Trust-Preserving Overlays for Fast Reputation Aggregation in Peer-to-Peer Grid Systems*

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

Abstract: In recent years, *Peer-to-Peer* (P2P) systems and *computational Grids* are evolving into a new distributed computing model, called *P2P Grid*. This paper presents a new approach to solving the trust and security problems in a P2P Grid. Establishing trust in P2P Grids is essential to build lasting working relationships among the peers joining collective Grid applications. A P2P reputation system is thus needed to collect peer trust scores and aggregates them to yield a global reputation in both P2P and Grid systems. We use a new *trust-preserving overlay network* (TON) to model the trust relationships among the peers. Examining the eBay transaction trace data, we discover a power-law distribution in user feedbacks, which is proven applicable to any dynamic P2P system or any P2P Grid system.

We develop a new P2P reputation system, *PowerTrust*, to leverage on power-law feedback characteristics. Our system is built with locality-preserving hash functions and a lookahead random walk strategy. Dynamic P2P system reconfiguration is enabled by using power nodes with well-established reputations. This power-node approach significantly reduces the global aggregation overhead. Through P2P simulation experiments on distributed file sharing and Grid *parameter-sweeping applications* (PSA) applications, we demonstrate the advantages of fast reputation convergence and accurate ranking of peer reputations. Simulated P2P Grid performance results are reported with enhanced query success rate, shortened job makespan, increased job success rate, and the alleviation of ill effects of malicious peers, after trust binding in either P2P systems or in P2P Grids.

Keyword: ***Peer-to-Peer systems, Grid computing, overlay network, trust management, distributed hash table, reputation system, distributed file sharing, parameter sweeping applications (PSA,* and *network security.***

---

* **Manuscript submitted to IEEE-TPDS, Nov. 5, 2005. This work was supported by NSF Grant ITR-0325409 at the University of Southern California.** **Corresponding author is Kai Hwang at Email: kaihwang@usc.edu, Tel. 213 740 4470, and Fax: 213 740 4418.**

1. Introduction

Dynamic P2P systems and static computational Grids are two popular distributed computing paradigms [4]. These two distributed system models have some commonalities as well as some conflicting goals as discussed in [12], [13], [32], [33]. *P2P Grids* are a natural merger of P2P and Grid systems [32]. Table 1 compares the architecture, control, security, and applications of the three distributed computing models. There is an increasing demand of fast and efficient reputation system to establish trust among the peers in either a P2P system or in a P2P Grid system. The PowerTrust system was built at USC to meet this demand. We provide both theoretical foundations and experimental results based on our PowerTusrt experiences. This paper extends from preliminary results we reported in an IPDPS-2006 paper [40].

P2P systems like the Gnutella, SETI@home and FightAIDS@home are client-oriented with scalable connectivity to mutiple millions of clients in content delivery and public information services [8], [23], [24]. A traditional *Peer-to-Peer* (P2P) computing system allows anonymous users (peers) to join and leave freely. Thus the P2P system is dynamically structured, highly scalable, and subject to abuses by malicious peer behaviors. This has posted a serious problem of trust, security, and privacy among the participating peers.

Existing computational Grids like the NSF TeraGrid and UK e-Science Grid are most supercomputer-oriented with limited scalability to serve a few hundreds of scientific users [2], [11]. The resources in a P2P Grid are mainly contributed by participating peers, which could be desktop clients [8] or deskside servers in a much larger quantity than existing Grids [33]. In other words, P2P Grids intend to merge the positive features from both P2P system and Grids. Killer applications of P2P Grids include scientific computing and web services. Jobs can be executed at local client machines or outsourced to remote peer machines on a P2P interaction basis.

In this paper, the in-and-out flexibility and fast search mechanisms in P2P systems [33] are explored for collective P2P Grid computing. The ultimate goal of building P2P Grids is to integrate the P2P, Grid, and web services in a unified system [13]. The P2P operation is inherently insecure due to the anonymity among the peers [28], [30]. Peers are autonomous, self-organizing, and thus are less structured, less secure, and less controllable than client-server or

Grid systems. The Grid security level is higher due to its accountability in resource registration and certified services provided [2], [11].

We propose a new approach to overcoming the security problem by developing a *trust-preserving overlay network* (TON) on top of the P2P Grid system. This paper considers mainly structured P2P Grids with decentralized resources from either participating peers or brokered Grid resources. The P2P Grids could be built from extending existing desktop Grids or scaling up an existing supercomputing Grid. The goat is to involve a large number of peer resources to be part of the Grid. This leads two classes of P2P Grids: Grids formed with PC desktops like the P-Grid [1], Entropia [8], and PC Grid [24] versus established Grids operating in a P2P setting like the community Grids [13] and other emerging Grids as discussed in [12].

Table 1 Comparison of P2P Systems, Computational Grids, and P2P Grids

Features	P2P Systems	Computational Grids	P2P Grids
Architecture and Connectivity	Flexible topology, highly scalable, autonomous users	Static configuration with limited scalability	P2P flexibility with Grid resource sharing initiatives
Control and Resource Discovery	Distributed control, client-oriented, free in and out, and self-organizing peers	Centralized control, server or supercomputer-oriented with registered participants	Policy-based control, operating with both P2P and Grid resource management
Security, Privacy and Reliability	Distrusted peers, insecure P2P interactions, and anonymity among peers	Guaranteed trust, more secure with federated users and accountability	Peer-layer reputation system and Grid-layer security infrastructure in a hierarchy
Applications and Job Management	General and commercial, self-organizing, peer initiated download services	Scientific computing, global problem solving, and hierarchical job management	Support desktop, distributed Grid computing, and community services
Representative Systems	Gnutella, Chord (DHT) [31], CAN, Tapestry, etc.	NSF TeraGrid , e-Science in UK [2], China Vaga Grid [33]	P-Grid [1], Entropia [8], P2P Grid [13], PC Grid [24]

The remainder of the paper is organized as follows: Section 2 reviews existing work on trust management in P2P systems and introduce the TON model for P2P reputation systems. We analyze in Section 3 the eBay trace data to reveal the power-law distribution of peer feedbacks. Section 4 introduces two mechanisms needed to build the PowerTrust reputation system. We specify the PowerTrust system construction and its updating algorithms in Section 5. We reveal in Section 6 the performance attributes of the PowerTrust system. The effects of these attributes are shown by simulation results. Then we report simulated experimental results on P2P system

and P2P Grid performances in Section 7. Finally, we conclude with discussions and suggestions for further research towards trusted P2P and/or Grid computing.

2. Trusted Computing in P2P Systems and P2P Grids

In this section, we consider the trust management issues that are specific to both P2P systems and P2P Grids. We introduce a new trust overlay approach to model the trust relationship among peers.

2.1 Trust Management in P2P and Grid Systems

In a P2P system or a P2P Grid, peers act as both clients and servers. Distributed resource registry/discovery and Grid job scheduling are supported by Grid middleware. Security in P2P Grids is managed at the local level as well as at the global level in a hierarchical manner. Policy-based control and peer participation are assumed. Special P2P reputation systems are needed to support trusted peer operations.

The P2P reputation system must have low-cost to build, easy to update, and fast in score dissemination and global reputation aggregation [30]. All peers have the freedom to interact with other peers freely and selectively. Grid-layer security are enforced by special middleware such Globus GSI and PKI services [3]. Peer-layer security relies on using reputation systems. We aim at developing an efficient reputation system that can support both P2P and Grid operations.

In the past, trust management in P2P systems was mainly supported by reputation systems built on top of peer feedbacks [1], [6], [10], [15]. For P2P Grids, the reputation systems [40] must be modified to deal with the collective resources put together by peer contributions. Building trust among the peers in a P2P Grid may encounter malicious [29] and selfish peers [16] in some P2P applications for e-commerce and on-line transactions and content delivery services.

Most contemporary P2P reputation systems are based on collecting, aggregating and disseminating feedbacks [6], [15], [22], [30], [37] among the peers, since a peer's past history is informative to predict its future behavior. Mining a large amount of P2P exchanges in P2P file sharing, collaborations, or distributed parallel computing, we will be able to reveal crucial features of peer feedbacks towards trusted P2P Grid computing [13].

Buchegger and Budded [6] presented a reputation evaluation scheme based on Bayesian learning technique. The EigenTrust mechanism [15] aggregates global reputation by a distributed calculation of the Eigenvector of the trust matrix over the peers. Song, et al [29], [30]suggested to use a fuzzy-logic trust management system to model the uncertainties involved in P2P transactions. Xiong and Liu [37] have developed the PeerTrust system for e-commerce applications. Our new approach is inspired by the above approaches. However, most trust management systems ignored the feedback properties of P2P systems by assuming an arbitrary feedback distribution, which may not agree with the reality in a P2P or Grid environment.

2.2 The Trust Overlay Approach

We introduce a new concept of *trust overlay network* (TON) to amend this ignorance. A TON is a virtual network on top of the P2P system. We represent a TON by a directed graph exemplified in Fig.1. The graph nodes represent peers and directed edges are the feedbacks between peers. The edge label represents *local trust score* between the source and destination peers. This example TON has 5 nodes. The node *N*5 downloads files from node *N*2 and node *N*7. The outgoing edges from *N*5 represent the feedbacks *N*5 left for *N*2 and *N*7. The global reputation is aggregated from all incoming local trust scores as shown for node *N*2.

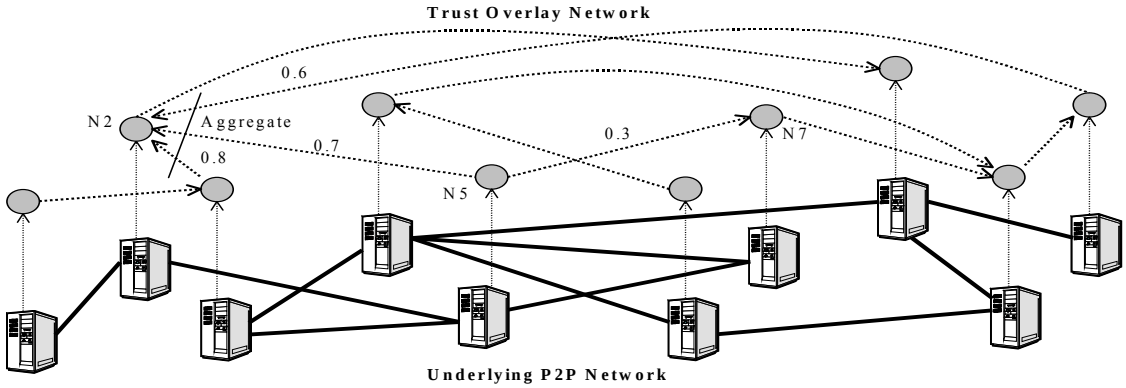


Figure 1. Graphical representation of the trust overlay network (TON) for a P2P Grid, where the nodes represent the peers and directed edges are labeled with local trust scores. The global reputation is aggregated from scores received on all incoming edges.

In a TON, the number of feedbacks a user sent to others is indicated by the *out-degree* of a peer node. The number of feedbacks a user received from others is represented by the *in-degree* of a peer node. We use TON to model the operations of the eBay reputation system during the past five years. We find that the eBay TON exhibits a *power-law distribution* in its node degrees. The power-law distribution is driven by two factors: the *dynamic growth* and *preferential node attachment* [23]. The former allows the network to expand with any newly added nodes. The later allows the new node to interact selectively with existing nodes.

We propose a dynamic trust management system, called *PowerTrust,* which leverages the power-law TON characteristics in dynamic P2P systems. This system uses a *look-ahead random-walk* (LRW) strategy to aggregate global reputation from local trust scores. Our scheme dynamically selects some power-nodes using a fully distributed ranking mechanism. This will ensure fast reputation convergence and defend against collusions by malicious peers. Although the trust management in eBay is centralized, we argue that the user behaviors are decentralized by nature and the feedback properties are user driven. We extend the *distributed hash table* (DHT) [28] and l*ocality preserving hashing* (LPH) [7] concepts to build our PowerTrust system by leveraging on the feedback properties.

3. Power Law of Peer Feedback Distribution

Power-law distribution is an inherent property of P2P systems. For example, the content distribution in Gnutella is power-law distributed [25]. We study the public-domain eBay reputation system to verify the conjecture that the feedback distribution of a typical P2P reputation system may also follow the power law. The distribution of received feedbacks is determined by the node in-degree in a overlay network.

Three important key parameters are identified in our study. The *feedback amount* of a node i is denoted by d_i, which is equal to the *in-degree* of node i. For example, node *N2* in Fig.1 has an in-degree of 3, meaning a feedback amount of 3. *Feedback frequency* f_d represents 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.

3.1 Collection Procedure of eBay Reputation Data

The eBay is by far the most successful cyber-exchange platforms based on a simple reputation mechanism [22]. The exchanging eBay users provide feedback to a centralized reputation center and report their experiences in eBay transactions. The scoring scheme in eBay is simple: positive 1 for a good feedback, negative 1 for a poor feedback, and zero for a neutral feedback. Every eBay user has a reputation by summing up all transaction scores, periodically.

It is difficult to collect all user feedbacks from eBay since the total number of eBay users is estimated to exceed 100 millions. We apply a sampling technique and collect 108 MB feedback data. We start from an arbitrary power user in eBay who has a global reputation score higher than 10,000. In order to infer the received feedback distribution (i.e. in-degree distribution) in TON, we gather a list of users to whom the power users left feedbacks from July 1999 to March 2005 and extract related information such as feedbacks received by those users.

Apparently, the more feedback a peer receives from others, the easier the user will be able to crawl in the overlay graph. Let p_d be the probability that the node with received feedback amount d is discovered by a random crawler, we have $p_d = d / \sum^n_{i=1} d_i$, where d_i is the received feedback amount of node i and n is the network size. Therefore, the probability that the node with received feedback amount d is still not discovered after k random crawls follow a Poisson distribution, i.e. $(1 - p_d)^k$. For a power node that gives k feedbacks to others, the probability that a node with received feedback d is crawled from the power node is :

$$P_d = 1 - \left(1 - p_d\right)^k = 1 - \left(1 - d / \sum\nolimits_{i=1}^{n} d_i\right)^k \quad \textbf{(1)}$$

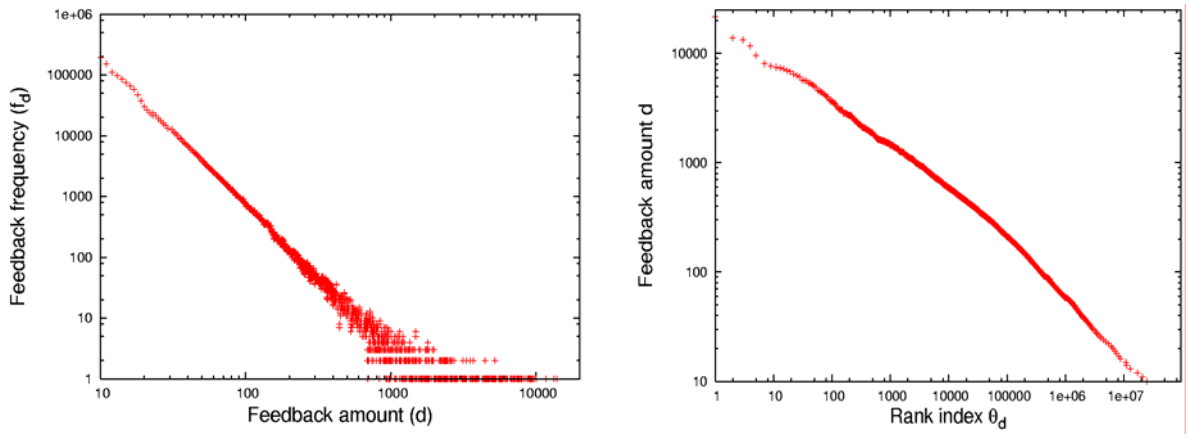
Let n_d be the original number of nodes with received feedback amount d in eBay TON, and $\hat{n}_d$ be the number of nodes with received feedback amount d in the sampling dataset, we have: $\hat{n}_d = E\left(n_d\right) \times P_d$ or the following expected value:

$$E\left(n_d\right) = \hat{n}_d / P_d \quad \textbf{(2)}$$

Equation (2) implies that we can estimate n_d from P_d and $\hat{n}_d$ to give more accurate account of the original feedback distribution in eBay. We call this procedure a *recovery process*.

3.2 Feedback Distribution in eBay Trace Data

Initially, we start with the sampling eBay trace over 11 thousands users (nodes). The eBay authority claims there are over 100 million users. Considering unregistered users and obsolete users, we assume that eBay has 80 million stable users. The average feedback amount per user in our trace data is 68. We approximate the total $\sum^{n}_{i=1} d_i$ by $80{,}000{,}000 \times 68 = 5.24 \times 10^9$. We apply the recovery process specified in Eq.(1) and Eq.(2) to model the eBay trace data and draw the feedback distribution in Fig.2(a). Only the nodes with received feedback amount larger than 10 were included. The figure plots the distribution of feedback frequency f_d. This quantity is proportional to the feedback amount d raised to the power of a *feedback exponent factor* $\beta \approx 2.4$, defined by $Pr(\deg(X) = d) = Cd^{-\beta}$, where X refers a node and C is a constant.



(a) Feedback frequency vs. feedback amount **(b) Feedback amount vs. rank index**

Figure 2. Power-law feedback distribution of eBay reputation system estimated from sampled eBay trace data over 10,000 users from July 1999 to March 2005

We plot in Fig.2 (b) the variation of the pairs (θ_d, d) using the recovered data, where θ_d is the ranking index of feedback amount d in the decreasing order. The plot is approximated well by a linear regression and the correlation coefficient is higher than 0.92, which implies that the feedback amount d is proportional to the feedback index θ_d in log-log scale. The distribution of the feedbacks in eBay transactions follows a power-law distribution as plotted in Fig.2.

We prove below how the power law distribution is observed in the eBay reputation system and why the result applies to any dynamic P2P systems. The detailed proof of this theorem is given in Appendix A.

***Theorem 1*:** In a general dynamic P2P system, the node degree of a TON follows the power-law connectivity, specified by the following power-law distribution:

$$\mathrm{P}_{rob}(\deg(X)=k)=\frac{X_k(t)}{t}=c_k=Ck^{-2} \quad \textbf{(3)}$$

The above theorem implies that the nodes with a small number of feedbacks are the common cases, whereas the node with a large number of feedbacks is rare. In a general dynamic P2P system, the corresponding TON follows this power-law connectivity, because the reputation system grows with new nodes that preferentially interact with the more reputable nodes. Effective use of power nodes can make a big difference in trust perseverance.

4. Reputation Aggregation and Ranking Mechanisms

Now, we are ready to specify the global reputation aggregation process in terms of the generation of initial reputation vectors, TON construction procedure, and reputation updating algorithm. We will prove the optimality of the algorithms under specific network conditions.

4.1 Lookahead Random Walk (LRW)

In a TON, every node keeps local trust scores for its neighbors. The eBay system generates local trust scores using the sum of positive scores for successful requests and negative scores for failed queries [22]. In our PowerTrust system, every node normalizes the local trust scores. Consider the *trust matrix* $R=(r_{ij})$ defined over an n-node TON, where r_{ij} represents the local trust score that *node i* rates *node j*. If there is no link in from node i to node j, the entry r_{ij} is set to 0. So for any $1 \le i, j \le n$, we have $0 \le r_{ij} \le 1$ and $\forall i, \sum_{j=1}^{n} r_{ij} = 1$.

We define the global trust score as a *global reputation* v_i for *node i*, suppose the global reputations for all nodes are stored in a vector $\vec{V}$, which is a normalized vector with $\sum v_i = 1$. The reputation vector $\vec{V}$ is computed by initializing the $\vec{V}^{(0)}$ and setting up an error threshold ε. For all $i = 1, 2, \ldots, l$, while $\left|\vec{V}^{(i)} - \vec{V}^{(i-1)}\right| > \varepsilon$, we compute the successive reputation vectors recursively by:

$$\vec{V}^{(i+1)} = R \cdot \vec{V}^{(i)} \quad \textbf{(4)}$$

This recursive process is motivated by the Markov random walk, which is widely used in ranking web pages. Imagine a random knowledge-surfer hopping from nodes to nodes in a TON 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 define a *greedy factor* α as the eagerness probability of a peer to link itself with a reputable power node. The higher is the value of α, the keener the peer wants to connect itself to a power node. We propose a *lookahead random walk* (LRW) strategy to efficiently aggregate global reputations. Each node in the TON not only holds local trust scores for its neighbors but also aggregates its neighbors first-hand local trust scores. The TON is a sparse power-law graph, the LRW does not cause heavy replication overhead. The replication overhead is limited by the number of edges in the TON, which is linear in a power law graph [19].

The efficiency of the LRW strategy is analyzed as follows. Given trust matrix R, initial reputation vector $\vec{V}^{(0)}$, and the converged reputation vector $\vec{V}'$, we have: $\vec{V}' = R^l \cdot \vec{V}^{(0)}$, where l is the number of iterations for regular random walk to converge. With the LRW strategy, every node aggregates the first-hand local trust scores of its neighbors, the trust matrix for LRW is specified by the fact that $R' = R^2$. So the number of iterations for LRW to converge is very close to half of the number required in a regular random walk.

We analyze the computation convergence time by checking the number of iterations of Eq.(4). Table 2 shows the speedup factors for graphs with different sizes. We generated 100 random graphs [19] and 100 Power-law graphs to compute the average speedup factor. The node degree distribution in a random graph is specified as follows:

$$\Pr ob\left(\deg(x) = k\right) = \binom{n-1}{k} p^k (1-p)^{n-k-1} \qquad \textbf{(5)}$$

where x is an arbitrary node, n is the graph size, and p = (Number of edges)/n^2. Our experiment results in Table 2 show that the LRW strategy greatly improves the convergence rate in a power-law graph or in a 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. This will preserve a lot of trust information around the neighbors of a power node.

Table 2: Speedup Factor of using Lookahead 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

4.2 Locality Preserving Hashing (LPH)

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

Algorithm 1: Distributed Top-m Ranking

Input: global reputations stored among score managers
Output: *m* most reputable nodes
for each score manager *j, suppose it is the score manager of node i* **do**
  hash reputation value v_i to a hash value $H(v_i)$ by using a LPH function
  insert the triplet (v_i, i, j) to the successor node of $H(v_i)$.
end for
/* the triplets are stored in the ascending order of their reputation values in the DHT hash space due to the property of LPH*/
initialize node *x* = successor node of the maximum hash value
Set a temporary variable *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

PowerTrust uses a *Distributed Hash Table* (DHT) such as Chord [31] that offers scalable key-based lookup for distributed aggregation. As in EigenTrust [15] every node has a score manager that accumulates its global reputation. We first hash the unique identifier of node i to a hash value k_i in the DHT hash space. Node j is assigned as the score manager of node i if node j is the *successor node* of k_i, denoted as *successor*(k_i). 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 are used by multiple score managers for each node to prevent malicious score manager from reporting wrong global reputation scores.

To select the m most reputable nodes, our distributed ranking mechanism applies a *locality preserving hashing* (LPH) to rank all nodes with respect to their global reputations. Cai, et al [7] suggested to use the LPH for resolving range queries in Grid information services. 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 $[H(v_k), H(v_j)]$.

The following theorem guarantees that Algorithm 1 always produce 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). The proof is left in Appendix B.

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

Figure 3 illustrates this trust-preserving scheme with an example of 5 nodes and 4-bit Chord circular hash space. Node $N15$ in Fig.3 is the score manager of node $N2$ whose global reputation is 0.2. Node $N15$ hashes the reputation value 0.2 using a simple LPH function $H(x) = 32x$. The resulted hash value is 6.4. Node $N15$ sends out a *Sort_Request*{*key*=6.4, (0.2, *N2, N*15)} message as a Chord insert request, which is routed to node. Node $N8$ stores the triplet (0.2, $N2$, $N15$), since it is the successor node of hash value 6.4.

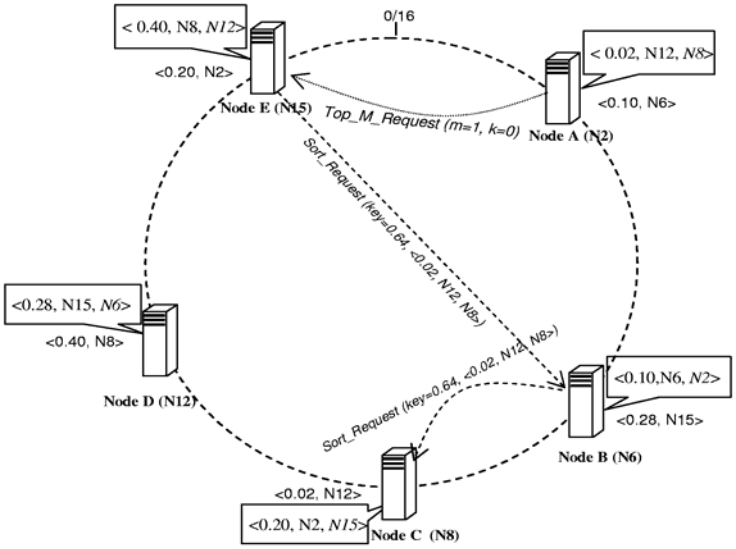


Figure 3 Distributed reputations ranking using the LPH function over a DHT-based P2P system with 5 peers

To illustrate, we show how to find the highest reputation node. Node *N*2 is the successor node of the maximum hash value 15 and is responsible for hash values in the range (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 to its predecessor node *N15*, which finds its stored triplet with value 0.4 being the highest one. So node *N8* is the most reputable node in this example system.

5. PowerTrust System Construction

Inspired by the good work of the EigenTrust [15] system, the PowerTrust system is built with the LRW and LPH mechanisms. This system provides a fully distributed solution to aggregate the global reputations of peers, such as millions of eBay users, simultaneously.

5.1 Initial PowerTrust System Construction

Algorithm 2 specifies the construction of the PowerTrust system in the first round of global reputation aggregation. Each node i sends the local trust scores to the score managers of its out-degree neighbors. If node i is the score manager of another node j, node i aggregates the local trust scores received from the in-degree neighbors of node j. The convergence overhead is measured by the inverse of the number of iterations in algorithm 2.

By Theorem 3 below, the bigger is the gap between λ_1 and λ_2, the faster is the rate of convergence. Fortunately, the power law property in a TON leads to a tight bound on the ratio λ_2 / λ_1 as proved in [14].

$$1 - \Omega(1/\log n) < \lambda_2 / \lambda_1 < 1 - \Omega(1/\log^2 n) \qquad \textbf{(6)}$$

As the ratio λ_2 / λ_1 gap becomes smaller, the power-law distribution of TON will guarantee the convergence at the very first round of global reputation aggregation. This result guarantee the convergence of Algorithm 1.

***Theorem 3*:** Given small error threshold ε, the number of iterations in Algorithm 1 is upper bounded by the smallest integer k such that $k = \log_{(\lambda 2/\lambda 1)} \varepsilon$, where λ_1 is the largest eigenvalue of trust matrix R defined over a TON and λ_2 is the second large one.

Algorithm 2 Initial PowerTrust Construction

Input: Local trust scores stored among nodes
Output: Global reputation for every node
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 ε
    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

5.2 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 node i is one of the

power nodes, 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 should be updated periodically, especially for the power nodes. The distributed updating of global reputation aggregation leverages on the use of the power nodes. The reputation updating process is specified in Algorithm 3.

Algorithm 3: Distributed 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 Algorithm1
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 ε
    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

Our PowerTrust scheme works as random walks on a Markov chain. The random surfer starts its journal on any node with the same probability. At any given node, the surfer selects a neighbor according to the local trust scores with a probability 1- α, where α is the *greedy factor* of the random walker. With a probability α, the surfer attaches itself with a power-node. The power-nodes are re-elected based on new global reputation value after each round because power-nodes are also dynamically changing over time. The *transition matrix* T is defined as:

$$T = (1-\alpha)(R')^T + \alpha P^T \qquad (7)$$

The matrix P is a rank-one matrix with most entries are zero except the entries with value $1/m$ in the columns associated with m power-nodes. The R' is the trust matrix based on using the LRW strategy. We can adjust *greedy factor* α to control the gap between the first and second largest eigenvalues of transition matrix T, given the largest eigenvalue $\lambda_1 = 1$ and the second largest eigenvalue as $\lambda_2 \leq 1-\alpha$. This was proved in [20].

6. Performance Analysis of PowerTrust System

The performance attributes of simulated PowerTrust system are analyzed below in three aspects: *reputation convergence overhead, reputation ranking discrepancy* and *aggregation error* caused by malicious behaviors.

6.1 Simulation Setup and Experiments Performed

Three sets of simulation experiments were performed. The first experiment evaluates the aggregation efficiency of global reputation by studying the convergence overhead. The second one demonstrates its accuracy in aggregated global reputation in terms of ranking discrepancy. The third one measures the RMS aggregation error in face of different malicious behaviors. Our simulation experiments were implemented on a dual-processor Dell server with 2GB of RAM running the Red-hat 9.0 Linux/OS with kernel 2.4.20. Each data point reported represents the average of at least 10 simulation runs. Simulation parameters and default values used in the experiments are summarized in Table 3.

Table 3 Simulation Parameters and Default Values

Parameter	Brief Definition	Default Value
n	Number of initial peers in a P2P system	1000
β	Power-law exponent factor	2.4
α	Greedy factor of random walker	0.15
f_d	Maximum Peer feedback amount	200
γ	Percentage of malicious peers in a P2P system	20%
p	Number of power nodes in a P2P systems	1%
ε	Threshold for global reputation convergence	10^{-4}

Our initial simulated TON for the P2P system was a fully connected power-law graph, consisting of 1,000 nodes with the maximum node degree $d_{max} = 200$ and exponent factor $\beta = 2.4$. We assume 80% honest peers and 20% malicious peers in the P2P system. We model two types of malicious behaviors: one type reports dishonest trust scores (such as reporting low local trust scores for good peers and vice versa). Another type collaborates with users to boost up their own ratings. They may rate the peers in their collusion group very high and rate others very low. We select only 1% power-nodes over the total number of nodes in a TON.

We compare below the performance between our PowerTrust system and Stanford EigenTrust system [15] for the following metrics: *convergence overhead, ranking discrepancy and RMS aggregation error.*

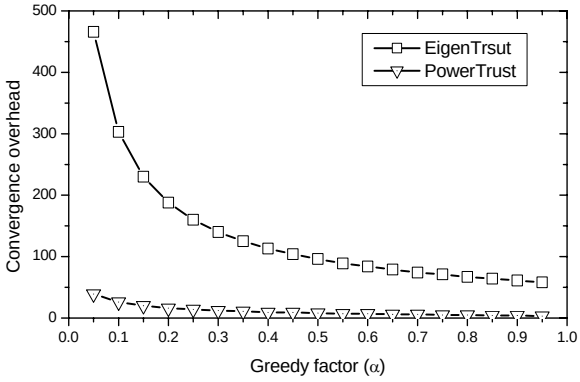
6.2 Reputation Convergence Overhead

The *convergence overhead* is measured as the number of iterations before the global reputations converging. The EigenTrust approach relies on a few pre-trust nodes to compute the global reputations. They assumed that some peers are known trustworthy, essentially the very first few peers joining the system. This assumption may not agree with the reality of decentralized P2P computing. We randomly choose some reputable nodes as pre-trust nodes in our simulation experiments. We report in Fig.4 the effects of different greedy factor α and system sizes n on the variation of the convergence overhead.

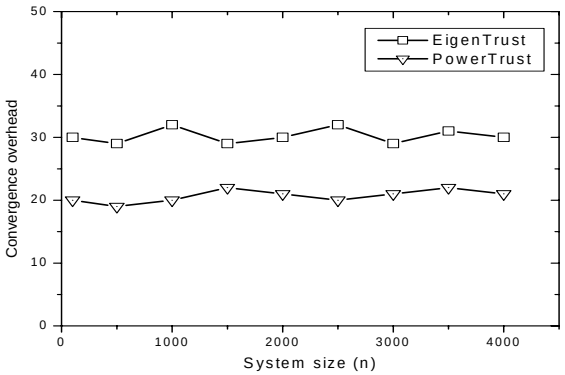
In a dynamic P2P system, peers are dynamically joining and leaving. For all fairness, we choose the same number of pre-trust nodes for EigenTrust as the number of power nodes in our simulation experiments. Figure 4(a) shows the convergence overheads for the two reputation systems, when there is no pre-trust or power node leaving the P2P network. We observe a sharp drop of iteration count in Fig.4 (a), when α increases from 0.15 towards 1. Figure 4(b) shows only a small fluctuation of the convergence overhead as the system size varies from 500 to 4,000 nodes. In both plots, the PowerTrust system outperforms the EigenTrust system.

Figure 4(c) shows the convergence overhead for the two reputation systems, when pre-trust or power node is allowed to leave the P2P network. We observe a sharp drop of iteration

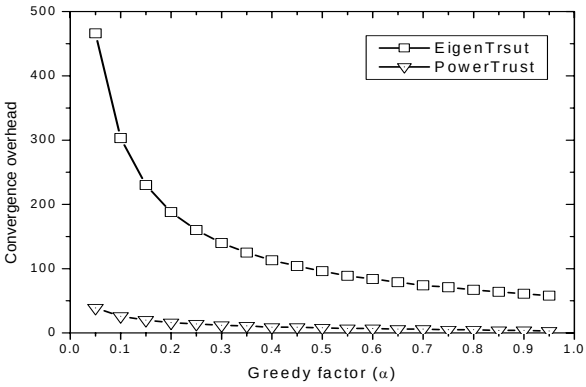
count in Fig.4(c), when α increases from 0.15 to 1. Figure 4(d) shows that our PowerTrust system has almost a flat small convergence overhead, as the greedy factor α is maintained small with a default value of 0.15, regardless of the system sizes. The EigenTrust system has high overhead exceeding 200 iterations under such a condition.



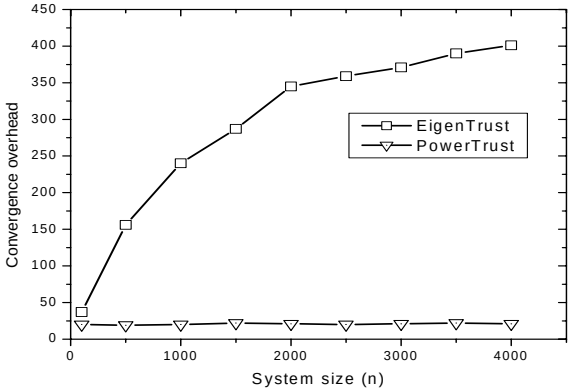
(a) Effects of greedy factor α without power nodes or per-trust nodes leaving (n=1000)



(b) Effect of system size n without power nodes or per-trust nodes leaving (α=0.15)



(c) Effects of greedy factor α allowing power nodes or per-trust nodes leaving (n=1000)



(d) Effect of system size n allowing power nodes or per-trust nodes leaving (α=0.15)

Figure 4. Convergence overhead of two P2P reputation systems: PowerTrust and EigenTrust under variable greedy factor and system sizes

In both plots, the PowerTrust system outperforms the EigenTrust system sharply. The EigenTrust system converges very slowly or even cannot guarantee its convergence when pre-trust nodes are allowed to leave the system freely. In the PowerTrust system, the power nodes will be re-elected after each aggregation round. Based on the distributed ranking mechanism, when some power nodes leave, the score managers of the departing power nodes notify the

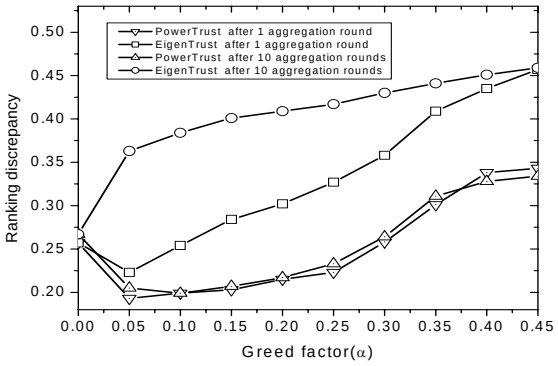
system to replace them with other qualified power nodes. The low overhead in using the PowerTrust system makes it attractive in performing highly scalable P2P Grid applications.

6.3 Reputation Ranking Discrepancy

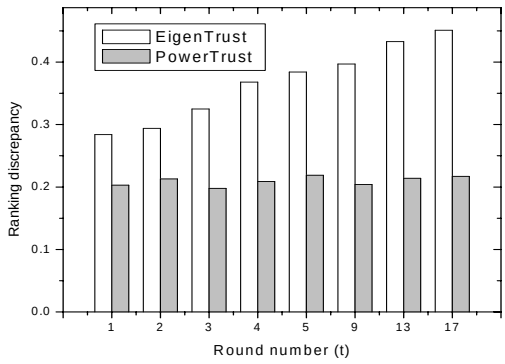
To estimate the accuracy of the aggregated global reputation, we rank the peers by their global reputation scores. We measure the *ranking discrepancy* between the obtained ranking and the true ranking. In our experiments, we use the normalized Euclidean distance [9] to measure the ranking discrepancy. During each round of global reputation computation, 100 new peers join the system and interact with the peers already in the system. The probability of a transaction conducted between node i and node j is determined by $d_i d_j / \sum^{n}_{k=1} d_k$, where d_i and d_j are the corresponding node degrees. This property ensures that the growing TON follows power-law connectivity [14].

Figure 5 shows the ranking discrepancy between the actual and obtained rankings as a function of greedy factor. The result is plotted in Fig.5(a) after the first round of global reputation aggregation and the 10-th round of aggregation respectively. After first round aggregation, both pretrust nodes in Eigentrust and power-nodes in PowerTrust system can reduce the effects of malicious nodes, when α is very small ($\alpha < 0.15$). When α is larger than 0.05, PowerTrust has smaller ranking discrepancy than EigenTrust. After 10 rounds of reputation, the ranking discrepancy of EigenTrust increases dramatically even for small α.

Figure 5(b) shows the aggregation effect with $\alpha = 0.15$, the ranking discrepancy in EigenTrust is increasing over the round number while the PowerTrust is always maintained low. The global reputation obtained from PowerTrust is more accurate than EigenTrust. Our PowerTrust system performs much better in terms of accuracy than EigenTrust as the ranking discrepancy of PowerTrust is always smaller than EigenTrust for different small α. Our power-nodes are the most reputable nodes that are dynamically chosen after each round of aggregation, while the pre-trust nodes are statically chosen, regardless of their future performance.



(a) Effect of greedy factor α after 1 and 10 aggregation rouds



(b) Effect of aggregation rounds under α=0.15

Figure 5. Ranking discrepancy of the PowerTrust system, compared with that of the EigenTrust system

6.4 Effects of Malicious Peer Behaviors

We evaluate the effectiveness and robustness of the PowerTrust system against malicious peer behaviors. The experiment was performed under both noncollusive and collusive P2P settings. We compute the *root-mean-square* (RMS) of the aggregated reputation of all peers. A lower RMS error indicates better performance. The RMS is defined as:

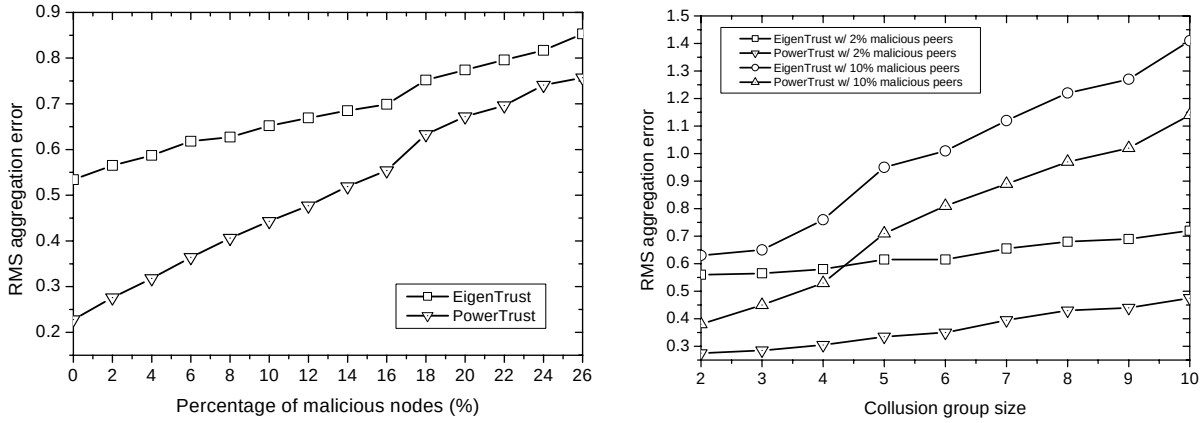
$$\text{RMS aggregation error} = \sqrt{\frac{\sum((v_i - v_i^{'})/v_i)^2}{n}} \qquad \textbf{(8)}$$

where v_i and $v_i^{'}$ are the actual and measured global reputation scores of peer node i, respectively.

We plot the RMS error against the percentage of malicious peers in Fig.6(a). The default peer greedy factor $\alpha = 0.15$ was assumed. The probability of a node being malicious is modeled by inverse of its true global reputation. Figure 6(a) shows the RMS aggregation error for a noncolusive peer setting, in which all malicious peers report false local trust scores, independently. With 2% malicious peers, the PowerTrust system has 53% 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.6 (b), we model the collusive peers working collaboratively to abuse the system. We report the RMS aggregation errors for different collusion sizes, the number of malicious

peers working in a group. In all cases (2% and 10% malicious peers), the PowerTrust show its robustness against collusive peer groups of various sizes. The EingenTrust is shown much less resistant to abuses by large collusive peer groups.



(a) Noncollusive peers **(b) Collusive peers**

Fig 6. Reputation aggregation error from reporting by malicious peers

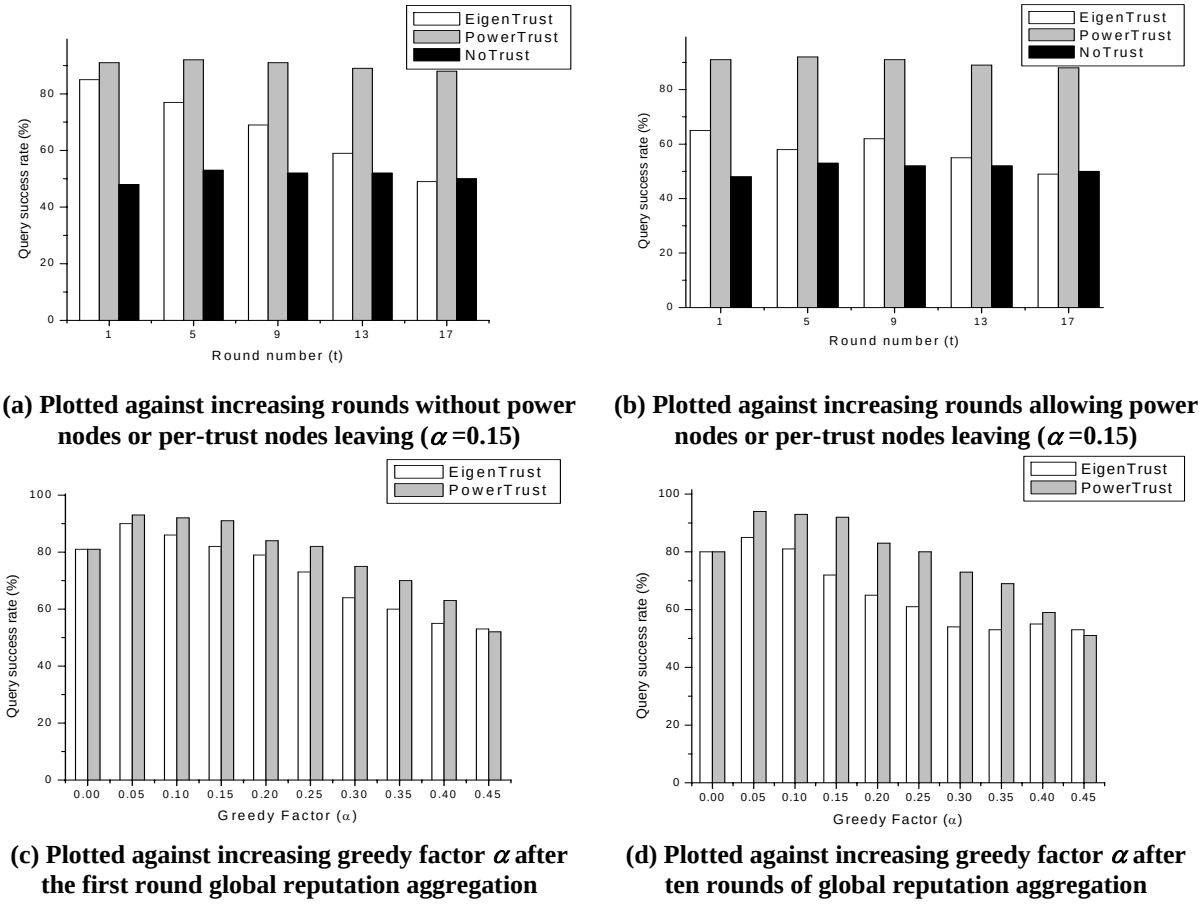
7. P2P Application Benchmark Results

In this section, we show the simulated P2P application performance results in using the PowerTrust to aggregate peer reputations. One application is distributed file sharing among the peers and the second is for distributed P2P computing over the benchmark of *parameter sweeping applications* (PSA), often used in Grid experiments [29].

7.1 Query Success Rate in Distributed File Sharing

We have applied the PowerTrust system on simulated P2P file-sharing applications. The *query success rate* in these P2P applications was evaluated here. The query model is the same as the one proposed in Marti and Garcia-Molina [13]. There are over 100,000 files being simulated in the P2P system. The number of copies of each file in the system is determined by a Power-law distribution with $\updownarrow$ = 1.2. Each peer is assigned with a number of files based on the Sarioiu distribution [19].

At each time step, a query is randomly generated at a peer and completely executed before the next query/time step. The popularity distribution determines the file the query intended to access. We use a Power-law distribution with a $\beta = 0.63$ for rank 1 to 250 and an $\beta = 1.24$ for the remaining ranks. This distribution models the query popularity in existing P2P systems. 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. Figure 7 shows the query success rate in using the PowerTrust and EigenTrust systems.



(a) Plotted against increasing rounds without power nodes or per-trust nodes leaving (α =0.15)

(b) Plotted against increasing rounds allowing power nodes or per-trust nodes leaving (α =0.15)

(c) Plotted against increasing greedy factor α after the first round global reputation aggregation

(d) Plotted against increasing greedy factor α after ten rounds of global reputation aggregation

Figure 7. Query success rates of two P2P reputation systems: PowerTrust vs. EigenTrust, under various trust overlay network conditions and aggregation rounds

The query success rate is measured by the percentage of success queries over the total number of queries issued. Every node has a dropping rate in the range from 0% to 100%. For

simplicity, the node dropping rate is modeled inversely proportional to its actual global reputation, given the zero dropping rate for the most reputable node and 100% dropping rate for the worst reputable nodes. We consider both cases of with and without Power-nodes or pre-trust nodes leaving the system.

Figure 7(a) shows the results without dynamic power nodes. There are 1000 queries evaluated after each round of global reputation aggregation. The query success rate of PowerTrust is higher than that of using the EigenTrust. The EigenTrust performs even lower as the number of rounds increases. Figure 7(b) shows the query success rates for both systems as a function of rounds, when one power nodes leave. The EigenTrust is not better than the no trust case when the number of rounds increases.

By no trust, we mean there is not any trust management in P2P system and we randomly select a node to download the desired file. Figure 7(c) plots the result against increasing value of greedy factor α, after the first round. The PowerTrust performs better than EigenTrust, when α is small. Figure 7(d) shows the results after ten rounds of aggregation. The EigenTrust performs low, even when α is small.

7.2 P2P Grid Performance over The PSA Workload

In this section, we use the following metrics to simulate the PowerTrust performance in P2P Grid job execution application.

(1) *Makespan:* Denote the total number of simulated jobs as M, and denote the completion time for a single job J_i as c_i, the makespan is defined as $\max\{c_i\}$.

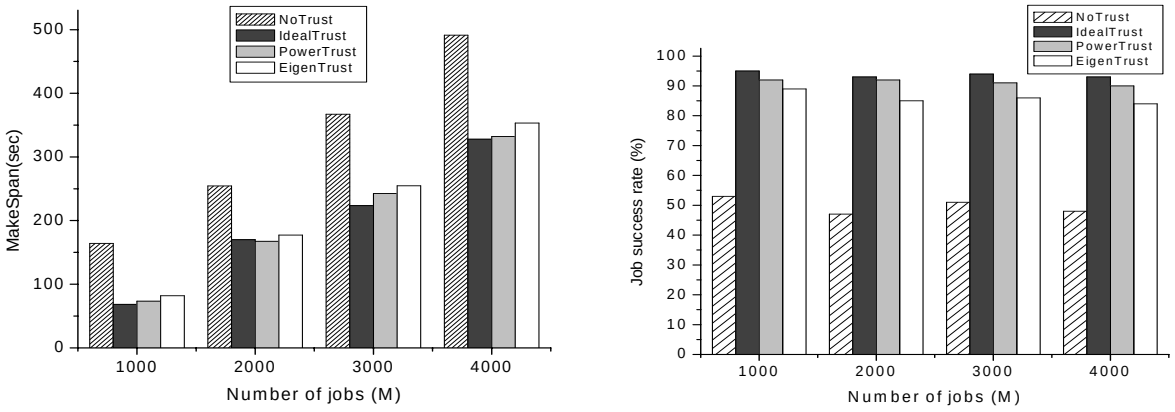
(2) *Job failing rate:* Job execution may fail at low reputation sites. M_{fail} counts the number of failed and rescheduled jobs. Job failing rate is defined as $F_{rate}=M_{fail}/M$.

We apply a realistic PSA workload in the simulation experiments. The PSA model is defined as a set of independent sequential jobs (i.e., no job precedence). The independent jobs operate on different datasets. A range of scenarios and parameters to be explored are applied to the program input values to generate different data sets. The execution model essentially

involves processing *M* independent jobs (each with the same task specification, but a different dataset) on *N* distributed sites where *M* is much larger than *N*.

A heuristic Min-Min scheduling is used for job scheduling. Per each job, the Grid sites having the shortest *expected time-to-completion* (ETC) is selected. 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 associated with the Grid site. Then the job with the minimum ETC is selected and assigned to the Grid site selected. After each job execution, the Grid sites update the trust score of other sites. These trust scores will be incremented by 1 for job successfully executed or 0 if failed. Therefore, the edges on the TON overlay will be relabeled with new scores periodically.

A job is executed if it is rejected no more than 3 times. Figure 8 shows the performance results of 4 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 peers's real global reputations are accessible. The light-gray and white bars correspond to using the PowerTrust and EigenTrust systems, respectively. Figure 8 reports the simulated PSA benchmark results.



(a) Job Makespan in second **b) Average job success rate**

Figure 8. PSA benchmark performance results on a simulated P2P Grid configuration with 100 peer-contributed resource sites

In Fig.8 (a), the job makespan of all 4 reputation systems increases with the job number. Figure 8(b) shows the average job success rate, which drops slowly with the workload size. As

predicted, the *NoTrust* has the longest makespan performance and significantly lower job success rate among the 4 systems. Both PowerTrust and EigenTrust have comparable makespan performance and job success rates. The job makespan of both systems are close to the ideal performance of the *IdealTrust* system.

In all cases, the PowerTrust slightly outperforms the EigenTrust system by about 2% and they both converge to the ideal performance with less than 4% of from the optimal value. Without trust, the job makespan increases 30% and the job success rate drops by 46%, compared with the fully trusted case. These results prove the effective of using global reputation aggregation in establishing trust among the participating peer machines in a P2P Grid system.

8. Conclusions and Further Work

In this paper, we developed a *trust-preserving 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 the overlay graph. This power-law distribution is not restricted to eBay reputation data. It is applicable to general dynamic P2P systems that allows free joining and departure of user nodes. Our prototype PowerTrust system offers the very first approach to aggregating local trust scores to yield global reputations by leveraging primarily on the power nodes to preserve most of the trust information. The system is built with *locality preserving hash* (LPH) functions, which can be implemented over a DHT-based P2P system.

The performance of the PowerTrust was evaluated by measuring the convergence overhead of global reputation, query success rate in P2P file sharing, and job makespan and success rate in simulated PSA Grid benchmark experiments. The PowerTrust advantages come mainly from the use of the LRW strategy in system construction and the use of the LPH function in fast reputation updating. These features help accelerate the reputation aggregation and enable security binding in both P2P systems and P2P Grids over trustworthy peers.

Based on the results reported, we reveal the following advantages of using a distributed reputation system to make P2P systems and P2P Grids more robust and resilient to trust variations or abuses in peer behaviors.

- P2P Grids broadcast messages and offers higher application flexibility than static Grid configurations. P2P Grids may deal with changing or unknown IP addresses from roaming users or firewalls. This gives more protection in privacy and anonymity [36].
- The OGSA protocols have been partially developed for Grids under the assumption of uniform trust and reliability. For P2P Grids, this assumption should be extended to follow the Power-law distribution in peer feedbacks. Resource discovery is now distributed and dynamic to avoid central servers as in conventional Grids.
- P2P Grid resources are autonomous, self-organizing, decentralized at user-space based network environments. These properties could be used to achieve higher client interactivity and fault tolerance in case of node failures.
- From resource sharing viewpoint, P2P Grids merges the strong points of both P2P and Grid. Much higher scalability in P2P Grids, faster search and storage capability, and higher degree of resource sharing at the edge and middle of the Internet [13], as compared with the rigid interconnects and centralized control in conventional Grids.

For further research, we suggest to extend the work along three orthogonal dimensions: First, different threat models should be investigated to secure P2P applications. We need to explore new mechanisms to build more secure and robust systems against malicious intrusions, especially collusions [39]. Second, we need to explore new killer applications of the P2P Grids beyond the file sharing and PSA applications reported here [33]. Third, the distrust problem will become even more complex in real-life selfish Grids [16], [29]. These issues demand the upgrade of existing P2P reputation systems in scalable P2P Grid applications, which may involve millions of participating peers that may join and leave freely in a global scale. Finally, the standardization of P2P Grids along the roadmap of OGSA is encouraged for both P2P and computing communities.

Acknowledgements: This work was supported by NSF ITR Grant ACI-0325409 at USC Internet and Grid Computing Lab. We appreciate the comments by Dr. Ricky Kwok of Hong Kong University, Dr. Jianping Pan of University of Victoria, and our colleagues, Min Cai and

Shanshan Song for their valuable suggestions to improve the quality and readability of this paper.

Appendix : Proofs of Theorems

A. Proof of Theorem 1: Consider a new user *u* interacts with an existing user *v*. The probability of such an engagement is indicated by the in-degree of user *v*. Let $X_k(t)$ be the random variable of users whose in-degree equals *k* after *t* interactions. This number increases to $X_k(t+1)$ at next interaction, if it has a transaction with a user having *k*-1 in-degree. Such a transaction has a probability $(k-1)X_{k-1}(t)/t$ to occur. The user decreases $X_k(t+1)$, if it interacts with the one within the $X_k(t)$ user group. Such a transaction has a probability $KX_k(t)/t$. So we have the expected value: $E(X_k(t+1)-X_k(t)) = ((k-1)X_{k-1}(t)-kX_k(t))/t$. Suppose the expected growth of $X_k(t)$ converges to c_k as $t \to \infty$, we have $c_k = (k-1)c_{k-1} - kc_k$, which yields $c_k = Ck^{-2}$. This completes the proof of the power-law distribution in Eq.(3).

B. Proof of Theorem 2: Suppose V=$\{v_i \mid 0 \le i < m\}$ are the *m* largest reputation values, we have $v_k \le v_i \le v_{max}$, where v_{max} is the largest reputation value. *Using a* LPH function *H* to map reputation value into the Chord identifier space, we map reputation value v_k to *successor*($H(v_k)$). We have $H(v_k) \le H(v_i) \le H(v_{max})$. Since all hash values of H are within the identifier space [0, 2^b-1), the $H(v_{max})$ must be equal to or less than 2^b-1. Therefore, we have $H(v_k) \le H(v_i) \le 2^b$-1. Thus the reputation value v_i is only assigned to nodes whose identifier is between *successor*($H(v_k)$) and successor(2^b-1). Thus by Algorithm 1, we are guaranted to find the *m* most reputable nodes after traversing from *successor*(2^b-1) to *successor*($H(v_m)$).

C. Proof of Theorem 3: Consider a matrix *R* with *m* eigenvectors $x_1, x_2, \ldots, x_m$ and their eigenvalues $\lambda_1 > \lambda_2 >, \ldots, > \lambda_m$. The initial reputation vector *y* is written as $\sum^m_{i=1} b_i x_i$. We have $Ry = \sum^m_{i=1} b_i Rx = \lambda_1(b_1x_1 + \sum^m_{i=2}(\lambda_i/\lambda_1)b_ix_i)$. Thus, $R^j y = \lambda_1^j(b_1x_1 + \sum^m_{i=2}((\lambda_i/\lambda_1)^j b_ix_i)$. Since $(\lambda_2/\lambda_1)^k = \varepsilon$, we have $(\lambda_i/\lambda_1)^k < \varepsilon$, where $2 < I \le m$. Therefore, $R^k y = \lambda_1^k b_1 x_1 + o(\varepsilon)$, which implies that the repeated application of *R* to *y* will converge in *k* iterations.

References:

[1] K. Aberer, "P-Grid: A Self-Organizing Structured P2P System", *Proc. of ICCIS*, Lecture Notes in Computer Science No. 2172, Springer Verlag, 2001.

[2] E. Adar and B. A. Huberman, "Free Riding on Gnutella", *First Monday*, Vol.5, No.10, 2000.

[3] F. Azzedin and M. Maheswaran, "A Trust Brokering System and Its Application to Resource Management in Public-Resource Grids", *Proc. IPDPS 2004*, Santa Fe, NM, April 2004.

[4] F. Berman, J. Fox, and T. Hey (Editors), "Grid Computing: Making The Global Infrastructure a Reality", Wiley and Sons, N.Y., April 2003.

[5] E. Bertino, E. Ferrari, and A. C. Squicciarini, "Trust-X: A Peer-to-Peer Framework for Trust Establishment", *IEEE Transactions on Knowledge and Data Engineering,* Vol.16, No.7, July 2004, pp.827 – 842.

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

[7] M. Cai, M. Frank and P. Szekely, "MAAN: A multi-attribute addressable network for grid information services", *Journal of Grid Computing*, Vol.2, No.1, 2004, pp.3-14.

[8] A. Chien, et al, "Architecture and Performance of an Enterprise Desktop Grid System", *Journal of Parallel and Distributed Computing,* Vol.63, No.5, May 2003, pp.597-610.

[9] C. Dellarocas, "Analyzing the Economic Efficiency of eBay-like Online Reputation Reporting Mechanisms", *Proc. of the 3rd ACM conference on Electronic Commerce,* Tampa, FL., 2001.

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

[11] I. Foster, C. Kesselman, and S. Tuecke, "The Physiology of the Grid", *Open Grid Service Infrastructure WG, Global Grid Forum,* June 22, 2002.

[12] I. Foster and A. Iamnichi, "On Death, Taxes, and Convergence of P2P and Grid Computing", *Proc. of the 2 nd International Workshop on Peer-to-Peer Systems* (*IPTP3'03*), Berkeley, Feb.2003.

[13] G. Fox, et al, "Peer-To-Peer Grids", Chapter 18 in *Grid Computing*, eds. Berman, Fox, and Hey, John Wiley & Sons, West Sussex, England, 2003.

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

[15] S. Kamvar, M. Schlosser, and H. Garcia-Molina, "The Eigentrust Algorithm for Reputation Management in P2P Networks", *ACM WWW'03,* Budapest, Hungary, May 2003.

[16] K. Kwok, S. Song, and K. Hwang, "Selfish Grid Computing: Game-Theoretic Modeling and NAS Performance Results", *in Proceedings of the International Symposium on Cluster Computing and the Grid (CCGrid-2005),* Cardiff, UK. May 9-12, 2005.

[17] S. Marti and H. Garcia-Molina, "Identity Crisis: Anonymity vs. Reputation in P2P Systems", *IEEE Intl. Conf. on Peer-to-Peer Computing*, Sydney, Australia, Sept. 2003.

[18] S. Marti and H. Garcia-Molina, "Limited Reputation Sharing in P2P Systems", Proc. of *the 5th ACM conference on Electronic Commerce,* New York, May 2004.

[19] M. Mihail, A. Saberi, P.Tetali, "Random Walks with Lookahead in Power Law Random Graphs", *WWW,* New York, May 2004.

[20] 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.

[21] T.G. Papaioannou and G.D.Stamoulis, "Effective Use of Reputation in Peer-to-Peer Environments", *Int'l Symp. on Cluster Computing and the Grid* (CCGrid'04), Chicago, April 2004, pp.259–268.

[22] 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, Volume 11 of Advances in Applied Microeconomics,* Amsterdam, Elsevier Science, 2002.

[23] 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.

[24] K. D. Ryu and J. K. Hollingsworth, "Unobtrusiveness and Efficiency in Idle Cycle Stealing for PC Grid", *Proc. IPDPS 2004,* Santa Fe, NM., April 2004.

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

[26] H. Shen, C. Z. Xu, and C. Chen, "Cycloid: A Constant-Degree and Lookup-Efficient P2P Overlay Network", *Proc. IPDPS 2004,* Santa Fe, NM, April 2004.

[27] A. Singh and L. Liu, "TrustMe: Anonymous Management of Trust Relationships in Decentralized P2P Systems", *IEEE Intl. Conf. on Peer-to-Peer Computing*, Sep. 2003.

[28] E. Sit and R. Morris, "Security Considerations for P2P Distributed Hash Tables", *Proc. IPTPS 2002*, Cambridge, MA., March 2003.

[29] S. Song, K. Hwang, and Y.K. Kwok, "Trusted Grid Computing with Security Binding and Trust Integration", *Journal of Grid Computing,* Vol.3, No.1, Sept 2005.

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

[31] Stoica, R. Morris, D. Liben-Nowell, D.Karger, M. Kaashoek, F. Dabek, and H. Balakrishnan, "Chord: A Scalable Peer-to-Peer Lookup Protocol for Internet applications", *Proceedings of ACM SIGCOMM,* San Diego, Aug. 2001.

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

[33] C. Tang, Z. Xu, and S. Dwarkadas, "Peer-to-Peer Information Retrieval using Self-organizing Semantic Overlay Networks", *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, Karlsruhe, Germany, Aug. 2003.

[34] P. Uppiluri, N. Jabisetti, U. Joshi, and Y. Lee, "P2P Grid: Service-Oriented Framework for Distributed Resource Management", *IEEE Int'l Conf. on Web Services,* Orlando, FL., July 2005.

[35] Y. Wang and J.Vassileva, "Trust and Reputation Model in Peer-to-Peer Networks", *Proc. of Third In'l Conference on Peer-to-Peer Computing,* Zurich, Switzerland, Aug. 2003, pp.150 – 157.

[36] L. Xiao, Z. Xu and X. D. Zhang, "Low-cost and Reliable Mutual Anonymity Protocols in P2P Networks", *IEEE Tran. on Parallel and Distributed Systems,* Sept. 2003, pp. 829 – 840.

[37] L. Xiong and L. Liu, "PeerTrust: Supporting Reputation-based Trust for Peer-to-Peer Electronic Communities", *IEEE Trans. Knowledge and Data Engineering,* Vol.16, No.7, 2004, pp. 843-857.

[38] S. Ye, F. Makedon, and J. Ford, "Collaborative Automated Trust Negotiation in Peer-to-Peer Systems", *Proc. of 4th Int'l Conf. on Peer-to-Peer Computing,* Aug. 2004, pp.108 – 115.

[39] H. Zhang, A. Goel and R. Govindan, "Making Eigenvector-based Reputation Systems Robust to Collusion", *The Third Workshop on Economic Issues in P2P Systems,* Berkeley, CA, June 2003.

[40] R.Zhou and K.Hwang, "Trust Overlay Networks for Global Reputation Aggregation in P2P Grid Computing", *Proc. IPDPS2006*, Rhodes Island, Greece, April 2006, (submitted).

Biographical Sketches:

Runfang Zhou received the B.S. and M.S. in computer science from Southeast University. She is currently pursuing the Ph.D. degree in Computer Science at the University of Southern California. She works at the USC Internet and Grid Computing Laboratory as a Research Assistant. Her research activities cover Peer-to-Peer reputation systems, overlay network design, web services performance improvement and trust and secure collaboration in Grid computing. She can be reached at: rzhou@usc.edu.

Kai Hwang is a Professor of Electrical Engineering and Computer Science and Director of Internet and Grid Computing Laboratory at the University of Southern California. He received the Ph.D. from the University of California, Berkeley. An IEEE Fellow, he specializes in computer architecture, parallel processing, Internet security, Grid, P2P, cluster and distributed computing systems. Presently, he leads the NSF-supported ITR GridSec project at USC. The GridSec group develops security-binding techniques and defense infrastructures for trusted P2P and Grid computing. Dr. Hwang can be reached via: kaihwang@usc.edu or through his personal web site http://GridSec.usc.edu/Hwang.html.