Trust Overlay Networks for Global Reputation Aggregation in P2P Grid Computing*

Runfang Zhou and Kai Hwang

University of Southern California

Abstract: This paper presents a new approach to trusted Grid computing in a *Peer-to-Peer* (P2P) setting. Trust and security in P2P Grids are essential to establish 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. We use a new *trust overlay network* (TON) to model the trust relationships among the peers. After analyzing the eBay transaction trace data, we discover a power-law distribution in user feedbacks. This power law is proven applicable to any dynamic P2P systems. We develop a new P2P reputation system, *PowerTrust*, to leverage on the power-law feedback characteristics.

The PowerTrust system is built with locality-preserving hash functions and a lookahead random walk strategy. Dynamic system reconfiguration is enabled by the use of power nodes with well-established reputations. This power-node approach significantly reduces the 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 P2P query success rate, job makespan and job success rate, after security binding with the reputation system in scalable P2P Grid applications

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

1. Introduction

Peer-to-Peer (P2P) systems and *computational Grids* are two popular distributed computing paradigms that are converging in recent years. The P2P systems like the Gnutella, SETI@home and FightAIDS@home are client-oriented with scalable connectivity to serve millions of clients in commercial/information service settings [6], [19], [20]. Existing computational Grids like the NSF TeraGrid and UK e-Science Grid are most supercomputer-oriented with limited connectivity to serve a handful (say hundreds) of scientific users [3], [8]. These two distributed systems have some commonalities as well as some conflicting goals as discussed in [9], [10], [28], [30].

P2P Grids are a natural merger of the above two prominent distributed computing technologies. The resources in a P2P Grid are mainly contributed by the participating peers, which could be desktop clients [6] or deskside servers in a much larger quantity than existing Grids [30]. The P2P Grids intend

* **Manuscript submitted to IEEE *IPDPS-2006, October 12, 2005*. This work was supported by NSF Grant ITR-0325409. Corresponding author is Kai Hwang at Email: kaihwang@usc.edu, Tel. 213 740 4470, and Fax: 213 740 4418.**

to merge the positive features from both P2P system and Grids. In particular, the in-and-out flexibility and fast search mechanisms [17] in P2P systems are explored for collective Grid computing. The ultimate goal of building P2P Grids is to integrate the P2P, Grid, and web services [10].

Killer applications of P2P Grids include both scientific computing and web services. Jobs can be executed at local client machines or outsourced to remote peer machines on a P2P interaction basis. The P2P operation is inherently insecure due to the anonymity among the peers [24], [26]. 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 [3], [8].

Table 1 compares the architecture, control, security, and applications of the three distributed computing models. This paper considers mainly structured P2P Grids with decentralized resources from either participating peers or brokered Grid resources. The P2P Grids may be built from extending existing desktop Grids or scale up from existing supercomputing Grids. This leads two classes of P2P Grids: Grids formed with PC desktops like the Entropia [6] and PC Grid [20] versus established Grids operating in a P2P setting like community Grids [10] and other emerging Grids [8].

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

Features	P2P Systems	Computational Grids	P2P Grids
Architecture and Connectivity	Flexible topology, scalable to millions of autonomous users	Static configuration with limited scalability	P2P flexibility with Grid resource sharing initiatives
Control and Operation Model	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 centralized or hierarchical job management	Support desktop, distributed Grid computing, and community services
Representative Systems	Chord (DHT) [27], CAN, Pastry, Tapestry, etc.	NSF TeraGrid , e-Science in UK [3], China Vaga Grid [29]	PC Grid [20], Entropia [6] Community Grid [10]

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 global reputation system. We describe the PowerTrust system construction and its configuration and updating algorithms in Section 5. We report extensive simulation results on P2P Grid performance in Section 6. Finally, we conclude with discussions and suggestions for further research work needed towards trusted P2P Grid computing.

2. Trusted P2P Grid Computing

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 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 [26]. 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 [2]. Peer-layer security relies on using reputation systems.

In the past, trust management in P2P systems was mainly supported by reputation systems built on top of peer feedbacks [1], [4], [7], [12], [14]. For P2P Grids, the reputation systems 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 [23] and selfish peers [13] 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 [4], [12], [18], [26], [32] 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 [10].

Buchegger and Budded [4] presented a reputation evaluation scheme based on Bayesian learning technique. The EigenTrust mechanism [12] aggregates global reputation by a distributed calculation of the Eigenvector of the trust matrix over the peers. Song, et al [25], [26]suggested to use a fuzzy-logic trust management system to model the uncertainties involved in P2P transactions. Xiong and Liu [32] 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 among peers, which may not agree with the reality in a P2P or P2P Grid environments.

2.2 A 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.

In a TON, the number of feedbacks a user sent to others is indicated by the *out-degree* of the peer node. The number of feedbacks a user received from others is represented as the *in-degree* of a peer node. We created the TON for modeling 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 fundamental causes: the *dynamic growth* and *preferential node attachment* [19]. The former allows the network to expand with any newly added nodes. The later allows the new node to interact selectively with existing reputable nodes in the system. These are common in any dynamic P2P systems.

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 sorting mechanism. This will ensure fast reputation convergence and defend against collusions by malicious peers. The simulation results show that

PowerTrust aggregates the global reputation faster with high accuracy and leads to higher success rates than the EigenTrust system in typical distributed job execution.

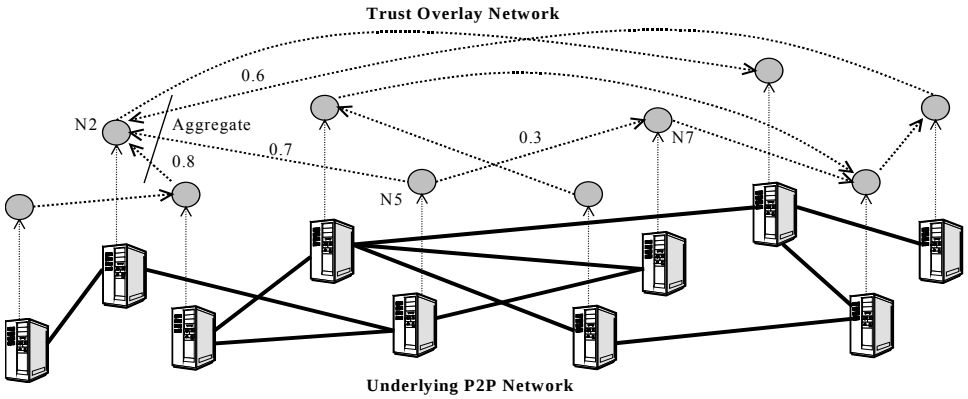


Figure 1. Graphical representation of the trust overlay network (TON) for a P2P Grid, where the nodes represent the peers and directed edges are labeled with local trust scores. The global reputation is aggregated from scores received on all incoming edges.

Unfortunately, there is no feedback information available in existing P2P systems. To model the properties of feedbacks in a real-life environment, we study the public-domain eBay reputation system. 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) [24] and l*ocality preserving hashing* (LPH) [5] concepts to build our PowerTrust system by leveraging on the feedback properties.

3. Implications from eBay Reputation Trace Data

Power-law distribution is an inherent property of P2P systems. For example, the content distribution in Gnutella is power-law distributed [21]. 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. In the following sections, we study the distribution of received feedbacks, or the node in-degree distribution in TON. Three important key parameters are identified in our study. The *feedback amount* of node i is denoted by d_i. *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 [18]. 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 an overall 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 108MB 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.

The above simple crawling method turns out to be inherently biased due to the sampling error. Apparently, the more feedback a user receives from others, the easier he or she will be crawled since the crawling process is random. Let p_d denotes the probability that the node with received feedback amount d is discovered by a given random crawl, we have $p_d = \sum^n_{i=1} d_i$, where d_i is the received feedback amount of node i and n is the network size of TON. For a power node that leaves k feedbacks for others, the probability that the node with received feedback amount d can be crawled from the power node is

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

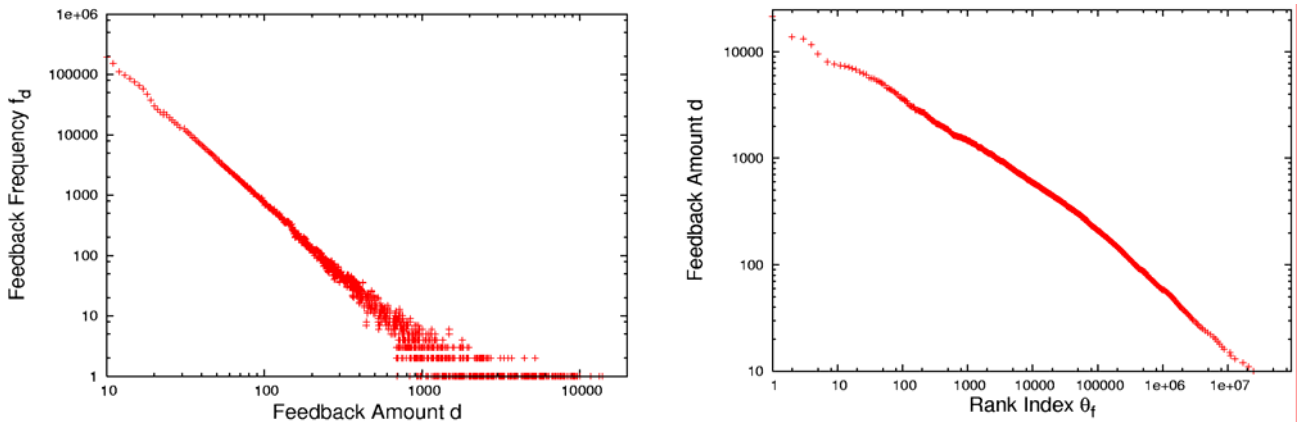
Let n_d be the original number of nodes with received feedback amount d in TON, and $\hat{n}_d$ be the number of nodes with received feedback amount d in the sampling dataset, we have: $\hat{n}_d = E(n_d)\times P_d$ or the following expected value:

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

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 the this procedure *recovery process*.

3.2 Feedback Distribution in eBay

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. So we approximate the total sum $\sum^n_{i=1} d_i$ by 80,000,000×68 = 5.24×10^9. We apply the recovery process to the sample eBay trace data and draw the original feedback distribution in Fig.2 (a). Only the nodes with received feedback amount larger than 10 were included, because users with less than 10 feedbacks are considered inactive. 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 shown in Fig.2. In other words, the node with a small number of feedbacks is common whereas the node with a large number of feedbacks is extremely 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.

4. Global Reputation Aggregation

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. Traditional method generates local trust scores using the sum of both positive ratings for successful requests and negative ratings for unsuccessful queries [18]. 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 for *node j*. If there is no link in TON 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, …., n, while $\left|\vec{V}^{(i)} - \vec{V}^{(i-1)}\right| > \varepsilon$, we compute the successive trust vectors recursively by:

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

This approach 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. After hopping for a while, the surfer is more likely located at some more reputable nodes. We define a *greedy factor* α as the eagerness probability of a random walker 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 the *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. As described in Section 3, the TON is a sparse power-law graph, the LRW does not cause heavy overhead because the resulting replication overhead is limited by the number of edges in a TON, which is linear in a sparse power law graph [15].

We analyze the aggregation convergence time by checking the number of iterations of Eq.(4). We generated 100 random graphs and 100 power-law graphs for each size to get the average speedup factor. The node degree distribution in the random graph is specified as follows:

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

where x is an arbitrary node, n is the graph size, and p = (Number of edges)$/n^2$.

Our experiment results show the LRW strategy greatly improves the convergence rate of both power-law graph and random graph, especially for power-law graph. The improvement comes from the random walker in a power-law graph can quickly hop towards highly reputable nodes, which keep a lot of trust information about their neighbors.

4.2 Locality Preserving Hashing (LPH)

Since node degrees in a typical TON are power-law distributed, there are some power nodes that have higher degree than others. These power nodes typically correspond to the most reputable peers in a P2P system. A distinction of our PowerTrust system is to leverage more on the power nodes to aggregate the global reputations. Considering a large-scale P2P system with poor reliability and frequent dynamic changes in its configuration, we propose a fully distributed sorting mechanism to dynamically select the m most reputable power nodes in the system.

PowerTrust uses a *Distributed Hash Table* (DHT) such as Chord [27] that offers scalable key-based lookup for distributed resources. As in EigenTrust [11], 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. 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 wrong global reputations.

To select the m most reputable nodes, our distributed sorting mechanism applies *locality preserving hashing* (LPH) to sort all nodes w.r.t their global reputations. Cai, et al [5] suggested to use 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 $[H(v_k), H(v_j)]$.

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 H 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.

Figure 3 presents an example 5-node PowerTrust system built on top of Chord with 4-bit circular hash space. Node *N15* is the score manager of node *N2* whose global reputation is 0.2. Node *N*15 hashes the reputation value 0.2 using a simple LPH function $H(x) = 32x$. The resulted hash value is 6.4. Node *N*15 sends out a *Sort_Request*{*key*=6.4, (0.2, *N*2, *N15*)} message as a Chord insert request, which is routed to node *N*8 . Node *N*8 stores the triplet (0.2, *N2*, *N*15), since it is the successor node of hash value 6.4. All the pairs and triplets stored on different nodes are shown in Fig.4. For simplicity, we illustrate how to find the highest reputation node corresponding to the case of $m = 1$.

Node *N*2 is the successor node of the maximum hash value 15 and is responsible for hash values in the range $(15, 16] \cup [0,2]$. 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. Multiple LPH functions could be used to prevent cheating by the participating peers.

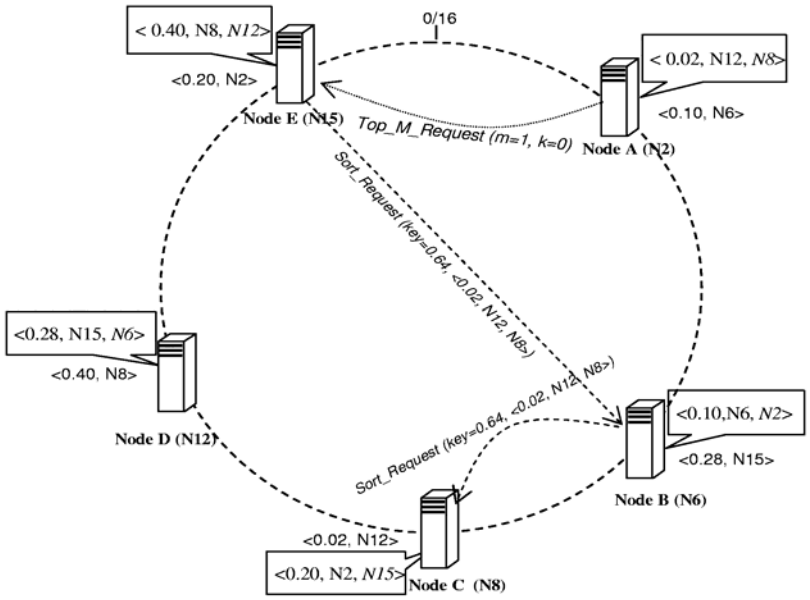


Figure 3 Distributed sorting based on locality preserving hashing over a DHT-based P2P system with 5 nodes (peers) in the PowerTrust system

5. P2P Reputation System Construction

Inspired by EigenTrust [12], the PowerTrust system is improved from using the LRW and distributed sorting 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 1 specifies the construction of the PowerTrust system in the first round of global reputation aggregation.

Algorithm 1 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

In algorithm 1, 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 as the number of iterations in algorithm 1. This rate is upper bounded by λ_2 / λ_1, where λ_1 and λ_2 are the first and second largest eigenvalues of the trust matrix *R* defined over the TON. Thus, 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 [11]. In other words, the power-law distribution of TON will guarantee the convergence at the very first round of global reputation aggregation. After first round aggregation, the score managers collaborate with each other to find the power nodes using the distributed sorting mechanism. Because the trust matrix *R* is dynamically changing with new peers joining and new transactions performed, the global reputation should be updated periodically, especially more often on the power nodes.

5.2 Global Reputation Updating Procedure

The distributed updating of global reputation aggregation leverages on the use of the power nodes. 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.

Algorithm 2: Distributed global reputation updating procedure

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**
    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

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:

$$T = (1-\alpha)R^T + \alpha P^T \quad \textbf{(6)}$$

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 defined in Section 4.1. The purpose of the update is to control the gap between the first and second largest eigenvalues of the transition matrix T, given the largest eigenvalue $\lambda_1 = 1$ and the second largest eigenvalue as $\lambda_2 \leq 1 - \alpha$ [16].

6. Simulated P2P Grid Performance Results

We have evaluated the PowerTrust system. Performance results are reported below in three aspects: reputation convergence rate, query success rate in distributed file sharing, and PSA benchmark results on simulated P2P Grid performance.

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 the transaction success rate in P2P file sharing application. The third one shows the performance of job execution in a P2P Grid by measuring the makespan and job success rate 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. Our discrete-event driven simulator was written in C.

We adjust the simulator with varying parameters to run different experiments. Our initial simulated TON for the P2P Grid 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 users: 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 other peers very low.

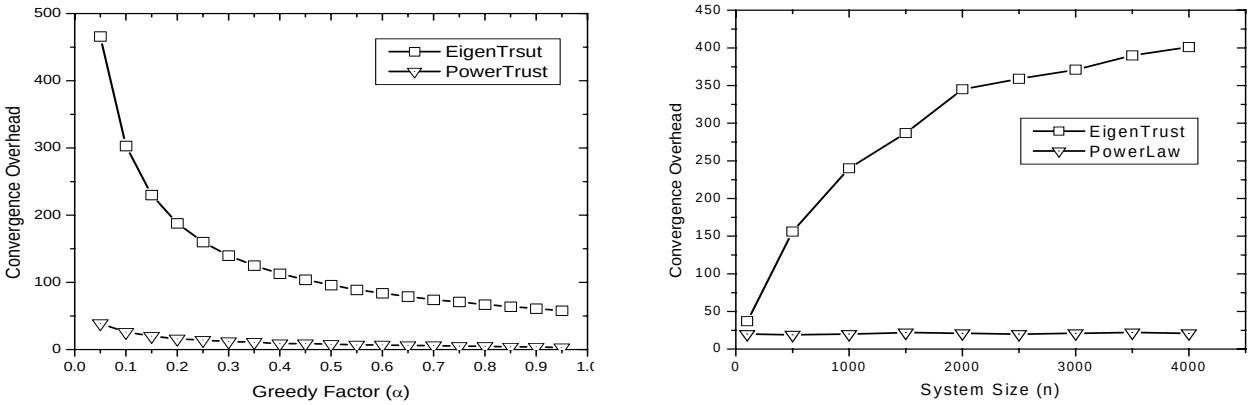
We select 1% power-nodes over the total number of nodes in a TON. *Convergence overhead, makespan, job success rate* are chosen as the metrics for performance evaluation. We compare below the performance between our PowerTrust system and Stanford EigenTrust system [12] over the P2P file sharing application and PSA workload, where The PSA (*Parameter-sweep application* (PSA) benchmark [3] is often used in Grid simulations on large number of independent jobs in parallel. The execution model processes M independent jobs (each has the same task over different dataset) on N distributed sites, where the job number M (say 4,000) is much larger than the site number *N (say 100)*.

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 most P2P systems, peers are dynamically joining and leaving. We simulated the case of 1 power-nodes and pre-trust nodes leaving from PowerTrust and EigenTrust system, respectively. 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 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 (a), when α increases from 0.15 to 1. Figure 4(b) 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 for a P2P system of 1,000 peers

(b) Effect of system size *n* with a fixed *greedy factor* α = 0.15

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

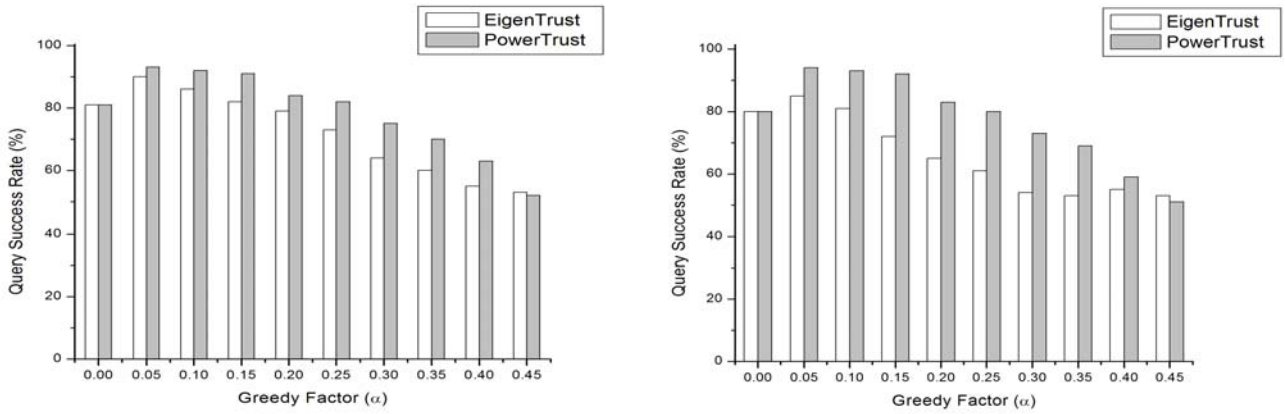
In both plots, the PowerTrust system outperforms the EigenTrust system sharply. The EigenTrust system converges very slowly or even cannot guarantee its convergence when pre-trust nodes are allowed to leave the system freely. In the PowerTrust system, the power nodes will be re-elected after each aggregation round. Based on the distributed sorting 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 and almost constant overhead in using the PowerTrust system makes it attractive in performing highly scalable P2P Grid applications.

6.3 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 [12]. 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 β = 1.2. Each peer is assigned with a number of files. Figure 5 shows the query success rate in using the two 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. The query success rate is measured by the percentage of success queries over the total number of queries issued. 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 the cases of using power nodes or pre-trust nodes. Figure 5(a) plots the result against increasing value of greedy factor α with only one round of aggregation. Figure 5(b) shows the results after ten rounds of aggregation. The PowerTrust outperforms the EigenTrust in almost all cases. In the best case after 10 rounds with a fixed low α = 0.15, the PowerTrust has 92% query success rate, about 39% higher than 65% query success rate of the EigenTrust system. This result implies that the PowerTrust has higher sustained performance in distributed file sharing applications than that of the EigenTrust system. When the peer greedy factor increases to a high value α = 0.45, both systems drop to less than 50% query success rate.

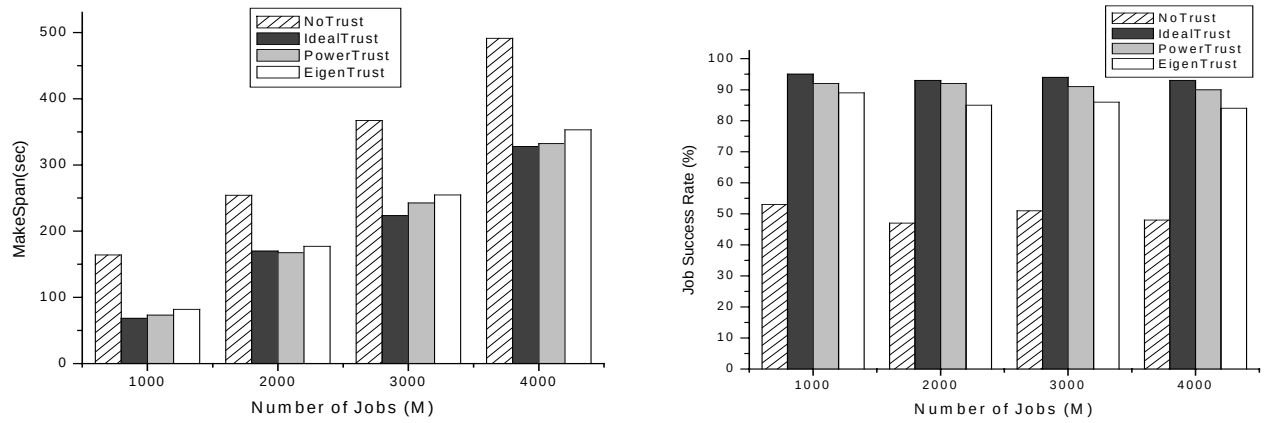


(a) After the first round global reputation aggregation **(b) After ten rounds of global reputation aggregation**

Figure 5. Query success rates of two P2P reputation systems: DynaTrust vs. EigenTrust, after 1 or 10 rounds of global reputation aggregation

6.4 Performance Result over the PSA Workload

In this section, we use the following two metrics to simulate the PowerTrust performance in large-scale P2P Grid job execution: The *makespan* is measured by the maximum time, Max$\{c_i\}$, to execute M jobs in parallel, where c_i is the completion time of job J_i. The *job success rate* is measured by S_{rate}= 1- M_{fail}/M, where M_{fail} accounts the number of failed jobs. The job arrivals assumed a random Poisson distribution on each Grid site. We have simulated up to 4,000 jobs distributed over 100 Grid peer sites. The results are plotted in Figure 6.



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

Figure 6. PSA benchmark performance results on a simulated P2P Grid configuration with 100 peer-contributed resource sites

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 as defined in section 6.3. 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.

We have assumed an average job execution time 5 sec/job and an average 2 jobs/sec arrival rate. A job is executed if it is rejected no more than 3 times. Figure 6 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 bars and white bars correspond to using the PowerTrust and EigenTrust systems, respectively.

In Fig.6 (a), the job makespan of all 4 reputation systems increases with the job number. Figure 6(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.

7 Conclusions and Further Work

In this paper, 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 the overlay graph. This power-law distribution is not restricted to eBay reputation data, but 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 aggregate local trust scores to yield global reputations by leveraging on the power-law property. The system is built with *locality preserving hash* (LPH) functions, which can be easily 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 LPH function and the LRW strategy in system construction and update processes. These advantages help accelerate the reputation aggregation, responses to trust enquiries , and security binding in both P2P systems and P2P Grids, significantly.

Based on the results reported, we reveal the following advantages of structured P2P Grids with distributed control over the computational Grids with centralized management.

- P2P Grids are more efficient in the way that it broadcast messages and offers higher scalability, and application flexibility than the static Grid configurations.
- 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.
- P2P Grids have to deal with changing IP addresses like roaming users or even unknown IP addresses from firewalls. This may give more protection in privacy and anonymity [31].
- 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.

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 [33]. Second, we need to explore new killer applications of the P2P Grids beyond the file sharing and PSA applications reported here [30]. Third, the distrust problem will become even more complex in real-life selfish Grids [13], [25]. These three issues all 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.

Acknowledgements: This work was supported by NSF ITR Grant ACI-0325409. We appreciate the comments by Dr. Ricky Kwok of Hong Kong University, Dr. Jianping Pan of University of Victoria, Canada, and our colleagues, Min Cai and Shanshan Song, at the USC Internet and Grid Computing Laboratory. Their valuable suggestions have greatly improved the quality and readability of this paper.

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

[3] F. Berman, J. Fox, and T. Hey (Editors), "Grid Computing: Making The Global Infrastructure a Reality", *Wiley Series in Communication Networking and Distributed Systems,* 2003.

[4] 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*, 2004.

[5] M. Cai, M. Frank and P. Szekely, "MAAN: A multi-attribute addressable network for grid information services", *Journal of Grid Computing*, 2004.

[6] A. Chien, et al, "Architecture and Performance of an Enterprise Desktop Grid System", *Journal of Parallel and Distributed Computing,* 2001.

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

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

[9] I. Foster and A. Iamnichi, "On Death, Taxes, and Convergence of P2P and Grid Computing", *IEEE Internet Computing,* Jan. 2003.

[10] 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.

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

[12] S. Kamvar, M. Schlosser, and H. Garcia-Molina, "The Eigentrust Algorithm for Reputation Management in P2P Networks", *ACM WWW'03,* pages 640-651, 2003.

[13] 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.

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

[15] M. Mihail, A. Saberi, P.Tetali, "Random Walks with Lookahead in Power Law Random Graphs", *WWW'04.*

[16] 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.

[17] S. Ren, L. Guo, S. Jiang, and X. Zhang, "SAT-Match: A Self-Adaptive Topology Matching Method to Achieve Low Lookup Latency in Structured P2P Overlay Networks", *Proc. IPDPS 2004.*

[18] 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.

[19] M. Ripeanu, I. Foster, and A. Iamnitchi, "Mapping the Gnutella Network: Properties of Large-scale Peer-to-Peer Systems and Implications for System Design", *IEEE Internet Computing,* Vol. 6, 2002.

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

[21] 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.

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

[23] S. K. Shah, K. Ramamritham, and P. Shenoy, "Resilient and Coherence Preserving Dissemination of Dynamic Data using Cooperating Peers", *IEEE Transactions on Knowledge and Data Engineering,* Vol. 16, No.7, July 2004, pp.799-812.

[24] E. Sit and R. Morris, "Security Considerations for P2P Distributed Hash Tables", *Proc. IPTPS 2002*.

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

[26] 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.*

[27] 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,* 2001.

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

[29] 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*, 2003.

[30] P. Uppiluri, N. Jabisetti, U. Joshi, and Y. Lee, "P2P Grid: Service-Oriented Framework for Distributed Resource Management", *IEEE International Conference on Web Services,* 2005.

[31] L. Xiao, Z. Xu and X. D. Zhang, "Low-cost and Reliable Mutual Anonymity Protocols in Peer-to-Peer Networks", *IEEE Tran. on Parallel and Distributed Systems,* Vol. 14, No. 9, Sept. 2003, pp. 829 – 840.

[32] L. Xiong and L. Liu, "PeerTrust: Supporting Reputation-based Trust for Peer-to-Peer Electronic Communities", *IEEE Transaction on Knowledge and Data Engineering,* pages pp. 843-857, 2004.

[33] 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.

Biographical Sketches:

Runfang Zhou received the B.S. and M.S. in computer science from Southeast University in China. 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 and wireless 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.