REPUTATION MANAGEMENT SYSTEM FOR PEER-TO-PEER COMMUNITIES

Punitha S.[1], Thompson.S[2]

[1]Lecturer, Amity school of Engineering and Technology, Bijwasan, New Delhi
[2]Student, Veltech Multitech Dr.Rangharajan Dr.Sakunthala Engineering College
Email: [1]punithaarputhaswamy@gmail.com

Abstract

Abstract *Peer-to-Peer* (P2P) reputation systems are essential to evaluate the trustworthiness of participating peers and to combat the selfish, dishonest, and malicious peer behaviors. The system collects locally-generated peer feedbacks and aggregates them to yield the global reputation scores. Surprisingly, most previous work ignored the distribution of peer feedbacks. We use a *trust overlay network* (TON) to model the trust relationships among peers. After examining the eBay transaction trace of over 10,000 users, we discovered a power-law distribution in user feedbacks. Our mathematical analysis justifies that power-law distribution is applicable to any dynamically growing P2P systems, either structured or unstructured. We develop a robust and scalable P2P reputation system, *PowerTrust*, to leverage the power-law feedback characteristics. The PowerTrust system dynamically selects small number of power nodes that are most reputable using a distributed ranking mechanism. By using a look ahead random walk strategy and leveraging the power nodes, the PowerTrust significantly improves in global reputation accuracy and aggregation speed. PowerTrust is adaptable to dynamics in peer joining and leaving and robust to disturbance by malicious peers. Through P2P network simulation experiments, we find significant performance gains in using PowerTrust. This power-law guided reputation system design proves to achieve high query success rate in P2P file-sharing applications. The system also reduces the total job make span and failure rate in large-scale, parameter-sweeping P2P Grid applications. A peer to peer network exploits diverse connectivity between participants in a network and the cumulative bandwidth of network participants rather than conventional centralized resources where a relatively low number of servers provide the core value to a service or application. A pure peer to peer network doesn't have the notion of clients or servers, but only equal peer nodes that simultaneously function as both clients and servers to the other nodes on the network. Peers act as equals, merging the roles of clients and server. There is no central server managing the network. There is no central router.

Keywords: Peer-to-Peer system, overlay network, distributed hash table, reputation system, eBay trace dataset, distributed file sharing, P2P Grids, PSA benchmark, and system scalability.

I. INTRODUCTION

In recent years, *peer-to-peer* (P2P) computing has gained its popularity in many large scale distributed applications over the Internet. These include distributed file-sharing, digital content delivery, and P2P Grid computing. Despite the demand of robustness and scalability of P2P systems, the anonymous and dynamic nature of peer activities make them often very vulnerable to abuses by selfish and malicious peers. For example, most P2P file-sharing networks, e.g. Gnutella, consist of autonomous peers with special self-interests. There is no Efficient way to prevent malicious peers from joining the open networks. To encourage resource sharing among Peers and combat malicious peer behaviors, reputation management is essential for peers to assess the trustworthiness of others and to selectively interact with more reputable ones. Without an efficient reputation Management facility, peers will have little incentive to contribute their computing or bandwidth resources. Identifying trustworthy peers is especially necessary in commercial P2P applications, such as P2P auctions, trusted content delivery, pay-per-transaction, and P2P service discovery. A *reputation system* calculates the global reputation score of a peer by considering the opinions (i.e. *feedbacks*) from all other peers who have interacted with this peer. After a peer completes a Transaction, e.g. downloading a music file, the peer will provide his or her feedback for other peers to use in future transactions. By making the reputation scores publicly available, peers are able to make informed decisions about which peers to trust.

The eBay reputation system is a simple and successful one, since it has a centralized authority to manage all user feedback scores. However, in an open and decentralized P2P system, peers will not have any S. Punitha. et al: Reputation Management System for Peer-to-Peer Communities centralized authority to

maintain and distribute reputation information. Instead, most existing P2P reputation systems calculate the global reputation scores by aggregating peer feedbacks in a fully distributed manner. Building an efficient P2P reputation system is a challenging task due to several intrinsic requirements of large-scale P2P systems. Listed below are six key issues that should be addressed in the design of a cost-effective P2P reputation system.

- ***High accuracy***. To help distinguish reputable peers from malicious ones, the system should calculate the reputation scores as close to their real trustworthiness as possible.
- ***Fast convergence speed***. The reputation of a peer varies over time. The reputation aggregation should converge fast enough to reflect the true changes of peer behaviors.
- ***Low overhead***. The system should only consume limited computation and bandwidth resources for peer reputation monitory and evaluation.
- ***Adaptive to peer dynamics.*** Peer joins and leaves an open P2P system dynamically. The system should adapt to this peer dynamics instead of relying on predetermined peers.
- ***Robust to malicious peers***. The system should be robust to various attacks by both independent and collective malicious peers.
- ***Scalability***: The system should be able to scale to serve a large number of peers in term of accuracy, convergence speed, and extra overhead per peer.

II. RELATED WORKS

A formal treatment of trust and reputation was given by Aberer and Despotovic in the context of P2P networks. Their approach is based on a decentralized storage method (P-Grid). The information provided by P-Grid is used to assess the probability that an agent will cheat in the future. This approach suffers from several shortcomings, e.g., trust is evaluated only according to referrals from neighbors, not based on all information in the system.

Buchegger and Budded presented a reputation evaluation approach based on Bayesian learning technique. In their approach, the first-hand information is exchanged frequently and the second-hand information is merged, if it is compatible with current reputation rating. Xiong and Liu presented an approach that avoids aggregation of the individual interactions. Their PeerTrust system computes the trustworthiness of a given peer as the average feedback weighted by the scores of the feedback originators.The limitation of this approach is that the computation convergence rate in large-scale P2P systems is not provided. The five factors used in their trust model must be retrieved with a heavy overhead.

III. OUR POWERTRUST SYSTEM APPROACH

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

(A) The PowerTrust System Concept

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

Figure 1 shows the major building blocks in a PowerTrust system. First, a *trust overlay network* (TON) is built on top of all peers (nodes) in a P2P system. All peers evaluate each other, whenever a transaction takes place between a peer pair. Therefore, all peers send *local trust scores* among themselves, frequently. These scores are considered as the raw data input to the PowerTrust system.

The system supposes to aggregate the local scores to calculate the global reputation score of each participating peer. All global scores form a *reputation vector*, $V = (v_1, v_2, v_3, \ldots, v_n)$, which is the output of the PowerTrust system. All global scores are normalized with $\Sigma_i v_i = 1$, where $i = 1, 2, \ldots, n$ and n is

the TON network size. The system is built with five functional modules as shown in Fig.1. The *regular random walk* module supports the *initial reputation aggregation.* The *look-ahead random walk* (LRW) module is used to update the reputation score, periodically.

To this end, the LRW also works with a *distributed ranking module* to identify the power nodes. The system leverages the power nodes to update the global scores reputation. PowerTrust achieves high aggregation speed and accuracy, robustness to resist malicious peers, and high scalability to support large-scale P2P applications. We will discuss the details of these functional modules in subsequent sections. S. Punitha. et al: Reputation Management System for Peer-to-Peer Communities

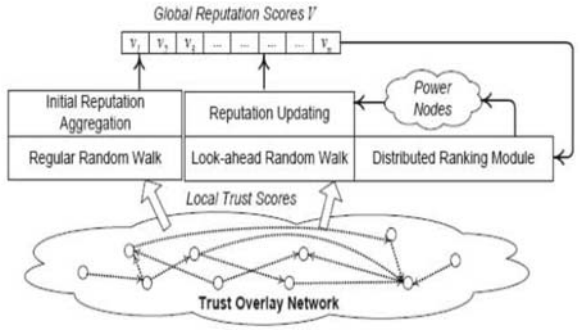

Figure 1 Functional modules in the PowerTrust System and the control flow pattern in local trust score collection and global reputation aggregation.

(B) Trust Overlay Network (TON)

A TON is a virtual network on top of a P2P system. We represent a TON by a directed graph in Fig.2. The graph nodes correspond to the peers. The directed edges or links are labeled with the feedback scores between two interacting peers. The feedback score is issued by a peer (source of the link) for the service provided by the interacting peer (destination of the link). For example, node *N*5 after downloading music files from nodes *N*2 and *N*7 issues the feedback scores, 0.7 and 0.3, to the two provider nodes, respectively.

If a node gets more than one service from the same provider, this consumer generates a newly updated score after each transaction. Our system can incorporate different methods to generate feedback scores, such as Bayesian learning. Each node *Ni* is rated with a *global reputation score* v_i. This global reputation is aggregated from local trust scores weighted by the global reputations of all in-degree neighbors.

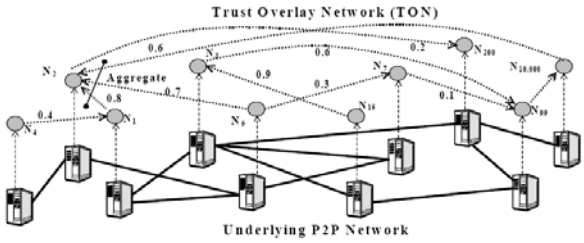

Figure 2. A trust overlay network (TON) for a P2P system with 10,000 nodes, where a node represents a peer and an edge is labeled with the peer feedback score for the service provided.

IV. POWER-LAW DISTRIBUTION OF PEER FEEDBACKS

Power-law distribution is well known in Internet community. We study the public domain eBay reputation system to verify the conjecture that the feedback distribution of a typical P2P reputation system follows the power-law. In eBay, feedback is generated after every transaction. However, nearly 90% seller-buyer pairs conducted just one transaction during the past 5 years. So the node in-degree in TON is approximated by the number of feedbacks received.

Three key parameters are used: The *feedback amount* of a node *i* is denoted by d_i, which is the *indegree* of this node. For example, node N_2 in Fig.1 has an in-degree of 3, meaning 3 feedback scores received. *Feedback frequency* f_d is the number of nodes with feedback amount *d*. The *ranking index* θ *d indicates the order of d* in a decreasing list of feedback amounts.

(A) Collection Procedure of eBay Reputation Data

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

eBay user has a time-varying reputation by summing up all transaction scores received up to the current time.

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

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

$$Qd = 1 - (1 - p_d)^4 = 1 - (1 - d/\Sigma_{1=0}^{n} d_i)^k \quad \ldots (1)$$

Let n_d be the initial number of nodes with feedback amount d in the eBay TON. Let n_d be the number of nodes with feedback amount d in the sample dataset. We calculate $n = E_n \times Q$.

V. POWERTRUST SYSTEM CONSTRUCTION

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

(A) Look-ahead Random Walk (LRW)

In our PowerTrust system, feedback scores are generated by Bayesian learning or by an average rating based on peer satisfaction. Each node normalizes all issued feedback scores. Consider the *trust matrix* $R = (r_{ij})$ *defined over an* n-node TON, where r_{ij} is the *normalized local trust score* defined by $r_{ij} = s_{ij}/\Sigma_j s_{ij}$, and s_{ij} is the most recent feedback score that node i rates node j. If there is no link from node i to node j, s_{ij} is set to 0. Therefore, for all $1 \le i, j \le n$, we have $0 = 1\ r_{ij}$ and $i\Sigma_n j = 1\ r_{ij} = 1$. In other words, matrix R is a stochastic matrix, in which all entries are fractions and each row sum equals 1. This demands that the scores issued by the same node to other peers are normalized. All global reputation scores $v_i$for n nodes form a *normalized reputation column vector* $V = (v_i)$, where $\Sigma_i v_i = 1$. For a system of n nodes, we can simply assume $v_i = 1/n$ to start with. For all $t = 1, 2, \ldots, k$, while $| V(i) - V(i-1) | > \varepsilon$, we compute the successive reputation vectors recursively by:

$$V_{(t+1)} = R^T \times V^{(t)} \quad \ldots (2)$$

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

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

The efficiency of the LRW strategy is analyzed below. Each peer node aggregates the first-hand local trust scores from its neighbors, the *enhanced trust matrix* S by using the LRW strategy is computed by $S = R2$. Define a *speedup factor* by comparing the number of convergence

iterations for a regular random walk to that of LRW. Table 2 shows the speedup factor for various graph sizes.

We generated 100 random graphs and 100 Power-law graphs to make the comparison. The node degree distribution of a random graph is specified by:

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

Where N is an arbitrary node, n is the graph size, and p = (Number of links) / $n2$. As shown in Table 2, the LRW strategy greatly improves the convergence rate in both Power-law graph and random graph. The Power-law graph has higher speedup in all network sizes. The improvement comes from the random walker in a power-law graph can quickly hop towards highly reputable nodes, which can preserve a lot of useful reputation information.

Table 2: Speedup Factors of using Look-ahead Random Walk Strategy in Random Graphs and Power-law Graphs TON Size Random Graph Power-law Graph

TON Size	Randon 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

(B) Distributed Ranking Mechanism

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

Power Trust uses a *Distributed Hash Table* (DHT) such as Chord to implement the distributed ranking mechanism. As in EigenTrust, every node has a score manager that accumulates its global reputation. When a new node i joins the system, node j is assigned as the score manager of node i if node j is the successor node of ki, where ki is the hash value of the unique identifier of node i by a pre-defined hash function. All other nodes can access the global reputation of node i by issuing a lookup request with key equal to ki. Different hash functions can be used to have multiple score managers for each node in case the S. Punitha. et al: Reputation Management System for Peer-to-Peer Communities

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

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

Output: m most reputable nodes

Procedure:

For each score manager j, suppose it is the score manager of node i **do** hash reputation value v_i to a hash value $H(v_i)$ using a LPH function insert the triplet (v_i, i, j) to the successor node of $H(v_i)$.

End for Initialize

node x = successor node of the maximum hash value Set p = the number of triplets with highest reputation values stored in node x

Loop: if $p > m$ then return; Else

node x sends a message to its predecessor node y to find the next $m - p$ highest reputation triplets node x = node y $m = m - p$ p = number of triplets stored in node y

Goto loop End if

Suppose node j is the score manager of node i, it stores a pair (v_i, i) for node i, where v_i is the global reputation of node i. Node j hashes the reputation value v_i using a LPH function to a hash value $H(v_i)$ and inserts the triplet (v_i, i, j) to the successor node of $H(v_i)$. This process repeats recursively until the m highest reputation triplets are found. Basically,

distributed reputation ranking requires two different hash overlays. One assigns peers to their score managers and another rank the peers by their global reputation scores.

VI. SYSTEM PERFORMANCE ANALYSIS

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

(A) Simulation Setup and Experiments Performed

Three sets of simulated P2P experiments were performed. We use the *convergence overhead* to measure the aggregation speed. We use peer dynamics to enable system scalability We use *ranking discrepancy* to measure the accuracy and *RMS aggregation error* to quantify the system robustness to malicious peers. Our simulated TON for a P2P system was a fully connected Power-law graph, consisting of 1,000 nodes initially with a maximum node degree $d_{max} = 200$ and a feedback factor $\beta = 2.4$. We assume 80% honest peers and 20% malicious peers in the simulated P2P system.

Table 3(a) Relationship among α, Convergenc Overhead and Ranking Discrepancy in a PowerTrust Reputation System over 1000 Peers Without malicious peers

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

Table 3(b) Relationship among α, Convergenc Overhead and Ranking Discrepancy in a PowerTrust Reputation System over 1000 Peers Without malicious peers

Without 20% maliclous peers		
Greedy factor α	**Convergence overhead**	**Ranking Discrepancy**
0	79	0.275
0.05	40	0.155
0.10	24	0.201
0.15	20	0.217
0.20	15	0.234
0.30	12	0.253
0.45	10	0.331

Table 3 shows the relationship between α, convergence overhead and ranking fiscrepancy in a 1000-node P2P reputation system under two network conditions: without any malicious peers and with 20% malicious peers. When there is no malicious peer in the S. Punitha. et al: Reputation Management System for Peer-to-Peer Communities system, as α increases, there is a tradeoff between convergence overhead and ranking discrepancy. With 20% malicious peers, the ranking discrepancy first decreases then increases, as α increases. So we choose $\alpha = 0.15$ as a default value to balance the tradeoff between efficiency and accuracy.

(B) Reputation Convergence Overhead

The *convergence overhead* is measured as the number of iterations before the global reputation convergence. Convergence means that distance between two consecutive reputation vectors is smaller than the threshold. The Eigen Trust approach relies on a few pre-trusted nodes to compute the global reputations. We report in Fig.5 the effects of different greedy factor α and system sizes *n* on the variation of the convergence overhead.

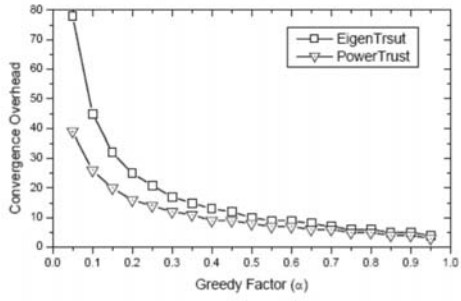

Fig. (a) *Disallowing departure of power nodes in PowerTrust or pre-trusted nodes in EigenTrust system*

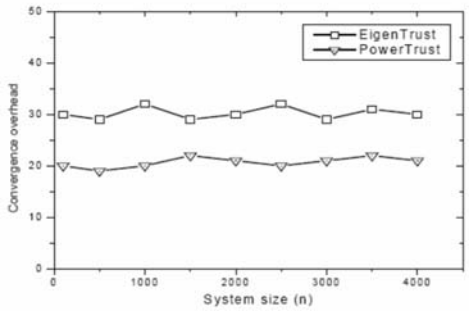

Fig. *(b) Disallowing departure of power nodes or pre-trusted nodes with a fixed* $\alpha = 0.15$

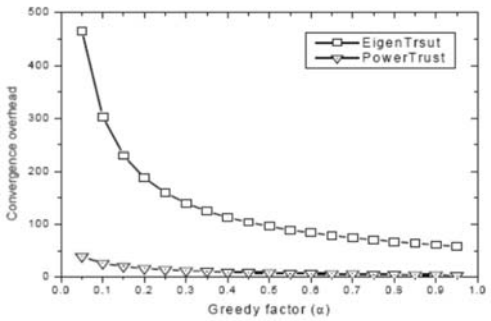

Fig. *(c) Allowing departure of power nodes in PowerTrust or pre-trusted nodes in Eigen Trust system*

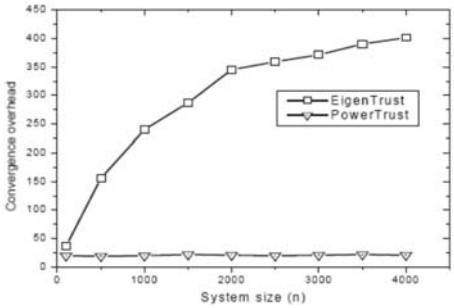

Fig. *(d) Effect of system size n with departure of power nodes or pre-trusted nodes under a fixed*

Figure 3. Convergence overhead of two reputation systems under varying peer greedy factor and increasing P2P system size

For all fairness, we choose the same number of power nodes equal to that of pre-trusted nodes used in EigenTrust. Figure 3(a, b) shows the convergence overheads for two reputation systems, assuming no pre-trusted node or power node leaving the P2P network. We observe the slight saving of iteration count in PowerTrust as shown in Fig.3(a).

The overhead drops to the same level as *á* increase toward 1. Figure 3(b) shows small fluctuation of the convergence overhead as the system size increases. In the case of a low $\alpha = 0.15$, we see an approximately 50% reduction in convergence overhead in using PowerTrust over EigenTrust system. The overheads in both systems S. Punitha. et al: Reputation Management System for Peer-to-Peer Communities are only moderately sensitive to the variation in network size. In Fig.3(c,d), the power nodes in PowerTrust and the pre-trusted node in EigenTrust are allowed to leave freely. These two plots show significant widening of the overhead gap between the two systems. We observe a sharp drop of iteration count in using PowerTrust to a flat small number less than 50 in Fig.3(c), when *á* increases from 0.15 to 1, while the EigenTrust still requires more than 100 iterations to converge. Figure 3(d) shows that our PowerTrust system has almost a flat low convergence overhead, independent of the system size under the default value of $\alpha = 0.15$.

The EigenTrust system overhead can reach as high as 400 iterations as the system increases to 4,000 nodes. In both plots, the PowerTrust system outperforms the EigenTrust system sharply. The EigenTrust system converges very slowly. The system cannot guarantee its convergence, when the pre-trusted nodes are allowed to leave the system freely. In the PowerTrust system, the power nodes are re-elected after each aggregation round. Based on the distributed ranking mechanism, the score managers of the departing power nodes notify the system to replace them timely with other more qualified power nodes. The decrease of computation overhead means significant traffic reduction on the network and less work for all peers involved. The low overhead in using the PowerTrust system makes it attractive in performing

highly scalable P2P applications, including P2P Grids as reported in.

(C) Reputation Ranking Discrepancy

To estimate the accuracy of the aggregated global reputation, we rank the peers by their global reputation scores. We measure below the *ranking discrepancy* between the estimated ranking and the actual ranking. The discrepancy comes mainly from greedy factor á and malicious peers reporting false trust scores. We use *normalized Euclidean distance* to measure the ranking discrepancy. During each round of reputation aggregation, we assume 100 new peers joining the system and transacting with existing peers. We refer each aggregation round to one full convergence of reputation vector computations. The probability of an interaction between nodes *i* and *j* is determined by the ratio $d_i d_j / \Sigma_{k=1}^{n} d_k$, where d_i and d_j are the corresponding node degrees. This property ensures that the growing TON follows the power-law connectivity.

VII. P2P APPLICATION BENCHMARK RESULTS

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

(A) Query Success Rate in Distributed File Sharing

We have applied the PowerTrust system on simulated P2P file-sharing applications. We choose the same query model used by Marti and Garcia-Molina. There are over 100,000 files in our simulated P2P systems. The number of copies of each file in the system is determined by a content Power-law distribution with $\beta = 1.2$. At each time step, a query is randomly generated at a peer and completely executed before the next query/time step.

The query distribution determines which file each query search for. We rank the queries according to their popularity. We use a query Power-law distribution with $\beta = 0.63$ for queries ranked 1 to 250 and $\beta = 1.24$ for the remaining lower ranking queries. This distribution models the query popularity in existing P2P systems. The query success rate of EigenTrust drops from 85% to 50% as the round number increases. This is due to the fact that pre-trusted nodes cannot cope with the dynamic variation of the peer reputations. The EigenTrust query success rate drops to 50% low, equal to that of a no trust system after 17 rounds of reputation aggregations.

VIII. CONCLUSIONS AND FURTHER WORK

In this paper, we report the design experiences and simulated performance of a new P2P reputation system, PowerTrust. Specifically, our contributions are summarized in four aspects, Power-law distribution of peer feedbacks -We developed a trust overlay network model for analyzing the feedback properties of P2P reputation systems. By collecting real-life data from eBay, we confirmed the power-law connectivity in TON graph. This power-law distribution is not restricted to eBay reputation system. Our mathematic analysis justifies its applicability to general dynamic P2P systems.

For further work, we suggest the following research task to solve the peer collusion problem, to extend the current PowerTrust system to work on unstructured P2P system as well, and to explore new killer P2P applications supported by reputation systems, Coping with peer abuses and selfishness -Various malicious behavior models should be investigated to secure P2P system applications. New mechanisms are needed to deal with intrusions, free riders, black mouths, collusions, and selfishness of peers. Game theoretic studies and benchmark studies are recommended. S. Punitha. et al: Reputation Management System for Peer-to-Peer Communities

REFERENCES

[1] K. Aberer and Z. Despotovic, “Managing Trust in a Peer-2-Peer Information System,” *Tenth International Conf. on Information and Knowledge Management,* New York, 2006.

[2] L. A. Adamic, “Zipf, Power-laws and Pareto – a ranking tutorial”, http://www.hpl.hp.com/research /idl/papers/ranking/ranki ng.html, HP Labs, CA. 2004.

[3] S. Buchegger and J.-Y. L. Boudec, “A Robust Reputation System for P2P and Mobile Adhoc Networks”, *Second Workshop on Economics of P2P Systems*, Boston, June 2004.

[4] 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.

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

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

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

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

[9]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.

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

S. Punitha got her ME degree in Computer Science from Easwari Engineering College, Chennai,India. which is affliated to Anna University in 2008. She got her BE degree in Computer Science from PMR Insitute of Technology, Chennai, India in 2004. She worked for Rajalakshmi Engineering college,Chennai,India.Currently she is working as Lecturer in the Department of Information Technology in Amity School of Engineering and Technology, NewDelhi, India which is affliated to Indraprastha University. Her Research interests are Wireless Networks and Mobile Computing.