PowerTrust: A Robust and Scalable Reputation System for Trusted Peer-to-Peer Computing

Runfang Zhou, *Member*, *IEEE*, and Kai Hwang, *Fellow*, *IEEE Computer Society*

Abstract—*Peer-to-Peer* (P2P) reputation systems are essential to evaluate the trustworthiness of participating peers and to combat the selfish, dishonest, and malicious peer behaviors. The system collects locally-generated peer feedbacks and aggregates them to yield the global reputation scores. Surprisingly, most previous work ignored the distribution of peer feedbacks. We use a *trust overlay network* (TON) to model the trust relationships among peers. After examining the eBay transaction trace of over 10,000 users, we discover a power-law distribution in user feedbacks. Our mathematical analysis justifies that power-law distribution is applicable to any dynamically growing P2P systems, either structured or unstructured. We develop a robust and scalable P2P reputation system, *PowerTrust*, to leverage the power-law feedback characteristics. The PowerTrust system dynamically selects small number of power nodes that are most reputable using a distributed ranking mechanism. By using a look-ahead random walk strategy and leveraging the power nodes, PowerTrust significantly improves in global reputation accuracy and aggregation speed. PowerTrust is adaptable to dynamics in peer joining and leaving and robust to disturbance by malicious peers. Through P2P network simulation experiments, we find significant performance gains in using PowerTrust. This power-law guided reputation system design proves to achieve high query success rate in P2P file-sharing applications. The system also reduces the total job makespan and failure rate in large-scale, parameter-sweeping P2P Grid applications.

Index Terms—Peer-to-Peer system, overlay network, distributed hash table, reputation system, eBay trace data set, distributed file sharing, P2P Grids, PSA benchmark, system scalability.

1 Introduction

In recent years, *peer-to-peer* (P2P) computing has gained its popularity in many large-scale distributed applications over the Internet. These include distributed file-sharing [22], digital content delivery [23], and P2P Grid computing [8]. Despite the demand of robustness and scalability of P2P systems, the anonymous and dynamic nature of peer activities make them often very vulnerable to abuses by selfish and malicious peers [11], [27]. For example, most P2P file-sharing networks, e.g., Gnutella, consist of autonomous peers with special self-interests. There is no efficient way to prevent malicious peers from joining the open networks.

To encourage resource sharing among peers and combat malicious peer behaviors, reputation management is essential for peers to assess the trustworthiness of others and to selectively interact with more reputable ones [34]. Without an efficient reputation management facility, peers will have little incentive to contribute their computing or bandwidth resources. The peers may hesitate to interact with unknown peers due to the concern of receiving corrupted or poisoned files or being exploited by malware [11]. Identifying trustworthy peers is especially necessary in commercial P2P applications, such as P2P auctions [18], trusted content delivery [23], pay-per-transaction [29], and P2P service discovery [20].

A *reputation system* calculates the global reputation score of a peer by considering the opinions (i.e., *feedbacks*) from all other peers who have interacted with this peer. After a peer completes a transaction, e.g., downloading a music file, the peer will provide his or her feedback for other peers to use in future transactions. By making the reputation scores publicly available, peers are able to make informed decisions about which peers to trust.

The eBay reputation system is a simple and successful one, since it has a centralized authority to manage all user feedback scores. However, in an open and decentralized P2P system, peers will not have any centralized authority to maintain and distribute reputation information. Instead, most existing P2P reputation systems calculate the global reputation scores by aggregating peer feedbacks in a fully distributed manner [1], [3], [6], [13], [15], [26], [29], [30], [34], [36]. Building an efficient P2P reputation system is a challenging task due to several intrinsic requirements of large-scale P2P systems. Listed below are six key issues that should be addressed in the design of a cost-effective P2P reputation system.

- ***High accuracy***. To help distinguish reputable peers from malicious ones, the system should calculate the reputation scores as close to their real trustworthiness as possible.
- ***Fast convergence speed***. The reputation of a peer varies over time. The reputation aggregation should converge fast enough to reflect the true changes of peer behaviors.
- ***Low overhead***. The system should only consume limited computation and bandwidth resources for peer reputation monitory and evaluation.

- *The authors are with the Department of Electrical Engineering, University of Southern California, 3740 McClintock Avenue, Room 212, Los Angeles, CA 90089-2562. E-mail: {rzhou, kaihwang}@usc.edu.*

Manuscript received 7 Nov. 2005; revised 21 Mar. 2006; accepted 26 June 2006; published online 9 Jan. 2007.
Recommended for acceptance by J. Srivastava.
For information on obtaining reprints of this article, please send e-mail to: tpds@computer.org, and reference IEEECS Log Number TPDS-0463-1105.
Digital Object Identifier no. 10.1109/TPDS.2007.1015.

TABLE 1
Comparison of PowerTrust with Two Established P2P Reputation Systems

Reputation System	Local Trust Evaluation	Global Reputation Aggregation	Implementation Overhead	Scalability & Reliability
EigenTrust at Stanford Univ. [13]	Using sum of positive and negative ratings	Using pre-trust peers to compute global scores from trust matrix	Moderate overhead experienced in assigning score managers and in messaging for global score aggregation	Limited scalability and reliability if pre-trust peers leave
PeerTrust at Georgia Tech [34]	Normalized rating on each transaction	Peer calculates trust score over five factors in a distributed manner.	Moderate overhead experienced in global score calculation over five factors and on the establishment of the trust manager	Partially scalable and resistant to malicious peers
PowerTrust at USC (This paper)	Use Bayesian method to generate local trust scores	Distributed ranking mechanism and LWR strategy to aggregate global reputation scores	Low overhead in using locality-preserving hashing to locate power nodes. Global aggregation time drops sharply with look-ahead random walk strategy applied	Highly scalable and robust with dynamic peer join and leave and malicious peers

- ***Adaptive to peer dynamics.*** Peer joins and leaves an open P2P system dynamically. The system should adapt to this peer dynamics instead of relying on predetermined peers.
- ***Robust to malicious peers***. The system should be robust to various attacks by both independent and collective malicious peers.
- ***Scalability***. The system should be able to scale to serve a large number of peers in term of accuracy, convergence speed, and extra overhead per peer.

As global reputation scores are aggregated from local feedbacks, the distribution property of feedbacks plays a significant role in the design of an efficient reputation system. Surprisingly, most previous work either ignored the distribution of peer feedbacks or assumed an arbitrary random distribution, which could be misleading.

We propose a *trust overlay network* (TON) to model the local trust and reveal feedback relationship among peers. We argue that the eBay user behavior is decentralized by nature, as the peers are autonomous entities to make decisions individually. So, it is fully justified to model the eBay user behavior by a decentralized trust model. After examining the eBay transaction traces of over 10,000 users, we discover a power-law distribution in user feedbacks. We design the PowerTrust system by leveraging the power-law distribution of peer feedbacks. This design leads to fast aggregation speed and accuracy, robustness against malicious peers, and high scalability in large-scale P2P applications. This article provides both theoretical foundations and experimental results to validate the design of the PowerTrust system, which extends significantly from our preliminary results reported in [38].

The remaining parts of this paper are organized as follows: Section 2 reviews existing work on P2P reputation systems. We introduce the new PowerTrust system concept and the use of trust overlay network in Section 3. We analyze in Section 4 the eBay trace data to reveal the power-law distribution of peer feedbacks. Section 5 specifies the detailed design of our PowerTrust system and the reputation aggregation algorithms used. We evaluate the performance attributes of the PowerTrust system in Section 6 and report its application benchmark results in Section 7. Finally, we conclude with a summary of contributions and make suggestions for further research work.

2 Related Works

A formal treatment of trust and reputation was given by Aberer and Despotovic [1] in the context of P2P networks. Their approach is based on a decentralized storage method (P-Grid). The information provided by P-Grid is used to assess the probability that an agent will cheat in the future. This approach suffers from several shortcomings, e.g., trust is evaluated only according to referrals from neighbors, not based on all information in the system. Buchegger and Budded presented a reputation evaluation approach based on Bayesian learning technique [3]. In their approach, the first-hand information is exchanged frequently and the second-hand information is merged, if it is compatible with current reputation rating.

Xiong and Liu [34] presented an approach that avoids aggregation of the individual interactions. Their PeerTrust system computes the trustworthiness of a given peer as the average feedback weighted by the scores of the feedback originators. The limitation of this approach is that the computation convergence rate in large-scale P2P systems is not provided. The five factors used in their trust model must be retrieved with a heavy overhead.

The EigenTrust mechanism [13] aggregates trust information from peer by having them perform a distributed calculation approaching the eigenvector of the trust matrix over the peers. EigenTrust relies on good choice of some pretrusted peers, which are supposed to be trusted by all peers. This assumption may be over optimistic in a distributed computing environment. The reason is that pretrust peers may not last forever. Once they score badly after some transactions, the EigenTrust system may not work reliably.

Table 1 compares our PowerTrust system with the established EigenTrust and PeerTrust systems in four technical aspects. Our system concept is introduced in Section 3.1. The system construction algorithms are described in Section 5. The table entries are justified in subsequent sections.

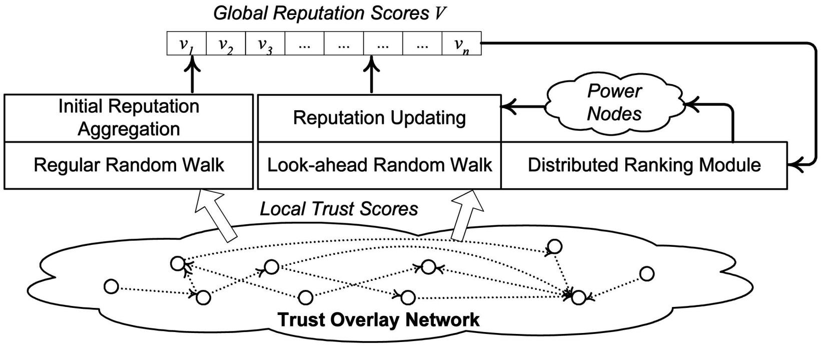

Global Reputation Scores V
v_1 v_2 v_3 ... ... v_n
Initial Reputation Aggregation
Reputation Updating
Power Nodes
Regular Random Walk
Look-ahead Random Walk
Distributed Ranking Module
Local Trust Scores
Trust Overlay Network

Fig. 1. Functional modules in the PowerTrust system and the control flow pattern in local trust score collection and global reputation aggregation.

3 Our PowerTrust System Approach

Our PowerTrust system makes a distinction in robustness and scalability from previously reported P2P reputation systems. In this section, we introduce the system concept and discuss new features in PowerTrust. The underlying trust overlay network is specified for modeling peer feedbacks in global reputation aggregation.

3.1 The PowerTrust System Concept

Inspired by the power-law findings in peer feedbacks, the PowerTrust system dynamically selects a few *power nodes* that are most reputable by using a distributed ranking mechanism. The good reputation of power nodes is accumulated from the running history of the system. Like a democratic system, power nodes are dynamically replaceable, if they become less active or demonstrate unacceptable behavior. They play a crucial role in both local and global scoring processes. We leverage more on their roles to aggregate and produce the global reputation scores.

Fig. 1 shows the major building blocks in a PowerTrust system. First, a *trust overlay network* (TON) is built on top of all peers (nodes) in a P2P system. All peers evaluate each other, whenever a transaction takes place between a peer pair. Therefore, all peers send *local trust scores* among themselves, frequently. These scores are considered as the raw data input to the PowerTrust system. The system supposes to aggregate the local scores to calculate the global reputation score of each participating peer. All global scores form a *reputation vector*, $V = (v_1, v_2, v_3, \dots, v_n)$, which is the output of the PowerTrust system. All global scores are normalized with $\sum_i v_i = 1$, where $i = 1, 2, \dots, n$ and n is the TON network size.

The system is built with five functional modules as shown in Fig. 1. The *regular random walk* module supports the *initial reputation aggregation*. The *look-ahead random walk* (LRW) module is used to update the reputation score, periodically. To this end, the LRW also works with a *distributed ranking module* to identify the power nodes. The system leverages the power nodes to update the global reputation scores. PowerTrust achieves high aggregation speed and accuracy, robustness to resist malicious peers, and high scalability to support large-scale P2P applications. We will discuss the details of these functional modules in the subsequent sections.

3.2 Trust Overlay Network (TON)

A TON is a virtual network on top of a P2P system. We represent a TON by a directed graph in Fig. 2. The graph nodes correspond to the peers. The directed edges or links are labeled with the feedback scores between two interacting peers. The feedback score is issued by a peer (source of the link) for the service provided by the interacting peer (destination of the link). For example, node N_5 after downloading music files from nodes N_2 and N_7 issues the feedback scores, 0.7 and 0.3, to the two provider nodes, respectively. If a node gets more than one service from the same provider, this consumer node generates a newly updated score after each transaction.

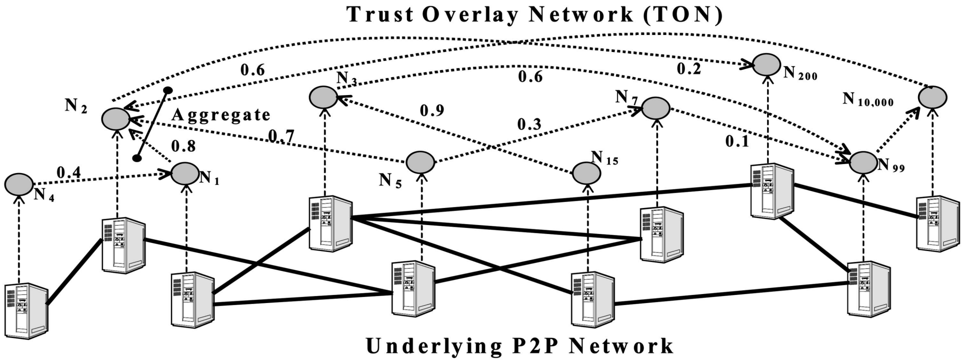


Fig. 2. A trust overlay network (TON) for a P2P system with 10,000 nodes, where a node represents a peer and an edge is labeled with the peer feedback score for the service provided. The global reputation of a peer is calculated by the weighted sum of local trust scores received on all incoming edges to that node.

Our system can incorporate different methods to generate feedback scores, such as Bayesian learning [3]. In a TON, every node keeps feedback scores for its neighbors. Because every peer has its own criteria to generate feedback scores, in our PowerTrust system, the feedbacks will be normalized to *local trust scores* defined in Section 4.3. Each node N_i is rated with a *global reputation score* v_i. This global reputation of a node is aggregated from local trust scores weighted by the global reputations of all its in-degree neighbors.

For example, the global reputation score of N_2 could be calculated below by weighting three incoming local scores (0.8, 0.7, 0.6) from N_1, N_5, and $N_{10,000}$, respectively: $v_2 = 0.8 v_1 + 0.7 v_5 + 0.6 v_{10,000}$. Given $v_1 = 0.04, v_5 = 0.0007, \textit{and}\, v_{10000} = 0.000005$, then we compute $v_2 = 0.8 \times 0.04 + 0.7 \times 0.0007 + 0.6 \times 0.0000005 = 0.032 + 0.00049 + 0.00003 = 0.032493$. It is interesting to note that node N_1 has a much higher reputation score v_1, compared with v_5 and $v_{10,000}$ in this example. Thus, node N_1 carries more weight in the global reputation aggregation process. We will consider N_1 a *power node* in the subsequent sections.

Detailed algorithms will be given in Section 4.3 to determine the global reputation scores in an iterative convergence process. In a TON, the number of users to whom a peer sends feedback scores is indicated by the *outdegree* of that node. The number of users from whom a peer receives feedback scores is represented by the *indegree* of that node. We accumulated a huge TON to study the feedback distribution in eBay reputation system, which equals the node degree distribution in the TON graph.

4 Power-Law Distribution of Peer Feedbacks

Power-law distribution is well known in the Internet community [7]. We study the public-domain eBay reputation system to verify the conjecture that the feedback distribution of a typical P2P reputation system follows the power-law. In eBay, feedback is generated after every transaction. However, nearly 90 percent seller-buyer pairs conducted just one transaction during the past five years [21]. So, the node in-degree in TON is approximated by the number of received feedbacks. Three key parameters are used: The *feedback amount* of a node i is denoted by d_i, which is the *indegree* of this node. For example, node N_2 in Fig. 1 has an in-degree of 3, meaning three feedback scores received. *Feedback frequency* f_d is the number of nodes with feedback amount d. The *ranking index* θ_d indicates the order of d in the decreasing list of feedback amounts.

4.1 Collection Procedure of eBay Reputation Data

The eBay is by far the most successful cyber-exchange platforms based on a simple reputation mechanism [21]. The eBay users provide feedbacks to a centralized reputation center and report their experiences in eBay transactions. The scoring scheme in eBay is simple: positive 1 for a good or successful transaction, negative 1 for a poor or failed feedback, and zero for a neutral or don't-care feedback. Every eBay user has a time-varying reputation by summing up all transaction scores received up to the current time.

It is difficult to collect all user feedback scores from eBay since the total number of eBay users exceeds 100 million. We apply a sampling technique to collect 108 MB feedback data over 10,000 users. We start from an arbitrary power user (a very reputable user) in eBay, who has a reputation score higher than 10,000. In order to infer the in-degree distribution in the TON, we put together a list of users to whom the power user left feedback scores from July 1999 to March 2005. Then, we extract the number of feedbacks received by each user in that list.

Apparently, the more feedback scores a peer has received from others, the easier the user is crawled. Let p_d be the probability that a node with feedback amount d is discovered by a random crawler, we have $p_d = d / \sum_{i=1}^{n} d_i$, where d_i is the received feedback by node i and n is the total number of nodes in the eBay TON. Therefore, the probability that this node is not discovered after k random crawls follows a Poisson distribution, i.e., $(1 - p_d)^k$. For a power user to issue k feedback scores, the probability of a node being crawled from the power node is estimated by (1), assuming d feedback scores received by this node.

$$Q_d = 1 - (1 - p_d)^k = 1 - \left(1 - d / \sum\nolimits_{i=1}^{n} d_i\right)^k. \qquad (1)$$

Let n_d be the initial number of nodes with feedback amount d in the eBay TON. Let $\hat{n}_d$ be the number of nodes with feedback amount d in the sample data set. We calculate $\hat{n}_d = E(n_d) \times Q_d$. So, the expected value $E(n_d) = \hat{n}_d / Q_d$. This implies that we recover n_d from Q_d and $\hat{n}_d$. This *recovery process* generates a more accurate distribution of the eBay trace data.

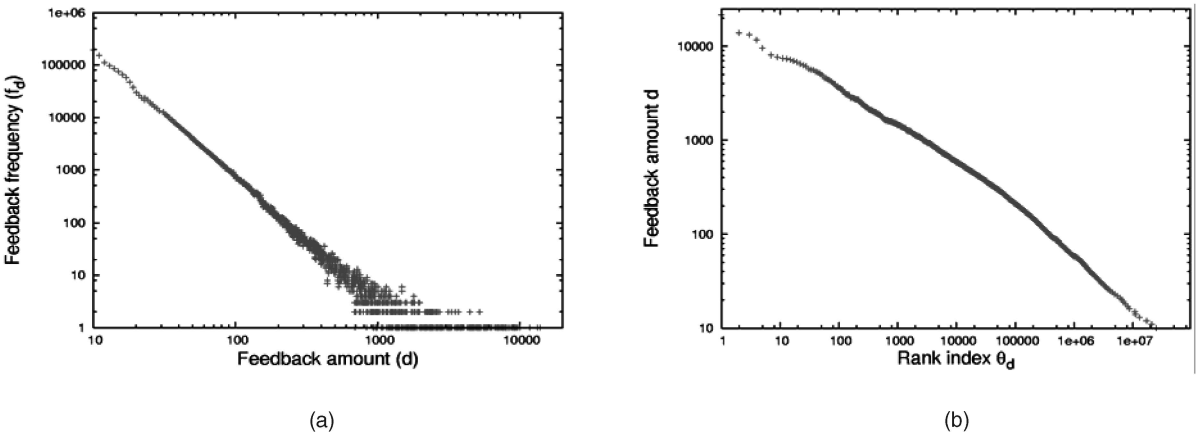
4.2 Feedback Distribution in eBay Reputation Data

Initially, we start with the sampling eBay trace of more than 11,000 users (nodes). Considering unregistered users and obsolete users, we assume that eBay has 80 million stable users out of 100 million claimed by eBay authority. The average feedback amount per user is 68 based on our trace data. We approximate the total $\sum_{i=1}^{n} d_i$ by $80,000,000 \times 68 = 5.24 \times 10^9$. We apply the recovery process specified in Section 4.1 to the eBay trace data. The feedback numbers in eBay follow the power-law distribution plotted in Fig. 3. There are more than 10,000 dots (nodes) forming the peer feedback distribution.

We plot in Fig. 3a the *feedback frequency* as a function of the *feedback amount*. This distribution shows the feedback frequency f_d is inversely proportional to the feedback amount d in log-log scale, which is approximated by (2) below. We plot in Fig. 3b the variation of the pairs (d, θ_d) using the recovered data, where θ_d is the ranking index of d in the decreasing order of the feedback amounts. The plot is approximated by a linear-regression with a correlation coefficient 0.92. In log-log scale, the feedback amount d is inversely proportional to the feedback index θ_d.

4.3 Feedback Distribution Analysis in P2P Systems

The power-law feedback distribution is resulted from two factors: *dynamic growth* of TON size and *preferential node attachment* [22]. Dynamic growth allows the network to expand freely. Preferential attachment enables the new node to interact with reputable peer nodes with higher



(a)

(b)

Fig. 3. Power-law peer feedback distribution extracted from the eBay transaction trace data over 10,000 users from July 1999 to March 2005. (a) Feedback frequency versus feedback amount. (b) Feedback amount versus rank index.

probability. Both factors are common in a P2P reputation system. The reputation system must make the global reputation scores accessible by all peers. We prove below why power-law feedback distribution applies to P2P reputation systems in general.

Theorem 1. *In a general dynamic P2P system, the received feedback amount or the indegree (N) of the associated TON graph follows the Power-law distribution specified by:*

$$Prob.\ [Indegree\,(N) = d] = c \times d^{-\beta}, \tag{2}$$

where N refers to any peer node in the TON graph, the exponent β decides the decreasing feedback frequency with respect to the increase of the feedback amount d.

Proof. Consider a new transaction taking place at time instant t. Let $X_k(t)$ be the number of users with in-degree k after t time steps. A user increases $X_k(t+1)$, if it interacts with the one with $(k-1)$ in-degree. Such a transaction has a probability $(k-1)X_{k-1}(t)/t$ to occur. On the other hand, the user decreases $X_k(t+1)$ with a probability $kX_k(t)/t$. We obtain the expected difference $E(X_k(t+1) - X_k(t)) = (k-1)X_{k-1}(t)/t - kX_k(t)/t$ between two transactions. Suppose $\Delta X_k(t)$ converges to c_k as $t \rightarrow \infty$, we have $c_k = (k-1)c_{k-1} - kc_k$. This completes the proof. □

In general, we know the range $1 \leq \beta \leq 3$. Both β and the constant c are experimental decided by traces of transaction data in a given P2P system. For the eBay trace distribution, we observed $\beta = 2.4$ by binning the data into exponentially wider bins [2]. The power-law distribution implies that the node with a few feedbacks is common, whereas the node with a large number of feedbacks is extremely rare. Therefore, only a few nodes have much higher degree than others. These nodes are selected as *power nodes* as described in Section 3.1.

5 POWERTRUST SYSTEM CONSTRUCTION

In this section, we describe methods to construct the PowerTrust system. We give details on all functional modules introduced in Fig. 1. Three construction algorithms are given below to show the initial construction, distributed ranking, and updating process of the PowerTrust system.

5.1 Look-Ahead Random Walk (LRW)

In our PowerTrust system, feedback scores are generated by Bayesian learning [3] or by an average rating based on peer satisfaction. Each node normalizes all issued feedback scores. Consider the *trust matrix* $R = (r_{ij})$ defined over an n-node TON, where r_{ij} is the *normalized local trust score* defined by $r_{ij} = s_{ij}/\sum_j s_{ij}$, and s_{ij} is the most recent feedback score that node i rates node j. If there is no link from node i to node j, s_{ij} is set to 0. Therefore, for all $1 \leq i$, $j \leq n$, we have $0 \leq r_{ij} \leq 1$ and $\forall i \sum_{j=1}^{\mathrm{n}} r_{ij} = 1$. In other words, matrix R is a stochastic matrix, in which all entries are fractions and each row sum equals 1. This demands that the scores issued by the same node to other peers are normalized.

All global reputation scores v_i for n nodes form a *normalized reputation column vector $V = (v_i)$, where $\sum_i v_i = 1$.* The reputation vector V is computed by (3), given an arbitrary initial reputation vector $V_{(0)}$ and small error threshold ε. For a system of n nodes, we can simply assume $v_i = 1/n$ to start with. For all $t = 1, 2, \ldots, k$, while $|V_{(i)} - V_{(i-1)}| > \varepsilon$, we compute the successive reputation vectors recursively by:

$$V_{(t+1)} = R^T \times V_{(t)}. \tag{3}$$

After a sufficient number of k iterations, the global reputation vector converges to the eigenvector of the trust matrix R [13]. This recursive process is motivated by the Markov random walk, which is widely used in ranking Web pages. This is similar to a random knowledge surfer hopping from nodes to nodes to search for a reputable node. At each step, the surfer selects a neighbor according to the current distribution of local trusts. The stationary distribution of the Markov chain is the converged global reputation vector.

We propose a *look-ahead random walk* (LRW) strategy to efficiently aggregate global reputations. Each node in the TON not only holds its own local trust scores, but also aggregates its neighbors' first hand ones. Compared to regular random walk, the surfer makes the decision based on knowledge from itself and all neighbors. The extra aggregation overhead grows linearly in sparse power-law graphs [17]. This is not true for random graphs.

TABLE 2
Speedup Factors of Using Look-Ahead Random Walk Strategy in Random Graphs and Power-Law Graphs

TON Size	Random Graph	Power-law Graph
1000	1.87	2.14
3000	1.93	1.95
5000	1.84	2.21
7000	1.98	2.17
9000	1.95	2.08

The efficiency of the LRW strategy is analyzed below. Each peer node aggregates the first-hand local trust scores from its neighbors, the *enhanced trust matrix S* by using the LRW strategy is computed by $S = R^2$. Define a *speedup factor* by comparing the number of convergence iterations for a regular random walk to that of LRW. Table 2 shows the speedup factor for various graph sizes. We generated 100 random graphs and 100 Power-law graphs to make the comparison. The node degree distribution of a random graph is specified by:

$$Prob.\ [Indegree\,(N) = d] = \binom{n-1}{d} p^d (1-p)^{n-d-1}, \qquad (4)$$

where N is an arbitrary node, n is the graph size, and $p = (\text{Number of links})/n^2$. As shown in Table 2, the LRW strategy greatly improves the convergence rate in both Power-law graph and random graph. The Power-law graph has higher speedup in all network sizes. The improvement comes from the random walker in a power-law graph can quickly hop towards highly reputable nodes, which preserve a lot of useful reputation information.

5.2 Distributed Ranking Mechanism

A distinction of our PowerTrust system is to leverage mainly the power nodes to aggregate the global reputations. However, in a large P2P system with frequent peer joining and leaving, we could not assume that there always exist some static and predetermined power nodes. Instead, we propose a fully distributed ranking mechanism to select the m most reputable power nodes, dynamically. The process to find the m most reputable nodes is described in Algorithm 1.

PowerTrust uses a *Distributed Hash Table* (DHT) such as Chord [31] to implement the distributed ranking mechanism. As in EigenTrust [13], every node has a score manager that accumulates its global reputation. When a new node i joins the system, node j is assigned as the score manager of node i if node j is the successor node of k_i, where k_i is the hash value of the unique identifier of node i by a predefined hash function. All other nodes can access the global reputation of node i by issuing a lookup request with key equal to k_i. Different hash functions can be used to have multiple score managers for each node in case the malicious score manager reports some wrong global reputation scores.

To select the m most reputable nodes, our distributed ranking mechanism applies *locality preserving hashing* (LPH) [4] to sort all nodes with respect to their global scores. Hash function H is a locality preserving hash function if it has the following two properties: 1) $H(v_i) < H(v_j)$, iff $v_i < v_j$, where v_i and v_j are the global reputations of node i and j, respectively, and 2) if an interval $[v_i, v_j]$ is split into $[v_i, v_k]$ and $[v_k, v_j]$, the corresponding interval $[H(v_i), H(v_j)]$ must be split into $[H(v_i), H(v_k)]$ and $[H(v_k), H(v_j)]$.

Algorithm 1: Selection of top-m peers (Power nodes)
Input: global reputations stored among score managers
Output: m most reputable nodes
Procedure:

```
for each score manager $j$, suppose it is the score manager of
    node $i$ do
    hash reputation value $v_i$ to a hash value $H(v_i)$ using a
    LPH function
    insert the triplet $(v_i, i, j)$ to the successor node of $H(v_i)$.
end for
initialize node $x =$ successor node of the maximum
                        hash value
Set $p =$ the number of triplets with highest reputation values
        stored in node $x$
  loop: if $p > m$ then return;
            else
            node $x$ sends a message to its predecessor
            node $y$ to find the
            next $m - p$ highest reputation triplets
            node $x =$ node $y$
            $m = m - p$
            $p =$ number of triplets stored in node y
            goto loop
        end if
```

Suppose node j is the score manager of node i, it stores a pair (v_i, i) for node i, where v_i is the global reputation of node i. Node j hashes the reputation value v_i using a LPH function to a hash value $H(v_i)$ and inserts the triplet (v_i, i, j) to the successor node of $H(v_i)$. The triplets are stored in the ascending order of their reputation values in the DHT hash space due to the property of LPH. Assume node x is the successor node of the maximum hash value and it stores k triplets with highest reputation values. If k is less than m, node x sends a message to its predecessor node y to find the next $m - k$ highest reputation triplets. This process repeats recursively until the m highest reputation triplets are found.

Basically, distributed reputation ranking requires two different hash overlays. One assigns peers to their score managers and another ranks the peers by their global reputation scores. Fig. 4 presents a 5-node PowerTrust system built on top of a Chord with 4-bit circular hash space. Node N_{15} is the score manager of node N_2 whose global reputation is 0.2. Node N_{15} hashes the reputation value 0.2 using a simple LPH function $H(x) = 32x$. The resulting hash value is 6.4. Node N_{15} sends out $Sort\_Request\{key = 6.4, (0.2, N_2, N_{15})\}$ message, which is routed to node N_8. Node N_8 stores the triplet (0.2, N_2, N_{15}), since it is the successor node of hash value 6.4. For simplicity, we illustrate below how to find the highest reputation node with $m = 1$.

Node N_2 is the successor node of the maximum hash value 15, so node N_2 initiates the process to find m power nodes. Node N_2 is responsible for the hash values in the union of two ranges (15, 16] ∪ [0, 2]. Since it has no corresponding triplets within the range (15, 16], it stores zero triples with highest reputation values, i.e., $k = 0$. Therefore, it sends a $Top\_m\_Request(m = 1, k = 0)$ message

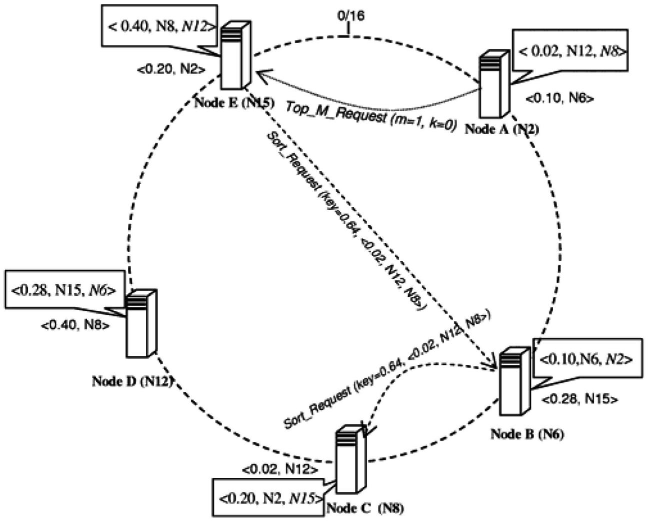


Fig. 4. Distributed reputation ranking using the locality-preserving hash function over a DHT-based P2P system with five peer nodes.

to its predecessor node N_{15}, which finds its stored triplet with value 0.4 being the highest one. So, node N_8 is the most reputable node in this example system. Multiple LPH functions could be used to prevent cheating by the malicious peers.

The following theorem guarantees that Algorithm 1 always produces the top-m reputation values in h hops, where h is the number of nodes between $Successor(H(v_k))$ and $Successor(2^b - 1)$, and v_k is the mth highest reputation score.

Theorem 2. *If we use a locality preserving hash function H to map reputation score v into the b-bit Chord circular space $[0, 2^b - 1)$, the nodes that store the top m largest reputation scores must have an identifier located between the $Successor(H(v_k))$ and $Successor(2^b - 1)$ nodes.*

Proof. Let $V = \{v_i | 0 \leq i < m\}$ be m highest reputation scores, we have $v_k \leq v_i \leq v_{max}$, where v_{max} is the highest score. Using a LPH function H, we map the score v_k to $successor(H(v_k))$, where $H(v_k) \leq H(v_i) \leq H(v_{max})$. Since all hashed values within the identifier space $[0, 2^b - 1)$, $H(v^{max})$ must not exceed $2^b - 1$. Therefore, we have $H(v_k) \leq H(v_i) \leq 2^b - 1$. Since $successor(H(v_k))$ is the first node which follows the identifier of $H(v_k)$, the reputation score v_i is assigned to the nodes between $successor(H(v_k))$ and successor$(2^b - 1)$. So, the distributed ranking module guarantees to find the m most reputable nodes by traversing nodes from $successor(2^b - 1)$ to $successor(H(v_k))$. □

5.3 Initial Global Reputation Aggregation

Algorithm 2 specifies the initial round of global reputation aggregation. Each node i sends all local trust scores to the score managers of its out-degree neighbors. Let λ_1 and λ_2 be the largest and the second largest eigenvalue of the trust matrix R defined over the TON. The Power-law property of a TON leads to a tight bound on the ratio λ_2/λ_1 [10]:

$$1 - \Omega(1/\log n) < \lambda_2/\lambda_1 < 1 - \Omega(1/\log^2 n). \qquad (5)$$

Because of the obvious gap between λ_1 and λ_2, the power-law feedback distribution of TON guarantees the convergence at the very first round of global reputation aggregation.

Algorithm 2: Initial Global Reputation Aggregation

```
Input: Local trust scores stored among nodes
Output: Global reputation for every node
Procedure:
for each node $i$ do
    forall node $j$, which is an out-degree neighbor of node $i$
    do
        Send the score message $(r_{ij}, i)$ to the score manager of
        node $j$
    end forall
    if node $i$ is the score manager of node $k$, then
    forall node $j$, which is an in-degree neighbor of node $k$
    do
        Receive the score message $(r_{jk}, j)$ from node $j$
        Locate the score manager of node $j$
        end forall
    Set a temporary variable $pre = 0$; initialize the error
    threshold $\varepsilon$
    and global reputation $v_k$ of node $k$
    Repeat
        Set $pre = v_k$; $v_k = 0$
        Forall received score pair $(r_{jk}, j)$, where $j$ is an
        in-degree neighbor of node $k$ do
            Receive the global reputation $v_j$ from the score
            manger of node $j$
            $v_k = v_k + v_j r_{jk}$
        end forall
        Compute $\delta = |v_k - pre|$ until $\delta < \varepsilon$
  end if
end for
```

Theorem 3. *Given a small error threshold ε and the ratio $b = \lambda_2/\lambda_1$, the number of iterations in Algorithm 2 is upper bounded by the smallest integer k such that*

$$k = \lceil \log_b \varepsilon \rceil. \qquad (6)$$

Proof. Consider a trust matrix R with m eigenvectors $x_1, x_2, \ldots, x_m$ and corresponding eigenvalues $\lambda_1 > \lambda_2 >, \ldots, > \lambda_m$. The initial reputation vector y is $\sum_{i=1}^{m} b_i x_i$. We have $Ry = \sum_{i=1}^{m} b_i R x_i = \lambda_1 (b_1 x_1 + \sum_{i}^{m=2} (\lambda_i/\lambda_1) b_i x_i)$ and $Rjy = \lambda_1^j (b_1 x_1 + \sum_{i=2}^{m} ((\lambda_i/\lambda_1)^j b_i x_i)$. Since $(\lambda_2/\lambda_1)^k = \varepsilon$, we have $(\lambda_i/\lambda_1)^k < \varepsilon$, where $2 < i \leq m$. Therefore, $R^k y = \lambda_1^k b_1 x_1 + o(\varepsilon)$. Algorithm 2 is thus converged with repeated application of R to y in k steps, where k is defined in (6). □

5.4 Global Reputation Updating Procedure

After first round aggregation, the score managers collaborate with each other to find the power nodes using Algorithm 1. If node x stores the triplet (i, v_i, j) and finds i a power node, node x will notify node j. Because the trust matrix R is dynamically changing with new peers joining and new transactions performed, the global reputation scores should be updated periodically, especially for power nodes. The updating of global reputation aggregation leverages the use of the power nodes.

TABLE 3
Parameters and Their Default Values Used in Simulation Experiments

Parameter	Basic Definition	Default Value
n	Number of initial peers (nodes) in a P2P system or P2P Grid	1000
β	Feedback factor for specifying power-law distribution	2.4
α	Greedy factor of a peer (user) as a random walker	0.15
$d_{\max}$	Maximum peer feedback amount (max node degree)	200
γ	Percentage of malicious peers in a P2P system	20%
m	Maximum number of power nodes in a P2P systems	1%
ε	Threshold for global reputation convergence	10^{-4}

The reputation updating process is specified in Algorithm 3. Our PowerTrust scheme works as random walks along a Markov chain. The random surfer starts its journal on any node with the same probability. We define a *greedy factor* α as the eagerness probability of the surfer jump directly to the power node. The higher is the value of α, the keener the surfer wants to attach to a power node.

At any given node, the surfer selects a neighbor according to the local trust distribution with a probability $1 - \alpha$. With a probability α, the surfer attaches itself with a power node. The power nodes are re-elected based on new global reputation score after each round of aggregation. We can adjust the *greedy factor* α to control the gap between the first and second largest eigenvalues of a transition matrix T, because largest eigenvalue $\lambda_1 = 1$ and the second largest eigenvalue $\lambda_2 \leq 1 - \alpha$ as proved in [19].

Algorithm 3: Global Reputation Updating Procedure

```
  Input: Local trust scores stored among nodes
  Output: Global reputation scores for all nodes for use by
  score managers collaboratively to find
      the $m$ most reputable nodes using Algorithm 1
  Procedure:
  for each node $i$ do
    forall node $j$, which is an out-degree neighbor of node $i$
    do
      Aggregate local trust scores from node $j$
      Send the score message $(r_{ij}, i)$ to the score manager of
      node $j$
    end forall
    If node $i$ is the score manager of node $k$, then
      forall node $j$, which is an in-degree neighbor of node $k$
      do
        Receive the score message $(r_{jk}, j)$ from node $j$
        Locate the score manager of node $j$
      end forall
      Set a temporary variable $pre = 0$; initialize the error
      threshold $\varepsilon$ and global reputation $v_k$ of node $k$
      repeat
        Initialize $pre = v_k$; $v_k = 0$
        forall received score pair $(r_{jk}, j)$, where $j$ is an
        in-degree neighbor of node $k$ do
            Receive node $j$ global reputation $v_j$ from score
            manager of node $j$
        end forall
      if node $k$ being a power node,
          then $v_k = (1 - \alpha) \sum(v_j \times r_{jk}) + \alpha/m$
          else $v_k = (1 - \alpha) \sum(v_j \times r_{jk})$
      end if
      compute $\delta = |v_k - pre|$, until $\delta < \varepsilon$
    end if
  end for
```

6 System Performance Analysis

The performance of the PowerTrust system is analyzed below in terms of *reputation convergence overhead*, *ranking discrepancy*, and *aggregation errors* by malicious peers.

6.1 Simulation Setup and Experiments Performed

Three sets of simulated P2P experiments were performed. We use the *convergence overhead* to measure the aggregation speed. We use peer dynamics to enable system scalability. We use *ranking discrepancy* to measure the accuracy and *RMS aggregation error* to quantify the system robustness to malicious peers. Our simulation experiments were implemented on a dual-processor Dell server. Each data point represents the average of at least 10 simulation runs.

Simulation parameters and default values used in the experiments are summarized in Table 3. Our simulated TON for a P2P system was a fully connected Power-law graph, consisting of 1,000 nodes initially with a maximum node degree $d_{\max} = 200$ and a feedback factor $\beta = 2.4$. We assume 80 percent honest peers and 20 percent malicious peers in the simulated P2P system.

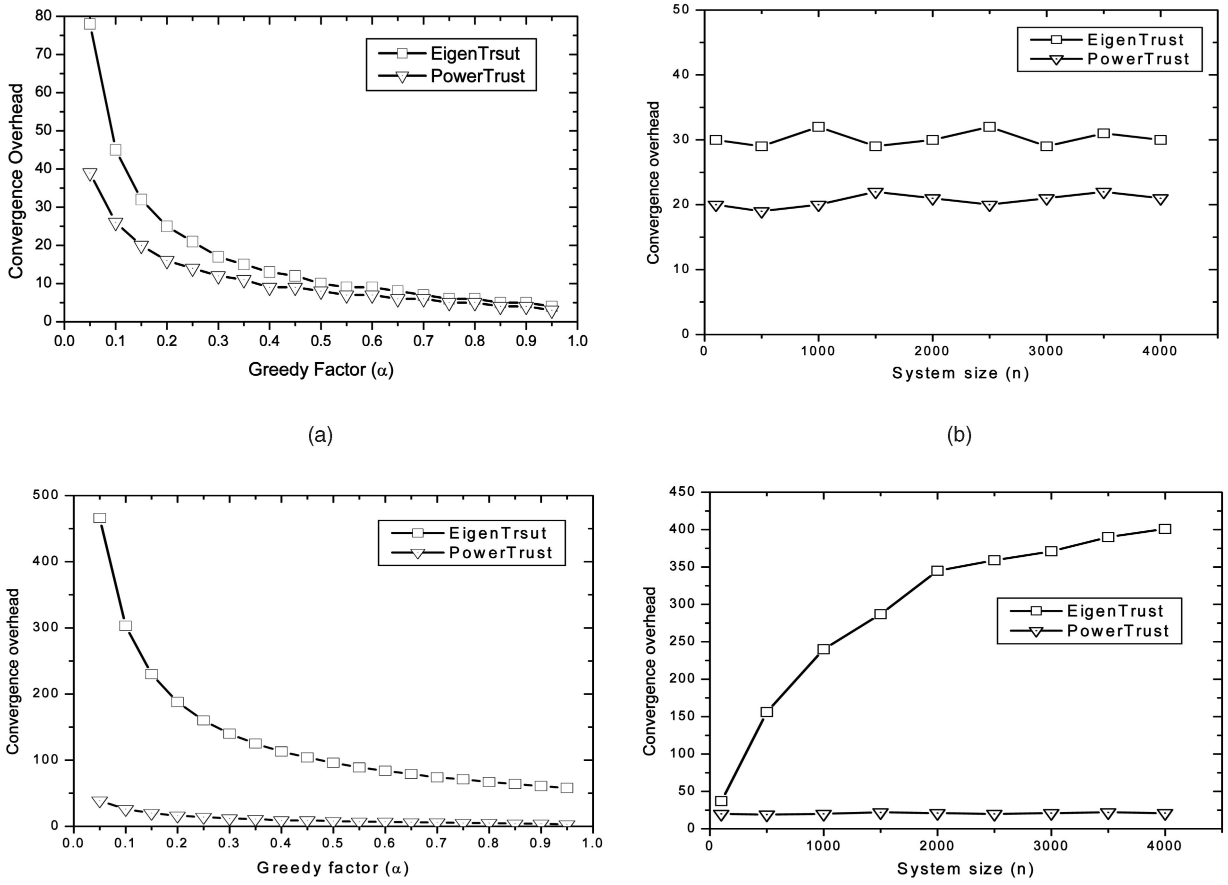
We model two types of malicious behaviors: One type reports dishonest trust scores (such as reporting low trust scores for good peers and vice versa). Another type of abusers collaborates with each other to boost up their own ratings. They may rate the peers in their collusion group very high and rate outsiders very low. The system selects up to 1 percent of the total number of nodes in a TON as the power nodes.

Table 4 shows the relationship between α, convergence overhead (defined in Section 6.2) and ranking discrepancy (specified in Section 6.3) in a 1,000-node P2P reputation system under two network conditions: without any malicious peers and with 20 percent malicious peers. When there is no malicious peer in the system, as α increases, there is a trade-off between convergence overhead and ranking discrepancy. With 20 percent malicious peers, the ranking discrepancy first decreases then increases, as α increases. So, we choose $\alpha = 0.15$ as a default value to balance the trade-off between efficiency and accuracy.

TABLE 4
Relationship among α, Convergence Overhead, and Ranking Discrepancy in a PowerTrust Reputation System over 1,000 Peers

Without malicious peers		
Greedy factor α	Convergence overhead	Ranking Discrepancy
0	82	0.0
0.05	39	0.123
0.10	25	0.167
0.15	19	0.215
0.20	15	0.237
0.30	13	0.313
0.45	10	0.357

With 20% malicious peers		
Greedy factor α	Convergence overhead	Ranking Discrepancy
0	79	0.275
0.05	40	0.155
0.10	24	0.201
0.15	20	0.217
0.20	15	0.234
0.30	12	0.253
0.45	10	0.331



(a)

(b)

(c)

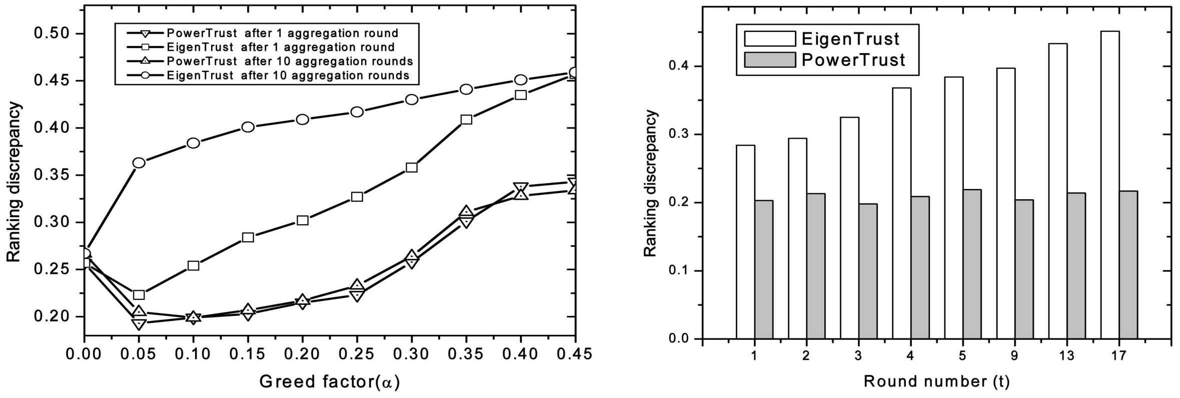
(d)

Fig. 5. Convergence overhead of two reputation systems under variable peer greedy factor and increasing P2P system sizes. (a) Disallowing departure of power nodes in PowerTrust or pretrusted nodes in EigenTrust system. (b) Disallowing departure of power nodes or pretrusted nodes with a fixed $\alpha = 0.15$. (c) Allowing departure of power nodes in PowerTrust or pretrusted nodes in EigenTrust system. (d) Effect of system size n with departure of power nodes or pretrusted nodes under a fixed $\alpha = 0.15$.

6.2 Reputation Convergence Overhead

The *convergence overhead* is measured as the number of iterations before the global reputation convergence. As indicated in Section 4, convergence means that distance between two consecutive reputation vectors is smaller than the threshold. The EigenTrust approach relies on a few pretrusted nodes to compute the global reputations. They assumed that some peers are known trustworthy, essentially among the very first few peers joining the system. This assumption may not agree with the reality of a decentralized P2P computing. We randomly choose some reputable nodes as pretrust nodes in our simulations. We report in Fig. 5 the effects of different greedy factor α and system sizes n on the variation of the convergence overhead.

For all fairness, we choose the same number of power nodes equal to that of pretrusted nodes used in EigenTrust. Figs. 5a and 5b shows the convergence overheads for two reputation systems, assuming no pretrusted node or power node leaving the P2P network. We observe the slight saving of iteration count in PowerTrust as shown in Fig. 5a. The overhead drops to the same level as α increases toward 1. Fig. 5b shows small fluctuation of the convergence overhead as the system size increases. In the case of a low $\alpha = 0.15$, we see an approximately 50 percent reduction in



(a)

(b)

Fig. 6. Ranking discrepancy percentage of two reputation systems with 1,000 nodes initially under variable peer greedy factor and aggregation round number. (a) Effects of greedy factor after one round and 10 rounds of aggregation. (b) Effect of aggregation round number under a fixed $\alpha = 0.15$.

convergence overhead in using PowerTrust over EigenTrust system. The overheads in both systems are only moderately sensitive to the variation in network size.

In Figs. 5c and 5d, the power nodes in PowerTrust and the pretrusted node in EigenTrust are allowed to leave freely. These two plots show significant widening of the overhead gap between the two systems. We observe a sharp drop of iteration count in using PowerTrust to a flat small number less than 50 in Fig. 5c, when α increases from 0.15 to 1, while the EigenTrust still requires more than 100 iterations to converge. Fig. 5d shows that our PowerTrust system has almost a flat low convergence overhead, independent of the system size under the default value of $\alpha = 0.15$. The EigenTrust system overhead can reach as high as 400 iterations as the system increases to 4,000 nodes.

In both plots, the PowerTrust system outperforms the EigenTrust system sharply. The EigenTrust system converges very slowly. The system cannot guarantee its convergence, when the pretrusted nodes are allowed to leave the system freely. In the PowerTrust system, the power nodes are re-elected after each aggregation round. Based on the distributed ranking mechanism, the score managers of the departing power nodes notify the system to replace them timely with other more qualified power nodes. The decrease of computation overhead means significant traffic reduction on the network, and less work for all peers involved. The low overhead in using the PowerTrust system makes it attractive in performing highly scalable P2P applications, including P2P Grids as reported in [38].

6.3 Reputation Ranking Discrepancy

To estimate the accuracy of the aggregated global reputation, we rank the peers by their global reputation scores. We measure below the *ranking discrepancy* between the estimated ranking and the actual ranking. The discrepancy comes mainly from greedy factor α and malicious peers reporting false trust scores. We use *normalized Euclidean distance* [9] to measure the ranking discrepancy. During each round of reputation aggregation, we assume 100 new peers joining the system and transacting with existing peers. We refer each aggregation round to one full convergence of reputation vector computations.

The probability of an interaction between nodes i and j is determined by the ratio $d_i d_j / \sum_{k=1}^{n} d_k$, where d_i and d_j are the corresponding node degrees. This property ensures that the growing TON follows the power-law connectivity [10]. Fig. 6 shows the ranking discrepancy between the actual and estimated rankings as a function of the greedy factor and aggregation rounds respectively.

The result is plotted in Fig. 6a after the first round and 10th round of global reputation aggregation. After the first round, both pretrusted nodes in EigenTrust and power nodes in PowerTrust system can reduce the effects of malicious nodes slightly, when α is very small ($\alpha < 0.05$). When α is larger than 0.05, PowerTrust has about 50 percent less ranking discrepancy than that of EigenTrust. PowerTrust discrepancy is independent of the number of aggregation rounds.

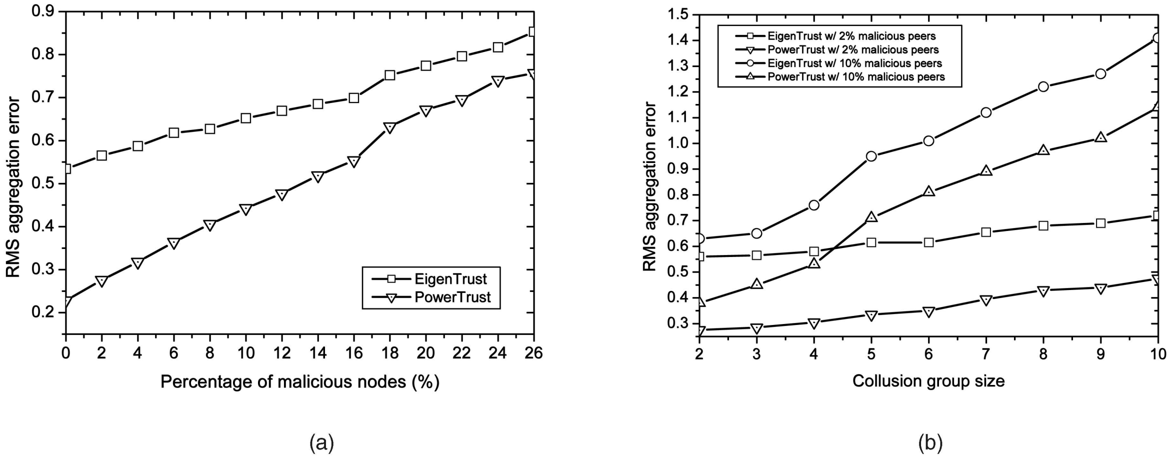
Fig. 6b shows the aggregation effect with $\alpha = 0.15$, the ranking discrepancy of EigenTrust increases from 28 percent to 44 percent with the increase of round number, while PowerTrust always maintains the low discrepancy at about 20 percent. This accuracy improvement shows that PowerTrust updates reputation scores more accurately than EigenTrust. The main reason is that our power nodes are the most reputable nodes, which are dynamically chosen after each aggregation round, while the pretrusted nodes are statically chosen in EigenTrust, regardless of their sustained performance.

6.4 Effects of Malicious Peer Behaviors

We evaluate the effectiveness and robustness of the PowerTrust system against various malicious peer behaviors. The experiment was performed under both noncollusive and collusive malicious settings. We compute the *root-mean-square* (RMS) of the aggregated global reputation of all peers. A lower RMS error indicates higher accuracy. The RMS error is defined by:

$$RMS\ aggregation\ error = \sqrt{\frac{\sum((v_i - v_i')/v_i)^2}{n}}, \qquad (7)$$

where v_i and v_i' are the actual and measured global reputation scores of peer i, respectively.



(a)

(b)

Fig. 7. Global reputation aggregation errors from fake trust scores reported by malicious peers in two P2P reputation systems over 1,000 nodes initially. (a) Independent malicious peers. (b) Collusive malicious peers.

We plot the RMS error against the percentage of malicious peers in Fig. 7a. The default greedy factor $\alpha = 0.15$ was assumed. The probability of a node being malicious is modeled by the inverse of its global reputation, because a node providing corrupted services is highly likely to issue dishonest scores. Fig. 7a shows the RMS aggregation error incurred by malicious peers reporting false local trust scores, independently. With 2 percent malicious peers, the PowerTrust system has 53 percent less aggregation error than that of the EigenTrust system. As the percentage increases, the error gap is closing up between the two systems.

In Fig. 7b, 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 rate each other high in the same group and rate outsiders very low. In all cases (2 percent and 10 percent malicious peers), the PowerTrust shows its robustness against collusive peer groups of various sizes. The EigenTrust system is less resistant to abuses by large collusive peer groups.

7 P2P Application Benchmark Results

In this section, we show two simulated P2P application performance results in using PowerTrust to aggregate peer reputations. One application is distributed file sharing among the peers and the second is distributed P2P supercomputing over the benchmark of *parameter sweeping applications* (PSA), often used in Grid evaluation experiments [28].

7.1 Query Success Rate in Distributed File Sharing

We have applied the PowerTrust system on simulated P2P file-sharing applications. We choose the same query model used by Marti and Garcia-Molina [15]. There are more than 100,000 files in our simulated P2P systems. The number of copies of each file in the system is determined by a content Power-law distribution with $\beta = 1.2$. Each peer is assigned with a number of files based on the Sarioiu distribution [24]. At each time step, a query is randomly generated at a peer and completely executed before the next query/time step. The query distribution determines which file each query search for. We rank the queries according to their popularity.

We use a query Power-law distribution with $\beta = 0.63$ for queries ranked 1 to 250 and $\beta = 1.24$ for the remaining lower ranking queries. When a query for a file is issued, the list of nodes having this file is generated and the one with the highest global reputation is selected to download the desired file. Fig. 8 shows the query success rates in using the PowerTrust and EigenTrust reputation systems, separately.

The query success rate is measured by the percentage of successful queries over the total number of queries issued. Every node may respond a query with inauthentic files. For simplicity, this behavior is modeled as inversely proportional to the node's global reputation. We consider both cases of allowing or disallowing power nodes or pretrusted nodes to leave the system. We also consider the case of a *no-trust system,* meaning no trust management in the P2P system. The no-trust system randomly selects a node to download the file without considering reputation.

Fig. 8a shows the results without the departure of power nodes or pretrusted nodes. There are 1,000 queries issued after each round of global reputation aggregation. The query success rate of PowerTrust is maintained at 90 percent level after just one round of reputation aggregation. The query success rate of EigenTrust drops from 85 percent to 50 percent as the round number increases. This is due to the fact that pretrusted nodes cannot cope with the dynamic variation of the peer reputations.

In Fig. 8b, the PowerTrust has a steady query success rate higher than 90 percent after only one round of aggregation. Allowing the pretrusted nodes to leave freely in the EigenTrust system makes the query success rate even worse only after first rounds of reputation aggregation. EigenTrust has a query success rate up to 65 percent, only slightly higher than that of using the no-trust system. EigenTrust depends on the stability of pretrusted nodes. We avoided this restriction. So, PowerTrust is more scalable and robust in this sense.

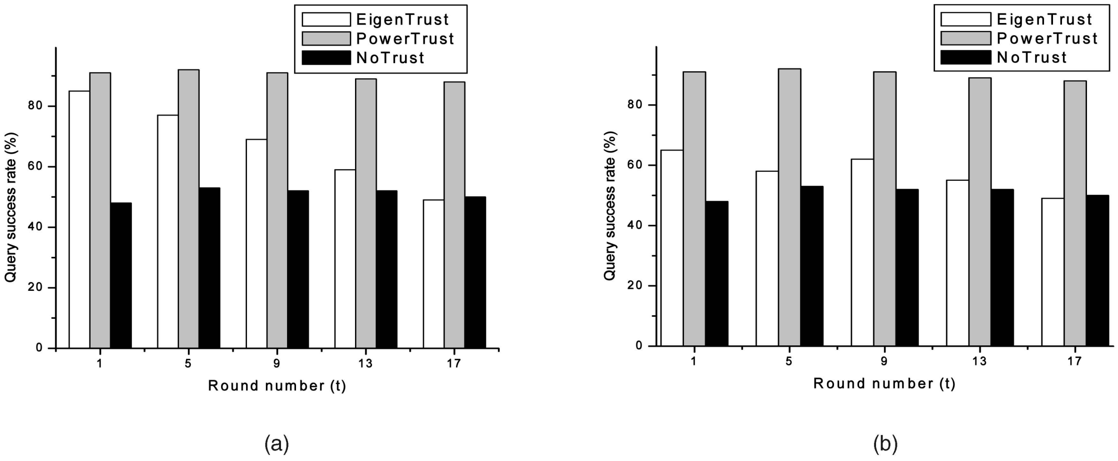


Fig. 8. Query success rates of two P2P reputation systems: PowerTrust versus EigenTrust under various rounds of reputation aggregation. (a) Effects of round number without departure of power nodes or pretrusted nodes $(\alpha = 0.15)$. (b) Effects of round number allowing the departure of power nodes or pretrusted nodes $(\alpha = 0.15)$.

7.2 P2P Grid Performance over the PSA Workload

In this section, we use two metrics to evaluate the PowerTrust performance in P2P Grid job execution over the PSA workload [28]:

1. *Makespan*: Denote the total number of simulated jobs as M and denote the completion time of a single job J_i as c_i, the makespan is defined by $Makespan = Max\{c_i\}$, where $i = 1, 2, \ldots M$.
2. *Job failing rate:* Job execution may fail at low reputation sites. M_{fail} counts the number of failed jobs. The job failing rate is defined by the ratio $F_{rate} = M_{fail}/M$.

We apply a realistic PSA workload of 20,000 to 80,000 jobs in the simulated experiments over a large-scale P2P Grid, consisting of 4,000 resource sites. The PSA benchmark runs independent jobs. A range of scenarios and parameters are applied to the input data subsets to generate parallel results. The execution model essentially involves parallel execution of M independent jobs on N distributed sites, where M is much greater than N.

A heuristic Min-Min algorithm is used for Grid job scheduling. Per each job, the Grid sites having the shortest *expected time-to-completion* (ETC) is selected. We compute the ETC $= real\_etc/(1 - fail\_rate)$, where the $real\_etc$ is the actual ETC of the Grid site and the $fail\_rate$ is the failing rate experienced with the Grid site, which is determined by the site's global reputation. Then, the job with the minimum ETC is selected and assigned to the selected Grid site. After each job execution, the Grid site will update the local trust scores of other sites according to job execution result. Therefore, the edges on the TON are relabeled with new scores, periodically.

A job will be executed if it has not been rejected more than three times. Fig. 9 shows the performance results of four different reputation systems over the PSA workload. The *NoTrust* in black bars corresponds to the worst case that the Grid site reputations are not considered in job scheduling. The *IdealTrust* in dark-gray bars corresponds to the ideal situation, where all Grid site global reputation scores are accessible. The light-gray and white bars

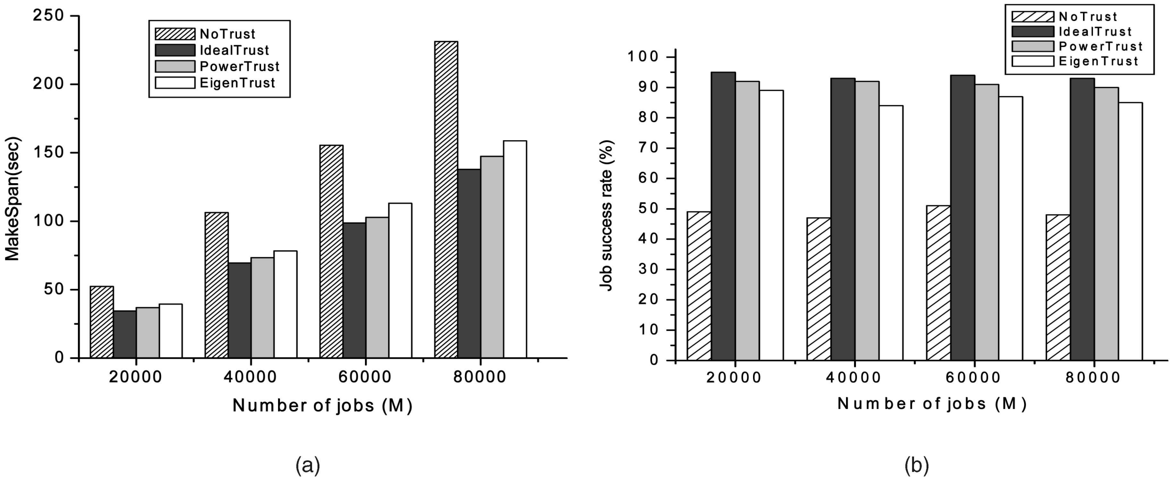


Fig. 9. PSA benchmark performance results on a simulated P2P Grid configuration of 4,000 peer-contributed resource sites (nodes). (a) Job Makespan in second. (b) Average job success rate.

correspond to using the PowerTrust and EigenTrust systems, respectively.

Fig. 9a plots the *job makespan* (completion time) of the PSA workload executed under four P2P reputation systems. They all increase linearly with respect to the increase of the job number. Fig. 9b shows the *average job success rate,* which drops slowly with the workload size. As predicted, the *NoTrust* has the longest makespan and lowest job success rate. In all cases, PowerTrust slightly outperforms EigenTrust by about 3 percent and PowerTrust is 3 percent to 6 percent lower than the optimal performance of the IdealTrust system. Without trust, the job makespan increases more than 40 percent and the job success rate drops by 45 percent, compared with the idealTrust case. These results prove the effectiveness of using global reputation to establish trust among the participating peers in a large-scale P2P Grid system.

8 Conclusions and Further Work

In this paper, we report the design experiences and simulated performance of a new P2P reputation system, PowerTrust. Specifically, our contributions are summarized in four aspects:

1. ***Power-law distribution of peer feedbacks***: We developed a trust overlay network model for analyzing the feedback properties of P2P reputation systems. By collecting real-life data from eBay, we confirmed the power-law connectivity in TON graph. This power-law distribution is not restricted to eBay reputation system. Our mathematic analysis justifies its applicability to general dynamic P2P systems.
2. ***Fast reputation aggregation, ranking, and updating***: Our PowerTrust system offers the very fast mechanisms for global reputation aggregation, ranking, and updating. Besides leveraging power-law peer feedbacks, we utilize *look-ahead random walk* (LRW) *strategy* and *locality preserving hash* (LPH) functions, which are easily implemented in a DHT-based P2P system.
3. ***System scalability and wide applicability:*** PowerTrust is applicable to P2P systems in general and to P2P Grids in particular. These are attractive to cope with dynamic growth of both P2P systems and collaboration Grid built with distributed peer resources.
4. ***System robustness and operational efficiency***: The robustness is resulted from curtailing malicious peers. The system is resilience to peer abuses in global reputation evaluation. The operational efficiency comes mainly from the use of reliable power nodes in PowerTrust.

For further work, we suggest the following research tasks to solve the peer collusion problem, to extend the current PowerTrust system to work on unstructured P2P system as well, and to explore new killer P2P applications supported by reputation systems:

1. ***Coping with peer abuses and selfishness:*** Various malicious behavior models should be investigated to secure P2P system applications. New mechanisms are needed to deal with intrusions, free riders, black mouths, collusions, and selfishness of peers [12], [14], [37]. Game theoretic studies and benchmark studies are recommended.
2. ***Reputation system for unstructured P2P System:*** PowerTrust, EigenTrust, and PeerTrust are all based on a DHT overlay network. However, most P2P systems deployed on the Internet are unstructured. Developing an efficient reputation system is even in greater demand for unstructured P2P networks. Without a fast searching or hashing mechanism, how to perform fast reputation aggregation is a major challenge in unstructured P2P systems. Our continued effort is focused on a gossip-based mechanism to solve this problem.
3. ***Explore new killer P2P applications:*** We need to explore new P2P applications for both structured and unstructured P2P systems. Most current P2P applications do not have strong collaboration among the users, except the pair of interacting parties. In particular, collaboration in P2P Grid applications should be explored [38].

Acknowledgments

This work was supported by US National Science Foundation ITR Grant ACI-0325409 at USC Internet and Grid Research Laboratory. The authors appreciate valuable suggestions from Dr. Ricky Kwok of Hong Kong University, Dr. Jianping Pan of the University of Victoria, and their colleague Min Cai. They would also like to thank the associate editor and the anonymous reviewers for their insightful suggestions to improve the quality of this paper.

References

[1] K. Aberer and Z. Despotovic, "Managing Trust in a Peer-2-Peer Information System," *Proc. 10th Int'l Conf. Information and Knowledge Management,* 2001.

[2] L.A. Adamic, "Zipf, Power-Laws and Pareto—A Ranking Tutorial," http://www.hpl.hp.com/research/idl/papers/ranking/ranking.html, HP Labs, Calif., 2002.

[3] S. Buchegger and J.-Y.L. Boudec, "A Robust Reputation System for P2P and Mobile Ad-hoc Networks," *Proc. Second Workshop Economics of P2P Systems,* June 2004.

[4] M. Cai, M. Frank, and P. Szekely, "MAAN: A Multi-Attribute Addressable Network for Grid Information Services," *J. Grid Computing,* vol. 2, no. 1, pp. 3-14, 2004.

[5] C. Dellarocas, "Analyzing the Economic Efficiency of eBay-Like Online Reputation Reporting Mechanisms," *Proc. Third ACM Conf. E-Commerce,* 2001.

[6] D. Dutta, A. Goel, R. Govindan, and H. Zhang, "The Design of a Distributed Rating Scheme for Peer-to-Peer Systems," *Proc. First Workshop Economic Issues in P2P Systems,* June 2003.

[7] M. Faloutsos, P. Faloutsos, and C. Faloutsos, "On Power-Law Relationship of the Internet Technology," *Proc. ACM SIGCOMM '99,* pp. 251-262, Aug. 1999.

[8] I. Foster and A. Iamnichi, "On Death, Taxes, and Convergence of P2P and Grid Computing," *Proc. Second Int'l Workshop Peer-to-Peer Systems (IPTP3 '03),* Feb. 2003.

[9] G. Fox et al. "Peer-to-Peer Grids," *Grid Computing,* chapter 18, Berman, Fox, and Hey, eds., John Wiley & Sons, 2003.

[10] C. Gkantsidis, M. Mihail, and A. Saberi, "Conductance and Congestion in Power Law Graphs," *Proc. ACM/IEEE SIGMETRICS,* June 2003.

[11] D. Hughes, G. Coulson, and J. Walkerdine, "Free Riding on Gnutella Revisited: The Bell Tolls?" *IEEE Distributed Systems Online,* vol. 6, June 2005.

[12] K. Hwang, Y.K. Kwok, S. Song, M. Cai., and Y. Chen, "Security Binding and Worm/DDoS Defense Infrastructure for Trusted Grid Computing," *Int'l J. Critical Infrastructures,* vol. 2, no. 4, 2005.

[13] S. Kamvar, M. Schlosser, and H. Garcia-Molina, "The Eigentrust Algorithm for Reputation Management in P2P Networks," *Proc. ACM World Wide Web Conf. (WWW '03),* May 2003.

[14] K. Kwok, K. Hwang, and S. Song, "Selfish Grids: Game-Theoretic Modeling and NAS/PSA Benchmark Evaluation," *IEEE Trans. Parallel and Distributed Systems,* accepted to appear.

[15] S. Marti and H. Garcia-Molina, "Limited Reputation Sharing in P2P Systems," *Proc. Fifth ACM Conf. Electronic Commerce,* May 2004.

[16] S. Marti and H. Garcia-Molina, "Identity Crisis: Anonymity versus Reputation in P2P Systems," *Proc. IEEE Int'l Conf. Peer-to-Peer Computing,* Sept. 2003.

[17] M. Mihail and A. Saberi, "Random Walks with Lookahead in Power Law Random Graphs," *Proc. World Wide Web Conf. (WWW),* May 2004.

[18] E. Ogston and S. Vassiliadis, "A Peer-to-Peer Agent Auction," *Proc. Int'l Conf. Autonomous Agents,* pp. 151-159, 2002.

[19] R.L. Page, S. Brin, and T. Winograd, "The Pagerank Citation Ranking: Bringing Order to the Web," technical report, Stanford Digital Library Technologies Project, 1998.

[20] T.G. Papaioannou, "Effective Use of Reputation in Peer-to-Peer Environments," *Proc. Int'l Symp. Cluster Computing and the Grid (CCGrid '04),* pp. 259-268, Apr. 2004.

[21] P. Resnick and R. Zeckhauser, "Trust among Strangers in Internet Transactions: Empirical Analysis of eBay's Reputation System," *The Economics of the Internet and E-Commerce,* vol. 11, 2002.

[22] M. Ripeanu, I. Foster, and A. Iamnitchi, "Mapping the Gnutella Network: Properties of Large-Scale P2P Systems and Implications for System Design," *IEEE Internet Computing,* vol. 6, no. 1, 2002.

[23] 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,* Dec. 2002.

[24] S. Saroiu, P.K. Gummadi, and S.D. Gribble, "A Measurement Study of Peer-to-Peer File Sharing Systems," *Proc. Multimedia Computing and Networking Conf. '02,* Jan. 2002.

[25] S. Sen and J. Wong, "Analyzing Peer-to-Peer Traffic across Large Networks," *Proc. ACM SIGCOMM Workshop Internet Measurement,* Nov. 2002.

[26] A. Singh and L. Liu, "TrustMe: Anonymous Management of Trust Relationships in Decentralized P2P Systems," *Proc. IEEE Int'l Conf. Peer-to-Peer Computing,* Sept. 2003.

[27] E. Sit and R. Morris, "Security Considerations for P2P Distributed Hash Tables," *Proc. Int'l Workshop Peer-to-Peer Systems (IPTPS '02),* Mar. 2003.

[28] S. Song, K. Hwang, and Y.K. Kwok, "Risk-Resilient Heuristics and Genetic Algorithms for Security-Assured Grid Job Scheduling," *IEEE Trans. Computers,* June 2006.

[29] S. Song, K. Hwang, R. Zhou, and Y.K. Kwok, "Trusted P2P Transactions with Fuzzy Reputation Aggregation," *IEEE Internet Computing,* pp. 18-28, Nov./Dec. 2005.

[30] M. Srivatsa, L. Xiong, and L. Liu, "Trustguard: Countering Vulnerabilities in Reputation Management for Decentralized Overlay Networks," *Proc. 14th Int'l World Wide Web Conf.,* pp. 422-431, 2005.

[31] I. Stoica, R. Morris, D. Nowell, D. Karger, M. Kaashoek, F. Dabek, and H. Balakrishnan, "Chord: A Scalable Peer-to-Peer Lookup Protocol for Internet Applications," *Proc. ACM SIGCOMM,* Aug. 2001.

[32] D. Talia and P. Trunfio, "Toward a Synergy between P2P and Grids," *IEEE Internet Computing,* July/Aug. 2003.

[33] Y. Wang, "Trust and Reputation Model in Peer-to-Peer Networks," *Proc. Third Int'l Conf. Peer-to-Peer Computing,* pp. 150-157, Aug. 2003.

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

[35] B. Yang, T. Condie, S. Kamvar, and H. Garcia-Molina, "Non-Cooperation in Competitive P2P Networks," *Proc. 25th IEEE Int'l Conf. Distributed Computing Systems (ICSCS '05),* 2005.

[36] M. Yang, Y. Dai, and X. Li, "Bring Reputation System to Social Network in the Maze P2P File-Sharing System," *Proc. IEEE Int'l Symp. Collaborative Technologies and Systems (CTS '06),* May 2006.

[37] H. Zhang, A. Goel, and R. Govindan, "Making Eigenvector-Based Reputation Systems Robust to Collusion," *Proc. Third Workshop Economic Issues in P2P Systems,* June 2003.

[38] R. Zhou and K. Hwang, "Trust Overlay Networks for Global Reputation Aggregation in P2P Grid Computing," *Proc. IEEE Int'l Parallel and Distributed Processing Symp. (IPDPS '06),* Apr. 2006.

Runfang Zhou received the PhD degree in computer science from the University of Southern California (USC) in May 2007. She received the BS and MS degrees in computer science from Southeast University, China in 1997 and 2002, respectively. She was a research assistant at the USC Internet and Grid Research Lab and USC/ISI computer network division from 2003 to 2006. Her research interests include peer-to-peer reputation systems, overlay network design, Web services performance improvement, and trust and secure collaboration in grid computing. Her thesis topic is "Scalable Reputation Systems for Peer to Peer Networks." She is a member of the IEEE.

Kai Hwang received the PhD degree from the University of California, Berkeley in 1972. He is a professor of electrical engineering and computer science and director of the Internet and Grid Research Laboratory at the University of Southern California. An IEEE Computer Society fellow, he specializes in computer architecture, parallel processing, Internet and wireless security, grid and cluster computing, and distributed computing systems. Dr. Hwang is the founding editor-in-chief of the *Journal of Parallel and Distributed Computing* (Elsevier). He is also on the editorial boards of *IEEE Transactions on Parallel and Distributed Systems* and of the *International Journal of High-Performance Computing and Networking*. He has published more than 200 scientific papers and seven books. His latest two books, *Scalable Parallel Computing* and *Advanced Computer Architecture*, are being adopted worldwide and translated into four languages. Presently, he leads a US National Science Foundation-supported GridSec project in developing security-binding techniques and distributed defense systems against worms and DDoS attacks for trusted Grid, P2P, and Internet computing.