Journal of King Saud University – Computer and Information Sciences (2015) **27**, 315–322

King Saud University

**Journal of King Saud University – Computer and Information Sciences**

www.ksu.edu.sa
www.sciencedirect.com

# HonestPeer: An enhanced EigenTrust algorithm for reputation management in P2P systems

**Heba A. Kurdi**

*Computer Science Department, King Saud University, Saudi Arabia*

Received 13 March 2014; revised 22 June 2014; accepted 23 October 2014
Available online 20 June 2015

**KEYWORDS**

P2P;
EigenTrust;
Reputation management;
Trust management;
File sharing

**Abstract** The visible success of the Peer to Peer (P2P) paradigm is associated with many challenges in finding trustworthy peers as reliable communication partners. Reputation management systems are emerging in the face of these challenges. The EigenTrust reputation management system is among the most known and successful reputation systems. On the other hand, a main drawback of this system is its reliance on a set of pre-trusted peers which causes nodes to center around them. As a consequence, other peers are ranked low despite being honest, marginalizing their effect in the system. To tackle this problem, this paper proposed enhancing the EigenTrust algorithm by giving peers with high reputation values (honest peers) a role in calculating the global reputation of other peers. Rather than solely depending on the static group of pre-trusted peers, the proposed algorithm, HonestPeer, selects the most reputable nodes, honest peers, dynamically based on the quality of the provided files. This makes HonestPeer more robust to the increase in the number of files and nodes in the system. Through simulation, it has been shown that HonestPeer has successfully maintained higher success rate and lower percentage of inauthentic downloads when compared to the original algorithm.

## 1. Introduction

The last few years have witnessed an escalating popularity of Peer-to-Peer (P2P) paradigm, due to its broad applications in large-scale distributed systems and Internet computing (Al-Muhtadi, 2007) as well as file sharing and social networks (Mekouar et al., 2006). In a P2P system, peers share data resources (such as content files) of common interest (Su et al., 2012) or computing resources (such as storage, CPU cycles and bandwidth) to complete massive tasks. Unlike traditional distributed systems, peers in a P2P system are strangers to one another. They are anonymous volunteers, which are highly dynamic with intermittent availability joining and leaving unexpectedly at all times. These special characteristics pose a grand challenge for building a trustworthy P2P environment, where a reliable reputation system is employed to distinguish between reputable honest peers and malicious, dishonest or selfish peers.

Recognizing and isolating malicious peers is significant in all P2P environments, otherwise peers will not have the initiative to share their resources and will hesitate to send requests

E-mail address: hkurdi@ksu.edu.sa

Peer review under responsibility of King Saud University.

ELSEVIER **Production and hosting by Elsevier**

http://dx.doi.org/10.1016/j.jksuci.2014.10.002
1319-1578 © 2015 The Author. Production and hosting by Elsevier B.V. on behalf of King Saud University.

to other peers in fear of receiving corrupted or inauthentic files or being exposed to malware. These concerns are even more critical in e-commerce applications (such as eBay) and content delivery networks (such as Gnutella, 2013), where different kinds of attacks were observed during the last few years. For instance, the Gnutella worm infected thousands of files in the last millennium.

Therefore, many trust and reputation management systems, such as EigenTrust (Kamvar et al., 2003), PeerTrust (Xiong and Liu, 2004) and PowerTrust (Zhou and Hwang, 2007), have been proposed to prevent such attacks on P2P systems (Hwang et al., 2012). The EigenTrust reputation system (Kamvar et al., 2003) is among the most known and used reputation systems (Lin et al., 2013). It has been the subject of frequent enhancements and several variants. This is due to its special characteristics, which include scalability and efficiency as an implementation-ready trust scheme that has been designed particularly for P2P systems (Chiluka et al., 2012; Shen et al., 2010; Nishikawa and Fujita, 2010; Abrams et al., 2005).

EigenTrust reputation system (Kamvar et al., 2003) aims at reducing downloads of inauthentic files through identifying sources of these files and isolating malicious peers distributing them, as well as advising other peers not to download from them. This is done by giving each peer local trust and global reputation values based on their previous behavior in the system. Additionally, a group of peers are designated as trustworthy peers. They are called pre-trusted peers and play a critical role in calculating the reputations of other peers. Although the idea of pre-trusted peers has efficiently reduced spreading inauthentic files, it causes peers to center around pre-trusted peers. As a consequence, other peers would be ranked low despite potentially being honest (Chiluka et al., 2012), marginalizing their role in deciding reputations of other peers and limiting their chances of being chosen as download sources even though they have files of better quality than pre-trusted peers. Furthermore, if a pre-trusted peer downloads an inauthentic file from a malicious peer, this would allow the file to be easily accepted by other peers, leading to a chain of inauthentic downloads.

To tackle this problem, this paper proposed enhancing the EigenTrust algorithm by giving peers with high reputation values a role in calculating the reputation of other peers. Through simulation, it has been shown that the proposed algorithm, HonestPeer, has successfully maintained a higher success rate and lower percentage of inauthentic downloads, by good peers, when compared to the EigenTrust algorithm. Among the main contributions of this paper are:

- An extensive review of variants of the well established reputation management system, EigenTrust.
- An enhancement to EigenTrust, HonestPeer Algorithm, that increases the success rate and decreases the percentage of inauthentic downloads, when compared to EigenTrust.
- A well controlled experimental framework to evaluate reputation management systems.

The remainder of the paper is organized as follows: Section 2 reviews related reputation management systems in general with more focus on the EigenTrust *t* algorithm and its variants. In Section 3, our proposed enhanced EigenTrust algorithm, HonestPeer, is introduced. The evaluation process and results discussion are presented in Sections 4 and 5 respectively. Finally, a conclusion with a summary and future work is provided in Section 6.

## 2. Related work

Reputation management systems are used to build trust in P2P systems by monitoring peer behavior in the system and allowing them to evaluate their transactio ns. In this section, we review the state-of-the-art in reputation management systems focusing on the EigenTrust algorithm and its variants.

### *2.1. Reputation management systems*

A reputation management system assigns a reputation value to each peer. In this way, peers get a full picture of each other. These reputation values help the system fight malicious, dishonest and selfish peers. Reputation systems vary in their techniques for computing the reputation values utilizing them. Since the early days of P2P systems, many reputation systems have emerged. This Section provides a brief review of some related work, with more details available in Zhou and Hwang (2007) and Hwang et al. (2012).

In (Aberer and Despotovic, 2001), peer reputation is based on the peer's past actions in the system. It is calculated dynamically according to the peers history, opinions of other peers who interacted with them and the probability that they might cheat. In this way, the peer can judge the reputation of other peers in the system, even those with whom he has not interacted.

The PeerTrust reputation management system proposed in Xiong and Liu (2004), computes the peer reputation as the average trust values weighted by the score of peers providing these values, the number of transactions in which they were involved and the credibility of their feedback, among other factors. The main drawback of this approach is the heavy overhead associated with retrieving weighting factors (Hwang et al., 2012).

In (Zhou and Hwang, 2007), the PowerTrust reputation system developed dynamically selects a small number of nodes that are most reputable, using a distributed ranking mechanism. These nodes are called Power Nodes and play an important role in calculating the global and local reputation values of other peers. By using a look-ahead random walk strategy and leveraging the power nodes, PowerTrust is known for its ability to improve the global reputation accuracy and the trust values aggregation speed. On the other hand, it takes time until power nodes are selected and identified.

In (Hu et al., 2012), a reputation system that introduced a probabilistic distributed trust model is presented. This model uses probability to assess the Quality-of-Service (QoS) provided by different peers and suggests a security protection mechanism. Besides being computationally expensive, this work relies only on successful transaction rates as a performance measure, thereby considering the successful transactions of both good and malicious peers equally. This measure does not give an indication of how good peers are protected and serviced better than malicious peers are.

A different approach that focuses on the reputation of the file being transmitted rather than the sending peer is proposed in Walsh and Sirer (2005); the reputation value is assigned to the file instead of the peer in other models. The approach provides a simple weighted voting protocol in which any peer can evaluate a file positively or negatively, and then votes are collected and aggregated for final evaluation.

### 2.2. EigenTrust algorithm

EigenTrust Algorithm (Kamvar et al., 2003) was proposed by professor Kmvar et al. from the Stanford University in 2003. The algorithm has been incrementally developed with additional features in each increment. The basic EigenTrust algorithm has a simple centralized reputation calculation strategy, while the advances include distributed, transitive and secured strategies for global calculations. For simplicity, the following explains the main idea of the distributed strategy of the algorithm, and more details are available in Kamvar et al. (2003).

Consider a P2P system consisting of $n$ peers. Each time peer $i$ downloads a file from peer $j$, it rates the transaction as *sat* $(i,j)$ if positive, and *unsat* $(i,j)$ if negative, and keeps a record for the number of each. Then the trust value $s_{ij}$ is defined as:

$$s_{ij} = sat(i,j) - unsat(i,j) \tag{1}$$

To insure that all trust values are between 0 and 1, the trust value $s_{ij}$ is normalized as follows:

$$c_{ij} = \frac{\max(s_{ij}, 0)}{\sum_j \max(s_{ij}, 0)} \tag{2}$$

Usually, there are some peers that are known to be trustworthy in any P2P system, so they are identified at an early stage of the system life as a set of pre-trusted peers, $P$. This is especially important for inactive peers or those who recently joined the system, as they do not trust any peer. Thus, the trust value is redefined as:

$$c_{ij} = \begin{cases} \frac{\max(s_{ij},0)}{\sum_j \max(s_{ij},0)} & |\text{if } \sum_j \max(s_{ij,0}) \neq 0 \\ p_i, & |\text{otherwise} \end{cases} \tag{3}$$

where

$$p_i = \begin{cases} \frac{1}{|p|}, & |\text{if } i \in P \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

$|P|$ is the number of pre-trusted peers

The value of $c_{ij}$ represents how much peer $i$ trusts peer $j$, based on the past experiences with the set of peers $B_i$, from which peer $i$ has downloaded files. This value is used to calculate the current reputation $t_i^{(k+1)}$ of peer $i$ among the set of peers $A_i$, which have downloaded files from $i$. To calculate, the trust values assigned to the peer $i$ by other peers, weighted by the reputation of the assigning peers are aggregated as follows:

$$t_i^{(k+1)} = \left(c_{1i}t_1^{(k)} + \ldots + c_{ni}t_n^k\right) \tag{5}$$

Sometimes, malicious users in P2P systems form a malicious collective where they assign arbitrary high trust values to each other and arbitrary low trust values to good peers. This issue is addressed by recalculating the current reputation of each peer as follows:

$$t_i^{(k+1)} = (1-a)\left(c_{1i}t_1^{(k)} + \ldots + c_{ni}t_n^k\right) + ap_i \tag{6}$$

where $a$ is a constant $\leqslant 1$

Among the main challenges facing distributed reputation management systems, is aggregating trust values of most peers without congesting the network. Therefore, EigenTrust introduced the notion of transitive trust where when a peer $i$ trusts peer $j \in B_i$ then it also trusts peer $x \in B_j$. This results in less messages complexity and less overall messages.

This algorithm has many well-known advantages, which include simplicity, scalability and efficiency as an implementation-ready trust scheme that has been designed particularly for P2P systems (Chiluka et al., 2012; Shen et al., 2010; Nishikawa and Fujita, 2010; Abrams et al., 2005), as indicated earlier. Its main drawback, however, is the high rank given to pre-trusted peers, which results in several problems. First, a pre-trusted peer may not last forever, due to the dynamic nature of P2P systems; this would reduce reliability of the system (Zhou and Hwang, 2007). Second, other good peers might be ranked low despite potentially being honest and owning more authentic files (Chiluka et al., 2012). Third, a centrality attack will have a system-wide negative impact when a pre-trusted peer downloads an inauthentic file by mistake. Fourth, although the entire algorithm is highly dependent on how the reputation, $t_i^{(k+1)}$, of each peer is calculated, it is difficult to find a suitable value for the proliferation constant, $a$, which is utilized to determine the weight of pre-trusted peers against other peers. Detailed analysis of the EigenTrust algorithm is presented in Chiluka et al. (2012).

### 2.3. EigenTrust variants

EigenTrust algorithm has gained vast attention as a renowned reputation management system for P2P systems. This is revealed as several studies analyzing its performance and suggesting further improvements. For instance, (Abrams et al., 2005) indicated that although EigenTrust can successfully alienate malicious peers and maximize file uploads, it is easy to manipulate, allowing selfish peers to lie about their recommendations. To overcome this shortcoming, it suggested a “non-manipulable” scheme, where peers are partitioned into groups and arranged in an ordering such that each peer only has incentives to query and download from peers in other groups.

The positive opinion network and the Inverse EigenTrust schemes were proposed in Donato et al. (2007), to extend EigenTrust to more kinds of malicious peers. The positive opinion network is a logical network represented as a directed graph of nodes and links between them. It is constructed by inserting a directed link between two peers if an authentic file is downloaded from the source to the destination. Applying EigenTrust to the transpose of the positive opinion network results in score zero for all kinds of malicious peers, so they can be easily detected.

In (Nishikawa and Fujita, 2010), the effectiveness of EigenTrust against natural attacks is stressed. However, it is indicated that the algorithm is less effective in dealing with unreliable peers, who behave honestly in some transactions and evilly in others. This usually results in chains of inauthentic downloads. As a solution, EigenTrust developed a probabilistic approach that takes the unreliable behavior of peers into account and eliminates the need for pre-trusted peers.

The Fuzzy Synthetic Evaluation Model, FSEM-Trust, is suggested in Rahman et al. (2010), where each peer is evaluated individually by several factors which are aggregated into a weighted sum calculated based on fuzzy logic. This weighted sum is regarded as the trust value, which can enhance the ability of identifying malicious peers.

In (Lin et al., 2013), the Personalized EigenTrust reputation system is introduced to enable each peer to choose its trusted peers from a social network. This approach aims at eliminating the need for pre-trusted peers and identifying more kinds of malicious attacks than the original EigenTrust algorithm.

All the above work aimed at minimizing the dependency of EigenTrust on pre-trusted peers. To do this, they proposed new approaches that might have improved the efficiency of the algorithm to some extent, sacrificing the algorithm's simplicity as a price. Therefore, the aim of this paper was to handle the problem of pre-trusted peers, while preserving the simplicity of the original EigenTrust through the concept of the honest peers as described in Section 3.

## 3. HonestPeer approach

HonestPeer was designed as a distributed reputation management algorithm for P2P systems. Its main objective is to better service reputable "good" peers by increasing their success rate and decreasing their inauthentic downloads.

In developing HonestPeer, the five design objectives of reputation systems identified by EigenTrust (Kamvar et al., 2003) were considered. These included: self-policing by defining and enforcing shared ethics by the peers, rather than a central authority; anonymity of all peers; no profit assigned to new comers; minimal computation, storage and message overheads and robustness to malicious collectives of peers. Therefore, we started from the original EigenTrust, following the same approach in calculating the trust value of each peer using Eqs. (1) and (3). Then, for a peer to compute its current reputation value, it needs to assist the reputations received during the last run to find the honest peer, $h$, with the maximum reputation value, $t_h$.

$$t_h^{(k)} = \max\left(t_1^{(k)}, \dots, t_n^{(k)}\right) \tag{7}$$

where $h \in A_i$

The honest peer, $h$, plays a critical role in calculating the value of the proliferation parameter, $a$, in Eq. (6), which is not constant, in contrast to the original EigenTrust algorithm. Consequently, if $h$ is part of the pre-trusted peers, $P$, pre-trusted peers should still have a high effect on deciding the reputations of other peers. Otherwise, if is not part of $P$, then pre-trusted peers effect in deciding the reputation of other peers should be marginalized. Based on that, the current reputation of a peer, $i$, is calculated as:

$$t_i^{(k+1)} \begin{cases} (1-a)\left(c_{1i}t_1^{(k)} + \dots + c_{ni}t_n^{k}\right) + ap_i, & \text{if}|h \in P \\ a\left(c_{1i}t_1^{(k)} + \dots + c_{ni}t_n^{k}\right) + (1-a)p_i, & \text{if } h \notin P \end{cases} \tag{8}$$

where

$$a = \begin{cases} t_h^{(k)}, & |\text{if } t_h^{(k)} > 0.5 \\ 1 - t_h^{(k)}, & |\text{if } t_h^{(k)} \leqslant 0.5 \end{cases}$$

In this way, HonestPeer helps each node to identify its honest peer based on the last transaction. It is important that the honest peer is dynamically replaceable, if he becomes less active or provides inauthentic files.

So when a query for a file is issued, a list of peers having this file is generated. The peer selects a download source based on the reputation metric (7). After downloading the file, the peer evaluates the transaction and updates trust values accordingly. This information is shared with all peers on the friends' list. The process is repeated until the algorithm converges. The complete algorithm is shown in Fig. 1.

## 4. System performance analysis

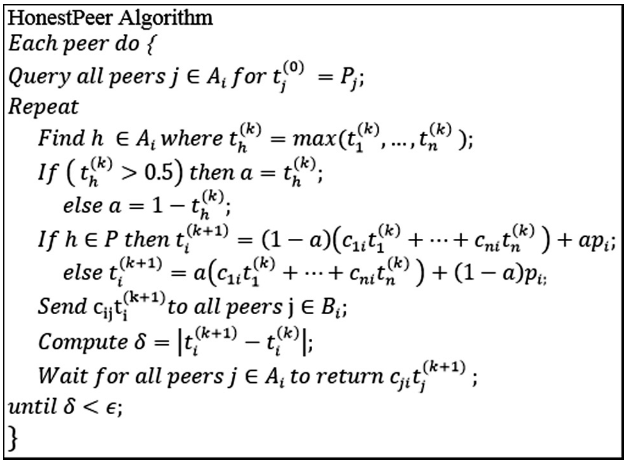
HonestPeer has been evaluated by analyzing its time complexity and studying its behavior based on a strictly controlled empirical framework that has been designed specially for reputation management systems and considered among the main contributions of this paper.

### 4.1. Algorithm complexity and convergence overhead

Critical analysis of the algorithm in Fig. 1 reveals that its complexity is entirely dependent on the computation of $t_i^{(k+1)}$ and the convergence overhead. The latter is measured as the number of iterations before the global reputation convergence. The computation of $t_i^{(k+1)}$ is not intensive, since most $c_{ij}$ have a value of zero. Moreover, $A_i$ and $B_i$ are small, which speeds up the process of searching for them (Kamvar et al., 2003). Hence, the algorithm complexity is bounded and it converges fast.

### 4.2. Experimental framework

The main difficulty faced by this research was finding a suitable evaluation framework for the proposed system. Despite the fact that reputation management issues are gaining considerable attention with a lot of recently proposed systems and models, surprisingly, systematic approaches and tools for evaluating them are drastically lacking. Instead, each proposed work designs an evaluation framework that best shows the



```
HonestPeer Algorithm
Each peer do {
Query all peers $j \in A_i$ for $t_j^{(0)} = P_j$;
Repeat
    Find $h \in A_i$ where $t_h^{(k)} = max(t_1^{(k)}, \dots, t_n^{(k)})$;
    If $(t_h^{(k)} > 0.5)$ then $a = t_h^{(k)}$;
        else $a = 1 - t_h^{(k)}$;
    If $h \in P$ then $t_i^{(k+1)} = (1-a)(c_{1i}t_1^{(k)} + \dots + c_{ni}t_n^{(k)}) + ap_i$;
        else $t_i^{(k+1)} = a(c_{1i}t_1^{(k)} + \dots + c_{ni}t_n^{(k)}) + (1-a)p_i$;
    Send $c_{ij}t_i^{(k+1)}$ to all peers $j \in B_i$;
    Compute $\delta = |t_i^{(k+1)} - t_i^{(k)}|$;
    Wait for all peers $j \in A_i$ to return $c_{ji}t_j^{(k+1)}$ ;
until $\delta < \epsilon$;
}
```

**Figure 1** HonestPeer algorithm.

efficiency of its reputation strategy compared to previous work. This means that no common performance measures or control variables are defined. There is a need for strategy-agnostic performance measures and control variables, in order to conduct systematic evaluation and comparisons between reputation management systems.

Therefore, a strictly controlled evaluation framework in a simulated P2P network model has been designed. It can be considered among the main contributions of this paper. The control variables and performance measures were selected from the application field, which is the file-sharing P2P network, to ensure that selection is not favoring a certain strategy. However, there are always tradeoffs. In this case, it can be difficult to explain the direct relationship between the obtained results and the strategy utilized which is a common issue among all heuristics from different fields (Bartholdi and Loren, 1988).

The end objective was to answer the question: does giving peers with high reputation values a role in calculating reputations of other peers have positive impacts on the success rate and the amount of inauthentic downloads by good peers?

To answer this question, we employed an open-source simulator, RM-SIM, developed in West et al. (2009). The simulator imitates a variety of network configurations and multiple malicious peer behavioral models. It comes in two versions: java and C. This paper was based on the java implementation for better portability. All experiments were carried on an i5 Intel core 2.40 GHz laptop, with a 6 GB RAM, running Windows 7 Home Premium of 64-bit. Software tools included NetBeans IDE NetBeans IDE, Visual C++ 2010 Express, and jGRASP version 1.8.8_23.

Two main P2P system issues were considered:

- Scalability to a larger number of nodes, which is important for P2P systems as indicated in Aberer and Despotovic (2001).
- Sustainability under various loads in terms of number of files and transactions between peers (Shen et al., 2010).

The evaluation framework involved the following main steps:

1. Determining characteristic design elements of P2P systems and deciding on the set to be considered: number of peers, number of transactions and number of distinct files.
2. Varying the experimental variables, number of peers and number of files to simulate a representative sample of P2P environments. Two sets of experiments were designed, as shown in Table 1: in the first set, the number of files was constant at 1000 files, while values for the number of peers were selected in the range of 200, 500, 1000, 1500 peers. In the second set, the number of peers was considered constant at 1000 peers while values for the number of files were selected in the range of 200, 500, 1000, 1500 files.
3. Stabilizing the number of transactions at the value of 5000 transactions and the percentage of pre-trusted peers at 10%, good peers (including pre-trusted) at 75% and malicious peers at 25%.

**Table 1** Experimental sittings.

| | Experiment | No. of users | No. of files | No. of pre-trusted peers |
|---|---|---|---|---|
| Set 1: variant No. of peers | Exp1.1 | 200 | 1000 | 20 |
| | Exp1.2 | 500 | 1000 | 50 |
| | Exp1.3 | 1000 | 1000 | 100 |
| | Exp1.4 | 1500 | 1000 | 150 |
| Set 2: Variant No. of files | Exp2.1 | 1000 | 200 | 100 |
| | Exp2.2 | 1000 | 500 | 100 |
| | Exp2.3 | 1000 | 1000 | 100 |
| | Exp2.4 | 1000 | 1500 | 100 |
| | Exp2.5 | 1000 | 2000 | 100 |

4. Adopting suitable models for application network and peers:
   - Application Model: similar to Aberer and Despotovic (2001) and Wang and Vassileva (2003), a P2P file-sharing application was considered to demonstrate the proposed reputation management system. However, the system is general and can be easily applied to other P2P applications, such as online auctions, e-commerce or P2P distributed computing. In query generation, the intelligent query model was considered, where a user may not request a file that he already possesses or has requested in the past and the requested file must exist in the network. Each peer had an equal opportunity of being a file requester. The decision of which file is requested is indicated by the Zipf distribution Schlosser and Kamvar, 2003. For simplicity, each user had an initial library of identical expected size.
   - Network model: as in West et al. (2009), this paper considered infinite bandwidth and a “closed world” where users within a network are static; they do not join and leave the network.
   - Peer models: four models of peers were studied: good peers who provide honest feedback and clean-up invalid files from their library; purely malicious who lie about feedback and retain invalid files; malicious collective which is a group of cooperating malicious users providing a positive feedback about peers within their group and a negative feedback for other peers; unreliable peers with inconsistent behaviour imitating good peers in some transactions and malicious peers in some anothers.

5. Identifying benchmarks: two cases were considered which comes in line with the methodology utilized in Zhou and Hwang (April 2007) and West et al., 2009:
   - EigenTrust algorithm: as describes in Section 2.2.
   - None: the case of the absence of any reputation management system where each peer chooses randomly the service provider from where to download files (denoted by none). This system is employed only to provide a baseline for comparison purposes. It represents the worst case in all scenarios.

6. Determining suitable performance measures: two performance measures were considered:

- Percentage of inauthentic file download by good peers: if the computed global reputation value accurately reflects peer behavior, then the number of inauthentic files downloaded by good peers should be minimized (Kamvar et al., 2003).
- Success rate of good peers: based on the algorithm advice good peers should achieve a higher success rate (West et al., 2009) which is defined as:

$$\text{Success Rate of good peers} = \frac{\#\text{valid files received by good peers}}{\#\text{transactions attempted by good peers}} \tag{9}$$

7. Comparing the performance of the three reputation management systems and analyzing the main findings.

Increasing the accuracy of this experimental study by repeating each experiment ten times (Zhou and Hwang, 2007) and calculating the mean outcome, measuring the uncertainty in data using the measure of standard error (SE) and displaying the values as error bars in all charts.

## 5. Results and discussion

The summary results of running 5000 transactions in both sets of experiments are presented in Figs. 2–5. The error bars in each figure represent the standard error of the mean.

### 5.1. Success rate

Figs. 2 and 3 illustrate the success rate for good peers calculated as the number of valid files received by good peers divided by the number of transactions attempted by good peers, based on each reputation system. When comparing Figs. 2 and 3, the graphs show results that are similar in some ways and different in others.

In Fig. 2 the success rate of good peers is plotted against the number of peers in the system. It shows that when the number of files is fixed, HonestPeer success rate increases steadily as the number of peers increases. Although the same is true for EigenTrust, the increase in success rate is lower than that of HonestPeer. Not to mention that the gap in success rate between the two trust systems is directly proportional to the number of peers in the system. This shows the scalability of HonestPeer to a large number of peers.

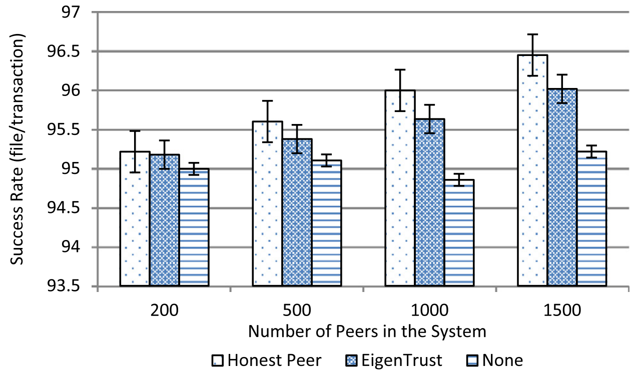


**Figure 2** Success rate of good peers when different number of peers are considered.

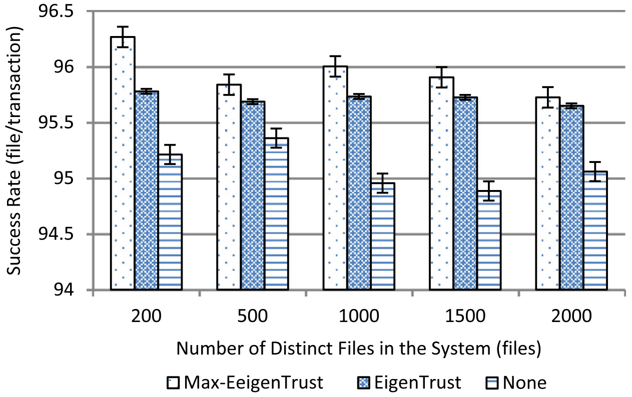


**Figure 3** Success rate of good peers when different number of distinct files are considered.

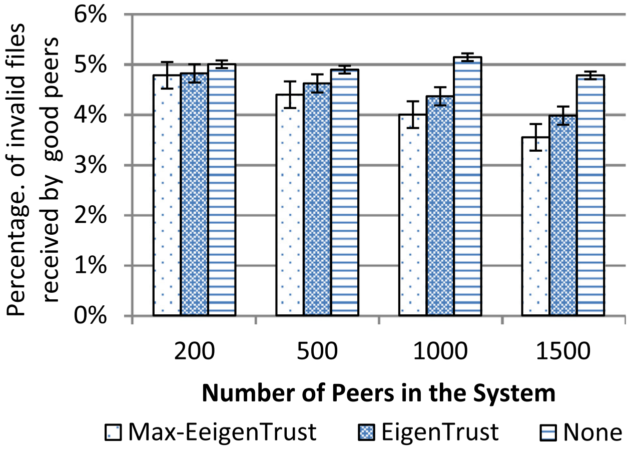


**Figure 4** Percentage of invalid files received by good peers when different number of peers is considered.

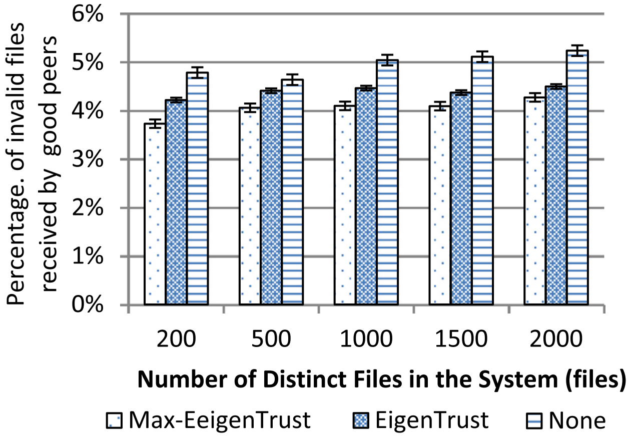


**Figure 5** Percentage of invalid files received by good peers when different number of distinct files is considered.

Fig. 3 plots the success rate of good peers against the number of distinct files in the system. Despite showing a slight decrease in success rate as the number of files in the system rises, HonestPeer still outperforms EigenTrust in all scenarios.

These graphs clearly depict HonestPeer surpasses EigenTrust in effectiveness and capability to help good peers to download valid safe files. This can be attributed to the ability of HonestPeer to dynamically choose the honest peers after each round, while the pre-trusted peers are statically chosen in EigenTrust irrespective to their performance.

As expected, both figures confirm that when using no trust system, there is a fluctuation in the results because the download process becomes random.

### *5.2. Percentage of invalid downloads*

The percentage of invalid files received by good peers, based on each reputation system is evaluated in Figs. 4 and 5.

In Fig. 2 the success rate of good peers is plotted against the number of peers in the system.

Fig. 4 illustrates the relationship between the percentage of invalid files received by good peers and the number of peers in the system. It shows that HonestTrust outshines EigenTrust also in regard to the percentage of invalid files received by good peers. When the number of peers is between 200 and 500, the difference is by a small margin. However, the gap increases dramatically when there are 1000–1500 peers in the system. This finding proves HonestTrust to be especially useful when the number of peers is particularly high. Overall, the percentage of invalid files downloaded by good peers drops by more than 1.5%.

Fig. 5 follows a similar pattern, as it showcases HonestPeer success in decreasing the percentage of invalid files compared to EigenTrust. Although there is a marginal increase in the percentage, HonestPeer manages to maintain its success despite such a large number of files. This is due to the fact that EigenTrust depends on a static group of pre-trusted peers, while HonestPeer released this restriction by selecting the most reputable nodes, honest peers, dynamically based on the quality of the provided file making HonestPeer more robust to the increase in the number of files.

## 6. Conclusion

Peer-to-Peer systems offer many advantages for free sharing of resources between nodes. However, they have associated risks where malicious peers spread inauthentic files that might disrupt the entire system. Therefore, reputation management systems are emerging to overcome this problem. Among the main paradigms in this area are the EigenTrust reputation management systems. It provides mechanisms to effectively reduce invalid download of inauthentic files by good peers. On the other hand, a well known drawback is its reliance on the concept of pre-trusted peers so other peers would be ranked low despite potentially being honest marginalizing their role in the system.

To tackle this problem, this paper, proposed enhancing EigenTrust by giving peers with the higher reputation values an important role in calculating the reputation of other peers. Through simulation, it has been shown that this approach has positive impact in reducing the percentage of invalid files and increasing the success rate of good files downloaded by good users.

These positive results have been maintained as the number of peers in the system increased gradually from 200 to 1500 suggesting better scalability of HonestPeer when compared to EigenTrust. Furthermore, when increasing the number of distinct files in the system gradually from 200 to 2000 the effectiveness of HonestPeer has also been maintained showing better sustainability over EigenTrust in all scenarios.

In our future work, we intend to study the impact of combining the file trust value with peer trust value in having a more robust reputation management system. We also planning to extend HonestPeer to handle more threat models such as malware and poisoning.

## Acknowledgments

This work was partially funded by the Long-Term Comprehensive National Plan for Science, Technology and Innovation of the Kingdom of Saudi Arabia, grant number 11-INF1895-08.

## References

Aberer, K., Despotovic, Z., 2001. Managing trust in a Peer-2-Peer information system. In: Proc. of the 10th Int'l Conf. Information and Knowledge Management.

Abrams, Z., McGrew, R., Plotkin, S., 2005. A non-manipulable trust system based on EigenTrust. ACM SIGecom Exchanges 5, 21–30.

Al-Muhtadi, Jalal., 2007. An efficient overlay infrastructure for privacy-preserving communication on the internet. J. King Saud Univ. Comp. Inf. Sci. 19, 39–59.

Bartholdi, J.J., Loren, P.K., 1988. Heuristics based space filling curves for combinatorial problems in euclidean space. Manage. Sci. 34 (3), 291–305.

Chiluka, N., Andrade, N., Dimitra, G., 2012. Personalizing EigenTrust in the face of communities and centrality attack. In: Proc. of the 26th IEEE International Conference on Advanced Information Networking and Applications, pp. 503–510.

Donato, D., Castillo, C., Paniccia, M., Cortese, G., Selis, M., Leonardi, S., 2007. New metrics for reputation management in P2P networks. In: Proc. of the 3rd International Workshop on Adversarial Information Retrieval on the Web, Banff, Canada, May 8, pp. 65–72.

Gnutella, The Gnutella Protocol Specifications v0.4. Document Revision 1.2. [online] Available <http://cryptnet.net/fsp/cpcd/gnutella_protocol_0.4.pdf> (accessed April 4, 2013).

Hu, J., Li, X., Zhou, B., Li, Y., 2012. SECTrust: a secure and effective distributed P2P trust model. In: Proc. of the Third International Symposium on Intelligent Information Technology and Security Informatics (IITSI), pp. 34–38.

Hwang, K., Fox, G.C., Dongarra, J.J., 2012. Distributed and Cloud Computing from Parallel Processing to the Internet of Things. Morgan Kaufmann.

Kamvar, S.D., Schlosser, M.T., Garcia-Molina, H., 2003. The EigenTrust Algorithm for Reputation Management in P2P Networks. In: Proc. of the 12th International Conference on World Wide.

Lin, Y., Yang, H., Yang, C., Lin, W., 2013. A traceable and fair transaction mechanism for digital rights management on P2P networks. J. Internet Technol. 14 (7), 1043–1052.

Mekouar, L., Iraqi, Y., Boutaba, R., 2006. Peer-to-Peer's Most Wanted: Malicious Peers. Computer Netw. 50 (4), 545–562.

Nishikawa, T., Fujita, S., 2010. An effective risk avoidance scheme for the EigenTrust reputation management system. In Proc. of the First International Conference on Networking and Computing, pp. 36–43.

Rahman, R., Hales, D., Vinko, T., Pouwelse, J.A., Sips, H.J., 2010. No more crash or crunch: Sustainable credit dynamics in a p2p community. In: Proc. of the 2010 Int. Conf. on High Performance Computing & Simulation (HPCS), pp. 332–340.

Schlosser, M.T., Kamvar, S.D., 2003. Simulating P2P Networks. Technical report, Stanford University.

Shen, R., Yong, W., Xiao-ling, T., 2010. The comprehensive trust model in P2P based on improved EigenTrust algorithm. In: Proc. of the International Conference on Measuring Technology and Mechatronics Automation, pp. 822–825.

Su, Z., Li, M., Guo, C., Ma, J., Park, J., Jeong, Y., 2012. Fuzzy set theory-based trust models in multi-agent environment. J. Internet Technol. 13 (1), 159–172.

Walsh, K., Sirer, E.G., 2005. Fighting PeertoPeer SPAM and Decoys with Object Reputation. In: Proc. of SIGCOMM'05 Workshops, August 22–26.

Wang, Y., Vassileva, J., 2003. Trust and reputation model in peer-to-peer networks. In: Proc. Of the Third International Conference on Peer-to-Peer Computing (P2P'03), Sept. 2003.

West, A.G., Kannan, S., Lee, I., Sokolsky, O., 2009. An evaluation framework for reputation management systems. In: Trust Modeling and Management in Digital Environments: From Social Concept to System Development: IGI Global, p. 27 (ch. 12).

Xiong, L., Liu, L., 2004. PeerTrust: supporting reputation-based trust for peer-to-peer electronic communities. IEEE Trans. Knowl. Data Eng. 16 (7), 843–857.

Zhou, R., Hwang, K., 2007. PowerTrust: a robust and scalable reputation system for trusted peer-to-peer computing. IEEE Trans. Parallel and Distrib. Syst. 18 (4), 460–473.