp. 460-473, Apr. 2007.

[15] R. Zhou, K. Hwang, and M. Cai, "GossipTrust for Fast Reputation Aggregation in Peer-to-Peer Networks," *IEEE Trans. Knowledge and Data Eng.,* vol. 20, no. 9, pp.1282-1295, Sept. 2008.

[16] M. Ripeanu, I. oster, and A. Iamnitchi, "Mapping the Gnutell Network: Properties of Large-Scale P2P Systems and Implications for System Design," *IEEE Internet Computing,* vol. 6, no. 1, pp. 50-57, Sept. 2002.

[17] S. Saroiu, K.P. Gummadi, R.J. Dunn, S.D. Gribble, and H.M. Levy, "An Analysis of Internet Content Delivery Systems," *Proc. Fifth Symp. Operating Systems Design and Implementation (OSDI '02),* pp. 86-90, 2002.

[18] L. Liu and W. Shi, "Trust and Reputation Management," *IEEE Internet Computing,* vol. 14, no. 5, pp. 10-13, Sept./Oct. 2010.

[19] S. Song, K. Hwang, R. Zhou, and Y.K. Kwok, "Trusted P2P Transactions with Fuzzy Reputation Aggregation," *IEEE Internet Computing,* vol. 9, no. 6, pp. 24-34, Nov./Dec. 2005.

[20] X. Li, F. Zhou, and X. Yang, "Developing Dynamic P2P Trust Model Using Theory of Entropy-Based Multi-Source Information Fusion," *Int'l J. Innovative Computing, Information and Control,* vol. 7, no. 2, pp. 777-790, 2011.

[21] X. Li, F. Zhou, and X. Yang, "A Multi-Dimensional Trust Evaluating Model for Large-Scale P2P Computing," *J. Parallel and Distributed Computing,* vol. 71, no. 6, pp. 837-847, 2011.

[22] X. Li and X. Gui, "A Comprehensive and Adaptive Trust Model for Large-Scale P2P Networks," *J. Computer Science and Technology,* vol. 24, no. 5, pp.868-882, 2009.

[23] The Gnutella Protocol v0.4 (Revision1.2), http://www.clip2.com/GnutellaProtocol04.pdf, 2001.

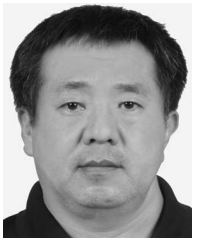
[24] W. Keller, *Statistics for Management and Economics,* eight ed. South-Western College Pub, 2008.

[25] F. Liu, Y. Chen, K. Dai, Z. Wang, and Z. Cai, "Research on Risk Probability Estimating Using Fuzzy Clustering for Dynamic Security Assessment," *Proc. 10th Int'l Conf. Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing,* vol. 3642/2005, pp. 539-547, 2005.

[26] H. Osman and H. Taylor, "Identity Support in a Security and Trust Service for Ad Hoc M-Commerce Trading Systems," *Proc. Int'l Conf. Advanced Information Networking and Applications*, pp. 285-290, 2011.
[27] J. Callas et al., OpenPGP Specification, *IETF RFC 4880*, 2007.
[28] Z. Liang and W. Shi, "Analysis of Recommendations on Trust Inference in Open Environment," *Performance Evaluation*, vol. 65, no. 2, pp. 99-128, 2008.
[29] S. Tisue, NetLogo, http://ccl.northwestern.edu, 2012.
[30] F.D. Rango, "Trust-Based SAODV Protocol with Intrusion Detection, Trust Management and Incentive Cooperation in MANETs," *Int'l J. Interdisciplinary Telecomm. and Networking*, vol. 1, no. 4, pp. 54-70, 2009.

Xiaoyong Li received the BS degree in computer science from Northwest Normal University, China, the MS degree in computer science from East China Normal University, China, and the PhD degree in computer science major from Xi'an Jiaotong University, in 1998, 2004, and 2009, respectively. Currently, he is working as an associate professor of computer science at Beijing University of Posts and Telecommunications University. Now, he is in charge of a project of the National Nature Science Foundation of China (No. 61003281), and a Beijing Natural Science Foundation (No. 4122051). His current research interests mainly include cloud computing, network security and trusted system. As the first author, he has published more than 30 journal papers, obtained five patents and three software copyrights in cloud computing, P2P computing, grid computing and other fields. In 2009, he was awarded outstanding graduates in Shaanxi Province, China.

Feng Zhou received the MS degree in computer science major from Beijing University of Posts and Telecommunications in 1989. He is a full professor of the School of Computer Science and Technology, Beijing University of Posts and Telecommunications, China. He is a director of the Center of Computer Architecture (CCR) at Beijing University of Posts and Telecommunications University. His research interests include mobile internet, embedded computing, and communication protocols. He is the author and coauthor of a high number of papers published in journals and conference proceedings. He is also a committee member of Information Storage Technology, China Computer Federation.



Xudong Yang received the MS degree in computer science major from Beijing University of Posts and Telecommunications in 1990. He is a full professor of the School of Computer Science and Technology, Beijing University of Posts and Telecommunications, China. He is vice director of the Center of Computer Architecture (CCR) at Beijing University of Posts and Telecommunications University. His research interests include Internet of Things (IOT), mobile internet, and embedded computing. He has published more than 20 papers in international journals and conferences.