Computing Conference 2017
18-20 July 2017 | London, UK

An Enhanced Bayesian-based Reputation System for P2P File Sharing

Nourah Fahad Janbi
Faculty of Computing and Information Technology-Khulis
University of Jeddah
Jeddah, Saudi Arabia
nfjanbi@uj.edu.sa

Milena Radenkovic
School of Computer Science
The University of Nottingham
Nottingham, United Kingdom
milena.radenkovic@nottingham.ac.uk

***Abstract*—Peer-to-peer (P2P) file sharing networks are widely used by Internet users but due to their anonymity and self-organizational nature, they are often targeted by malicious users. Therefore, reputation systems are used to distinguish between good and malicious peers. Many reputation systems propose the use of the Bayesian model and it has proved to be effective for reputation calculation in a small, fully distributed system. This paper focuses on the distributed rank-based reputation system (DRank), one of the latest systems applying the Bayesian model in P2P streaming. DRank developers evaluated it on a small network with a maximum 200 nodes. In this paper, DRank is examined in a P2P file sharing system and on larger networks. This examination showed a drop in reputation propagation and a rise in the number of polluted files downloaded. Therefore, we propose a number of optimizations to DRank, such as unstructured reputation propagation, and two mechanisms of peer selection: (1) the first peer in the rank, and (2) randomly from a set of top ranked peers. For evaluation purposes, a simulator is built using a P2P simulator and DRank is examined with and without our proposed improvements. The experiments showed that our proposals improve DRank performance in preventing malicious peers from obtaining high reputation values.**

Keywords—Peer-to-Peer; P2P network; Reputation system; P2P file sharing

I. INTRODUCTION

Peer-to-peer (P2P) is a network architecture in which each peer is an equally privileged node that works as both the client and server. This architecture improves resource usage and reduces the cost of running and maintaining a central server in normal client/server architecture. However, despite the scalability of P2P and its efficiency in handling huge numbers of requests, it is vulnerable to attacks and unauthenticated data due to the lack of centralized administration [1]. P2P is used in many application domains such as file sharing, voice over IP, media streaming, and distributed computing. P2P file sharing applications have been targeted by many malicious users and even companies trying to protect their copy-righted files [2] [3] [4]. Peers participating in a P2P network cannot distinguish between malicious and good peers, which leads them to deal with untrusted peers. Therefore, a trust mechanism is required, whereby each peer has a reputation value calculated depending on his behavior on the network. A variety of studies have been conducted to analyze and propose reputation and trust systems (such as [3], [5], [6], [7], [8], and [9]).

These reputation systems can be divided into two main categories: centralized and distributed systems. In centralized systems (such as [10] and [11]) there is a central server (or servers) where the reputation values are stored. Although these systems are easy to design, use, and manage, they have a single point of failure and attack [12]. In contrast, in distributed systems (such as [3], [5], [6], [7], [8], and [9]), peers calculate their neighbors' reputation value and these values might be combined, as a global trust, or not. Despite the strength and effectiveness of reputation systems, they are still vulnerable to attacks [13] [14] such as pollution, whitewashing, Sybil, and collusion. In the pollution attack, malicious peers spread polluted files to reduce availability of the original file or degrade system performance. In whitewashing attacks, malicious peers after misbehaving get a new clean identity. In the Sybil attack, malicious peers use multiple identities and group them to look like a distinct group [3] or use them to vote one identity to a high reputation. Finally, in the collusion attack, malicious peers cooperate to obtain high reputation values by assigning high values to each other and/or low values to other peers [14].

The Bayesian estimation model was first introduced to reputation systems by [15]; it is used to estimate the trustworthiness of a node. Other studies followed that proposal (such as [9], [16], [17], and [18]); in most of them, results were based on small networks. It is unknown how such systems will perform when the network grows, especially fully distributed systems. DRank is one of the latest systems applying the Bayesian model system and it is designed for P2P media streaming [9]. DRank was also evaluated on a small network (200 nodes), and the results showed that DRank can fight network pollution robustly in P2P streaming. In this paper, DRank is examined in a general file sharing system and on a larger network. The concern is that when the network size increases, the reputation propagates more slowly as DRank broadcasts reputations only to direct neighbors. This leads to unrecognized malicious peers and the sharing of more polluted files. Therefore, in this paper the reputation propagation and peer selection mechanisms are enhanced to perform better in a scalable network.

Although sending reputation values to direct neighbors reduces the overhead of broadcasting to the whole network, it limits the spread of the reputation, especially if the network does not change frequently. Therefore, we propose an unstructured propagation mechanism wherein peers send the

978-1-5090-5443-5/17/$31.00 ©2017 IEEE

stored reputation values to a specified number of peers chosen in an unstructured way. In this way, the overhead is not increased, as we are still sending to the same number of peers, and the reputation is sent to a variety of peers each time. Moreover, in most reviewed systems (such as [6], [8], and [9]), the peer with the best reputation is selected, which increases the load on that peer. One way to reduce this load is by using local experience to personalize the selection [5]; this method is considered in the DRank reputation calculation. In this paper, a new method will be examined, wherein a random peer will be selected from a group of best peers. We expect that by introducing randomness in the selection, the load on the peer with the best reputation will be lower. Both proposals are investigated against the original mechanisms to evaluate their effectiveness in preventing malicious peers from achieving high reputation values and distributing polluted files.

The paper is organized as follows: first, a review of related work is presented (Section 2). After that, the methodology is described (Section 3), followed by the protocol implementation (Section 4). The experiment design is presented in Section 5, and finally, Section 6 discusses our results and Section 7 concludes and outlines future work.

II. Related Works

A variety of studies have been conducted to analyze and propose reputation and trust models. Basically, these can be divided into two main categories: centralized and distributed (or decentralized) systems. In centralized systems, there is a central server (or servers) where the reputation values are stored, such as eBay [10] or Amazon [11]. These values can be calculated in the central servers or locally and then sent to the server. Although these systems are easy to design, use, and manage, they have a single point of failure and attack [12]. In contrast, in distributed systems (such as [3], [5], [6], [7], [8], and [9]), peers calculate their neighbors' reputation value and these values might be combined, as a global trust, or not. Global reputation values are calculated in reputation managers and distributed using distributed hash tables (DHT). In local reputation systems, the values are kept locally and not aggregated as global ones are. Therefore, the peer is protected from reputation system attacks, as the values are from his own experience. However, large and dynamic systems require a long time to determine a peer's behavior [19].

Two of the early reputation systems are EigenTrust [5] and PeerTrust [6]. Both of these calculate global trust values for all peers in the network. These values are combined and stored in trust managers and distributed using DHT such as Tapestry [20] or Chord [21]. PeerTrust differs from EigenTrust in the number of factors that are included in the calculation. PeerTrust uses five factors: satisfaction, number of transactions, credibility of feedback, transaction context, and community context. In contrast, EigenTrust uses only satisfaction and credibility of the feedback (the trustworthiness of the peer giving the feedback). Although adding more factors may improve the accuracy of the system, they add a great deal of overhead in terms of sending and storing them. Moreover, EigenTrust does not consider how long the history is kept, which is also an issue that might be affected by malicious attacks.

Credence [7] is another decentralized reputation system that focuses on the object's (file) reputation. It includes factors similar to EigenTrust, vote and trust, although they are assigned to the objects. Furthermore, votes in Credence are explicit and represented by a statement. This is one of the biggest differences in its design, as most systems use binary voting (0 satisfied, 1 unsatisfied) while Credence uses statements that have a common syntax. Although this voting is simplified by the provided user interface, from our point of view it is still complex for normal users. These votes are stored in the client's machine on a local database and to calculate the reputation value of an object, votes are collected using a pull-based technique, and then they are aggregated to obtain the total vote value of the object.

Sorcery [3] introduced social networks to reputation systems. Each peer stores a list of his friends who will share reputation values with them. This list is kept securely in the client's machine and used to compare the overlapped voting of friends and strangers. One of the main benefits of the social network is that it addresses the cold start problem. Cold start refers to a newcomer joining the network and being deceived because of lack of experiences [3]. In our opinion, although the social network is a great feature that makes an important contribution to the system, it should be combined with a strong reputation system. This is because P2P file sharing networks are huge and there is a higher probability of communicating with a stranger if we compare the friend list size to the network size even though the friends' experiences are also included. SocialTrust [8] is a recent social network reputation system that uses the concept of a real social relationship. The relationship is built using the frequency of communication between peers and it is bi-directional, meaning each person will be added to the other's friends list. Peers can also delete peers from their friendship list after a bad experience. SocialTrust combines both the reputation system and social degree to improve reputation evaluation. Although SocialTrust overcomes some of the drawbacks of Sorcery, it still does not take into consideration that P2P file sharing networks are usually large and frequent communication between peers is sometimes rare.

DRank [9] on the other hand, as previously mentioned, is a distributed rank-based reputation rating system designed for P2P media streaming that is based on the Bayesian model from [15]. DRank adapts a balanced solution between using either global or local reputations, where peers calculate a local reputation value using their experience (first-hand information) combined with their direct neighbors' experience (second-hand information) of the same peer. Second-hand information is not used directly; it is first examined with the current local reputation, which may results in discarding it. DRank considers two dynamic factors in the calculation, satisfaction and trust, and five constant factors: (1) how long history should be kept, (2) the weight by which reputations from neighbors will affect the local reputation, (3) threshold value to define the acceptable difference between the first- and second-hand information, (4) threshold value to define the acceptable trust level, and (5) threshold value to define the acceptable reputation level. All these constants were examined closely in [19] to determine their affect and their most suitable values. The greatest benefit

978-1-5090-5443-5/17/$31.00 ©2017 IEEE

DRank offers over the previously discussed systems is that it does not require much calculation to defend against different attack models and it uses a very simple basic calculation.

III. Methodology

There are three main techniques used to study reputation-based P2P systems: (1) collecting data from an existing system and then applying the new algorithm to that data or just analyzing it, (2) implementing the system, launching it, and then monitoring it, and (3) simulating the system and analyzing the simulation results. Although (1) and (2) might be more accurate, they require a long time to collect a sufficient amount of data. Therefore, simulation is used in this paper.

In order to choose a simulator that suits the purpose of our study, we evaluated three simulators QueryCycle [22], NS-2 [23], and PeerSim [24]. PeerSim (peer-to-peer simulator) is chosen as it is specifically designed for P2P networks and provides high scalability by abstracting the network layer details. Moreover, PeerSim implements both a cycle-driven engine and an event-driven engine and they can be used synchronously or separately, while QueryCycle only has a cycle-driven engine and NS-2 has an event-driven engine.

Fig. 1 shows the main layers of the reputation system we are building, which includes a P2P file sharing network, a reputation and trust system, and file sharing. The DRank model was adopted to work on a general P2P file sharing network, as it had been designed for media streaming. This includes reputation calculation and sharing and simple file searching. File searching in the P2P network is a separate area of research not considered in our study. Thus, the search was simulated in an abstract way. In addition, in the search procedure, message routing is also beyond the focus of our study. Therefore, it is assumed that the search and routing processes are optimized. The nodes' connection and organization are static, which is initialized from the beginning of the experiment and does not change throughout it. Nodes are either good or malicious and their behavior differs depending on that. Malicious peers always accept files and send them as polluted. For reputation and trust calculation and sharing, the exact specification in [9] was used. Moreover, the reputation propagation is implemented in both the DRank specification and the suggested unstructured propagation. Finally, two forms of peer selection were implemented: (1) choose the peer with the highest value, or (2) randomly choose one peer out of the peers with the highest reputations.

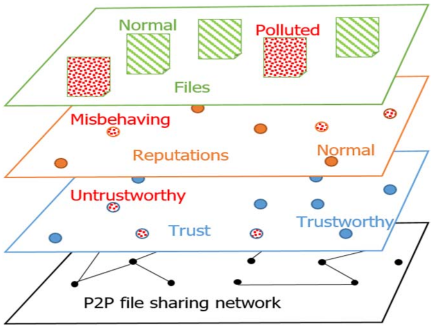

Fig. 1. Layers included in our system

In the experiments, DRank was compared with a simple P2P file sharing system that has no reputation system. In addition, it was examined in the presence of a collusion attack.

In the collusion attack scenario, malicious peers always give a good reputation to other malicious peers and half of the time they give the good peers a negative reputation.

IV. Protocol

The built simulator is divided into two main parts: (1) the P2P file sharing protocol, and (2) the reputation propagation protocol. The first part includes the file sharing process, local reputation update after each transaction, and peer selection, while the second part includes the reputation values propagation (or broadcast). Fig. 2 summarizes the reputation exchange process in the reputation system. The first step occurs when peer i receives a file from peer j and then the reputation object of j is created or updated in peer i. Peer k also had a previous experience with peer j and has a reputation object of j. In the second step, peer k broadcasts (or sends) the reputation of j to peer i. In the final step, peer i will have the reputation of j as a combination of his experience and peer k's experience. Moreover, a reputation object of k will be created or updated to represent the experience that i had with k.

A. P2P File Sharing and Peer Selection

In the file sharing process, the file searching process is abstracted as mentioned in the Methodology, but our aim here is to examine the method of peer selection. After finding the peers with the desired file, three scenarios were implemented: (1) choosing a node randomly without considering the reputation values, (2) choosing the node with the best reputation value, and (3) choosing randomly from a set of nodes with the best reputation values. Then a request message is sent to the selected peer.

There are two primary types of messages to be handled by peers: (1) request file messages (ask) and (2) replay messages (replay), which contain the file. For request messages (1), malicious peers send polluted files and good peers send normal files. In contrast, for replay messages (2), when the file is received the good peers add normal files to their files list and ignore the polluted files. However, malicious peers always add the files to their list.

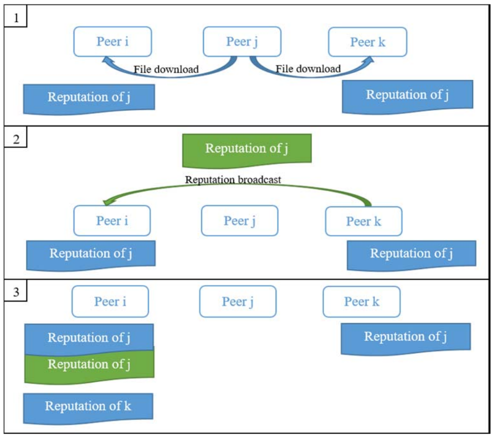

Fig. 2. Steps of reputation exchange

978-1-5090-5443-5/17/$31.00 ©2017 IEEE

After adding the files, the good peers update the reputation depending on the experience. On the other hand, if the receiver is a malicious peer, there are two cases: (1) If the sender is also malicious, then the reputation of the sender is updated as a satisfied experience. This increases the reputation of the sending peer. (2) If the sender is a good peer, the reputation of the sender will be updated, with a 50% probability of either satisfied or not. This probability is used to illustrate that malicious peers do not always give good peers a bad reputation because they are trying to keep their trust high.

B. Peers' Reputation Propagation

Reputations have to be shared with other peers in the network to build a better view of the network. As mentioned earlier, keeping reputation locally on the peer's device reduces the effectiveness of the system. There are two main ways to share reputations: pull-based and push-based. DRank is push-based; it periodically broadcasts the reputations to direct neighbors. To implement the original reputation broadcast, in each cycle each node sends a copy of the reputations it has to its direct neighbors. In our opinion, although broadcasting to direct neighbors reduces the overhead of broadcasting on the network, it limits the spread of the reputation unless the network is frequently changed. This is because in DRank, only first-hand information is shared and this means that peers cannot resend what they have received from their neighbors. Therefore, in this paper we suggest an enhancement to DRank by broadcasting the reputation to a set of randomly selected peers. We believe this will improve the propagation of the reputations and will not greatly affect the performance as they are only sent to a specific number of peers that should be configured early. To implement the unstructured reputation propagation, in each cycle each node sends a copy of the reputation's vector to *N* randomly selected nodes, where *N* is specified in the configuration file. When the reputation vector is delivered to a receiver, each entry in the vector is processed as follows: First, the value of the received reputation is compared with the value that the receiver stores and the trust will be updated depending on that. Second, the trust value of the sender will be compared with the threshold value for the trustworthiness of the peer. If the sender is considered trustworthy, then the sent reputation value will be added to the receiver set of reputations. The full reputation calculation can be found in [19].

V. EXPERIMENT DESIGN

In this section, the process of experiment preparation is explained. The first part shows some of the sensitivity analysis results and the second part shows the experiment configuration.

A. Sensitivity Analysis

Sensitivity analysis is used to determine which factor has more impact on the model [25]. This section presents the results of two factors. The first analysis is for k value, the number of neighbors for each node, which is expected to have an impact on the reputations as the original DRank only sends the reputation values to direct neighbors. From the results shown in Fig. 3, it is clear that there is a great impact after number of neighbors increases from 7 to 14. However, the impact is much smaller after the number of neighbors is increased to 21.

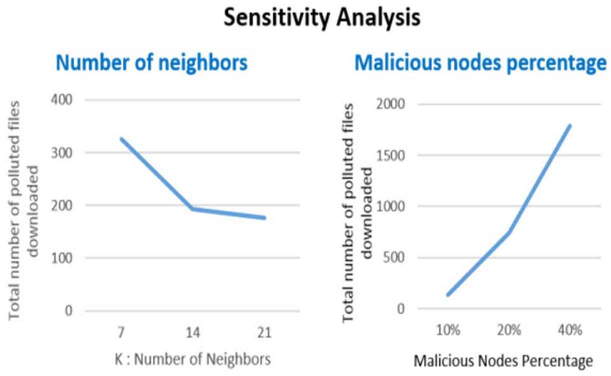

Fig. 3. Sensitivity analysis

Although in both cases 7 nodes were added, above 14 the impact dropped. This might be a sign that the system can improve to a certain level, in our case 14, and that there will be only a slight improvement despite the rise in the number of neighbors.

The second analysis is for the percentage of malicious peers. Three proportions were tested: 10%, 20%, and 40%, with a total of 500 nodes in the network. As shown in Fig. 3, the improvement is almost steady aside from the slight increase after 20%. This is an expected result as the more malicious nodes there are in the networks, the more polluted files will be sent.

B. Configuration

After analyzing the simulator, the following configurations are chosen:

- Number of replications: A replication is a run of an experiment with specific parameters. Two methods were used to determine how many replications were needed, confidence interval and the graphical method. It was found that 20 replications was the most appropriate number, giving 95% accuracy of the average value.
- Number of neighbors: In this study, we decided to set the number of neighbors to 7 (for the 500-node network and 10 for the 700-node network) as we wanted to test the worst case by minimizing the connections between the nodes so that the reputation values take longer to be shared over the network.
- Percentage of malicious peers: After performing the sensitivity analysis on the percentage of malicious peers, it was found that this parameter is positively correlated with the number of polluted files shared. Therefore, we chose 20% malicious peers in our experiments; the relation can be applied for a higher percentage.
- Number of cycles: To specify the number of cycles that the simulator should run in each experiment, we examined different numbers of cycles (max 40) and we found that the system persists with the same pattern. Therefore, 20 cycles were used and persistence is assumed.

978-1-5090-5443-5/17/$31.00 ©2017 IEEE

VI. Results

This section presents the results obtained from the experiments.

A. Unstructured Reputation Propagation

The reputation values were investigated in a 500-node network and the number of polluted files downloaded and sent was compared in both the original and the enhanced technique. From Fig. 4 it is clear that unstructured reputation propagation decreases the total number of polluted files. Moreover, the chart shows that the number of polluted files sent as a percentage of all file is also lower by around 4% percent.

B. Peer Selection Method

Two methods of peer selection were evaluated. The first is selecting the first best peer and the second is randomly selecting one of the groups of best peers. Fig. 5 shows the results of a simulation of a 200-node network that was run 20 times for each scenario and the requests each node received were collected and analyzed. Fig. 5 shows that the minimum number of requests is 0 in all experiments, meaning that at least one node did not receive any requests. This is expected, especially when the network size increases. However, the maximum number of requests of the randomly selected best peers is lower than the number of requests using first best peer selection, although the difference in the unstructured propagation is very small. The variance showed a similar pattern to the maximum even though the difference in unstructured propagation is larger. This shows that our goal of reducing the load on the best nodes is achieved to some extent. The only concern is that unstructured propagation shows higher values than the original system. The maximum number for unstructured propagation is around 480, which means at least one node received 480 requests out of 3800 (12% of total requests). The reason behind this could be that the reputations are shared more in unstructured propagation, which means that peers know more about good peers and try to choose them.

C. Network Size

This paper aimed to evaluate the original DRank in larger networks and to compare it with our enhanced DRank. Fig. 6 shows DRank performance in networks of 200, 500, and 1000 nodes. It is clear from the figure that the performance of DRank decreases dramatically when the network size increases. This behaviour is expected because when the network becomes larger, the reputations barely spread, especially as the peers only share their own experience.

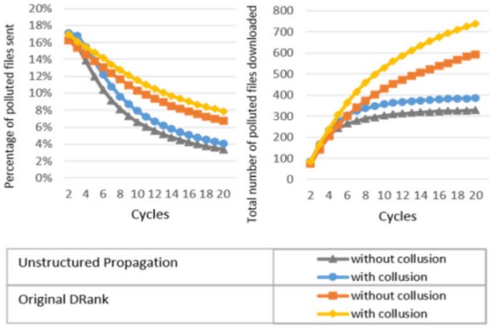

Fig. 4. Comparison of polluted files downloaded with both original and unstructured reputation propagation

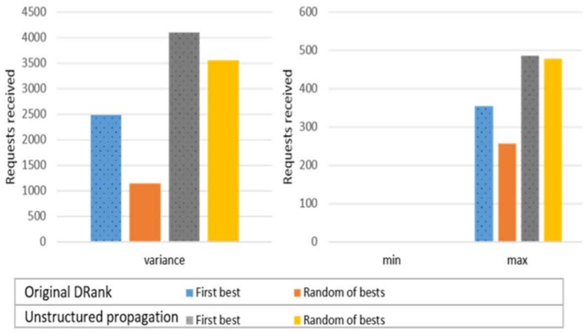

Fig. 5. Peer selection method results

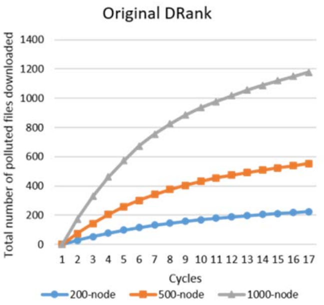

Fig. 6. Comparison of polluted files downloaded with 200-, 500-, and 1000-node networks in original DRank

Moreover, peers only send their reputation to their direct neighbors, which we tried to improve in enhanced DRank. Fig. 7 shows the total number of polluted files downloaded for the 500- and 700-node networks. From that we can demonstrate the effectiveness of the enhanced DRank in a medium-sized network up to 700 nodes.

D. Collusion Attack

Lastly, one focus of this paper is to observe the effect of a collusion attack on the P2P file sharing network. A collusion attack scenario was compared with the normal scenario in almost all experiments. In every instance, the performance of the reputation system was lower in the presence of a collusion attack. Fig. 4 and Fig. 7 show that in both the original DRank and with unstructured propagation, a higher rate of polluted files is downloaded with the collusion attack.

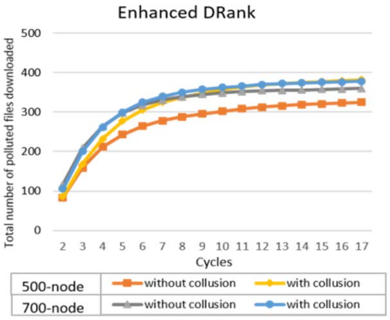

Fig. 7. Comparison of polluted files downloaded with 500- and 700-node networks in enhanced DRank

Although collusion attacks have an effect on DRank performance, this effect is clearly addressed so that it does not

978-1-5090-5443-5/17/$31.00 ©2017 IEEE

expand to a higher level. This is because the line of DRank in the presence of a collusion attack always has the exact same shape as the line of DRank but in a higher position. This means that the malicious peers could not control the whole network even though the attack scenario was running during the entire experiment.

VII. Conclusion and Future Work

In this paper, we focused on P2P file sharing systems and the use of reputation systems to mitigate the effect of malicious peers on the performance of the system and on the file sharing process. Peers participating in P2P networks cannot distinguish between malicious and good peers, which leads to peers downloading polluted files from untrusted peers. Therefore, reputation systems were proposed and reviewed in recent years to provide a way to recognize a malicious peer.

DRank is a reputation system designed for P2P media streaming and has already been shown to robustly fight network pollution in P2P streaming [9]. Those results were based on a simulation of a 200-node network in a fully simulated network architecture. In this paper, we built a DRank simulator in the P2P file sharing system using PeerSim in an abstracted underlying network architecture. The DRank reputation sharing mechanism was evaluated and compared with our introduced unstructured propagation mechanism. Unstructured propagation reduced the number of polluted files downloaded and improved the performance of DRank in the long term due to the expanded sharing it provides. Peer selection was also investigated in two ways: (1) selecting the first peer in the rank and (2) selecting randomly from a set of top-ranked peers. The second method showed better results for both the original and unstructured propagation, although it produced a higher value with the original one. The suggested reason is that the reputations are shared more in unstructured propagation, which means peers know more about good peers and try to choose them.

This paper showed that our proposal of unstructured propagation and random selection improves DRank performance. Because our results were obtained with network sizes of up to 700 nodes, performance for networks with more nodes requires further investigation in the future. Moreover, malicious peers could not control the whole network in the presence of a collusion attack model, as slight effects remained. This should be investigated in the future.

Acknowledgment

I would like to thank PeerSim developers for their great work in developing PeerSim and for the support they provide to users.

References

[1] S. Marti and H. Garcia-Molina, "Taxonomy of trust: categorizing P2P reputation systems," Comp. Networks, vol. 50, no. 4, pp. 472–484, March 2006.

[2] C. Costa and J. Almeida, "Reputation systems for fighting pollution in peer-to-peer file sharing systems," in IEEE Int. Conf. Peer-to-Peer Computing, Galway, 2007, pp. 53–60.

[3] E. Zhai et al., "Sorcery: Could we make P2P content sharing systems robust to deceivers?," in IEEE 9th Int. Conf. Peer-to-Peer Computing, Seattle, WA, 2009, pp. 11–20.

[4] N. Christin, A. S. Weigend, and J. Chuang, "Content availability, pollution and poisoning in file sharing peer-to-peer networks," in The 6th ACM Conf. Electronic Commerce, New York, NY, 2005, pp. 68–77.

[5] S. D. Kamvar, M. T. Schlosser, and H. GarciaMolina, "The Eigentrust algorithm for reputation management in P2P networks," in 12th Int. Conf. World Wide Web, New York, NY, 2003, pp. 640–651.

[6] L. Xiong and L. Liu, "PeerTrust: Supporting reputation-based trust for peer-to-peer electronic communities," IEEE Trans. Knowl. Data Eng., vol. 16, no. 7, pp. 843–857, July 2004.

[7] K. Walsh and E. G. Sirer, "Experience with an object reputation system for peer-to-peer filesharing," in the 3rd Conf. Networked Systems Design & Implementation, Berkeley, CA, 2006.

[8] K. Chen and K. Sapra, "A social network based reputation system for cooperative P2P file sharing," IEEE Trans. Parallel Distrib. Syst., vol. 26, no. 8, pp. 2140–2153, August 2015.

[9] M. Tauhiduzzaman and M. Wang, "Fighting pollution attacks in P2P streaming," Comput. Network, vol. 79, pp. 39–52, 2015.

[10] ebay. [Online]. http://www.ebay.co.uk/

[11] Amazon. [Online]. http://www.amazon.co.uk/

[12] L. Mekouar, Y. Iraqi, and R. Boutaba, "Reputation-based trust management in peer-to-peer systems: Taxonomy and anatomy," in Handbook of Peer-to-Peer Networking. NY: Springer, 2010, pp. 689–732.

[13] Q. Lian et al., "An empirical study of collusion behavior in the maze P2P file-sharing system," in 27th Int. Conf. Distributed Computing Systems, Toronto, ON, 2007, p. 56.

[14] Z. Li, H. Shen, and K. Sapra, "Collusion detection in reputation systems for peer-to-peer networks," in 41st Int. Conf. Parallel Processing, Pittsburgh, PA, 2012, pp. 98–107.

[15] S. Buchegger and J.-Y. Le Boudec, "A robust reputation system for P2P and mobile ad-hoc networks," in Proc. 2nd Workshop Economics of P2P Systems, Cambridge, MA, June 2004.

[16] W. J. Adams and N. J. Davis, "Toward a decentralized trust-based access control system for dynamic collaboration," in Proc. 2005 IEEE Workshop on Information Assurance and Security, West Point, NY, June 2005.

[17] Y. Wang et al., "Bayesian network based trust management," in Autonomic and Trusted Computing, Third Int. Conf., ATC. LNCS, Vol. 4158. Springer: Wuhan, China, 2006, pp. 246–257.

[18] C. Zouridaki et al., "Robust cooperative trust establishment for MANETs," in Proc. Fourth ACM Workshop on Security of Ad Hoc and Sensor Networks, Alexandria, VA, October 2006.

[19] M. Tauhiduzzaman and M. Wang, "A system analysis of reputation-base defences against pollution attacks in P2P streaming," in IEEE 31st Int. Performance Computing and Communications Conf. (IPCCC), Austin, TX, 2012, pp. 152–161.

[20] B. Y. Zhao et al., "Tapestry: a resilient global-scale overlay for service deployment," IEEE J. Sel. Areas Commun., vol. 22, no. 1, pp. 41–53, January 2004.

[21] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan, "Chord: A scalable peer-to-peer lookup service for internet applications," in Proc. 2001 Conf. Applications, Technologies, Architectures, and Protocols for Computer Communications, New York, NY, 2001, pp. 149–160.

[22] Querycycle. [Online]. http://p2p.stanford.edu/

[23] NS-2. [Online]. http://www.isi.edu/nsnam/ns/

[24] PeerSim: A Peer-to-Peer Simulator. [Online]. http://peersim.sourceforge.net/

[25] S. Robinson, Simulation: The Practice of Model Development and Use. Chichester, England: John Wiley & Sons Ltd, 2004.

978-1-5090-5443-5/17/$31.00 ©2017 IEEE