The Knowledge Engineering Review, Vol. 29:4, 463–483.
doi:10.1017/S0269888914000174

Taxonomy of reputation assessment in peer-to-peer systems and analysis of their data retrieval

FARAG AZZEDIN

Information & Computer Science Department, King Fahd University of Petroleum and Minerals, Saudi Arabia;
e-mail: fazzedin@kfupm.edu.sa

Abstract

The need for reputation assessment is particularly strong in peer-to-peer (P2P) systems because the peers' personal site autonomy is amplified by the inherent technological decentralization of the environment. However, the decentralization notion makes the problem of designing a P2P-based reputation assessment substantially harder in P2P networks than in centralized settings. Existing reputation systems tackle the reputation assessment process in an *ad hoc* manner. There is no systematic and coherent way to derive measures and analyze the current reputation systems. In this paper, we propose a reputation assessment process and use it to classify the existing reputation systems. Simulation experiments are conducted and focused on the different methods in selecting the recommendation sources and collecting the recommendations. These two phases can contribute significantly to the overall performance owing to precision, recall, and communication cost.

1 Introduction

Peer-to-peer (P2P) systems have flourished in the Internet and have become the number one application in Internet bandwidth usage. P2P systems are used in different applications, ranging from file sharing, real-time communication, and up to searching for extraterrestrial intelligence.

Along with various benefits such as scalability, there are several top issues in P2P systems, among which are reputation issues (Rodriguez *et al.*, 2006; West *et al.*, 2010; Yan & Prehofer, 2010). Reputation issues are important as they affect the confidence of peers on the network (Azzedin *et al.*, 2006). Users must be given the ability to measure the trustworthiness of a transaction partner (Aberer & Despotovic, 2004). Benign nodes can lead to untrustworthy transactions (Yan & Prehofer, 2010). As such, benign nodes are vulnerable to risks because of unknown, incomplete, or distorted information (Quercia, 2009; West *et al.*, 2010).

Many popular P2P applications have not implemented a mechanism to assess peers' reputation. Therefore, it is difficult to prevent malicious transactions. An example of malicious transaction in P2P file-sharing applications is the pollution of popular files (Liang *et al.*, 2005). In distributed computing applications, this can have a major effect on the invalid results owing to improper computation by malicious peers. To incorporate reputation mechanisms in P2P systems, we need to consider the effectiveness and efficiency of those mechanisms. That is, we should balance between the overhead of reputation assessment and the capability to eliminate malicious transactions.

The field of reputation online has spawned the interest of a number of scholars in technical as well as non-technical fields. For example, groups such as iTrust (2008) aim to explore the role of trust and reputation and to explore effective ways to implement them, whereas groups such as Quercia (2009) design algorithms for portable devices to identify reputable nodes and learn from recommendations. Effective

reputation modeling is believed to be an enabler for a range of new computing services including enhanced e-commerce, ubiquitous computing, Grid computing, P2P computing, and probably a variety of collaborative and cooperative online activities. Many researchers have proposed reputation systems (Aberer & Despotovic, 2004; Aringhieri *et al.*, 2006; Azzedin *et al.*, 2006; Rodriguez *et al.*, 2006) to assess the trustworthiness of a peer based on the recommendations from other peers in the P2P environment.

From a practical point of view, reputation (referred to as second-hand information, referrals, or ratings) schemes are already being used in many successful commercial systems such as eBay, BizRate, and Amazon. These online commercial systems allow their clients to give recommendations on other members or other resources.

1.1 Motivation and contributions

Various problems exist in practical as well as academic reputation systems. These problems are tackled in an *ad hoc* manner and there is no systematic and coherent way to derive measures and analyze the current reputation systems (Azzedin *et al.*, 2006; Jsang *et al.*, 2006). This motivated us to propose a reputation assessment process and evaluate its effectiveness to accurately predict the reputation of a target peer while minimizing the overhead of collecting, filtering, adjusting, and aggregating the recommendation requests.

This paper contributes to the reputation issue in P2P environments by proposing a reputation assessment process with the following features: (a) the proposed reputation assessment entails various functions that traditionally have been tightly coupled, which is not only an obstacle in identifying the common vulnerabilities of reputation systems but also makes the analysis and comparison of reputation systems hindering to identify emerging trends and open research issues; and (b) the proposed reputation assessment enables us to address the focus of each reputation system and either inject strengths or remedies into the reputation system. We develop a reputation assessment process that can be used as a classification, comparison, and analysis tool for reputation systems.

For the rest of this paper, we will use the following terminology: (a) trustworthiness to describe the behavior of a peer, (b) a peer that has the first-hand information is called a witness, (c) a recommender is a peer to whom a recommendation request is sent. A recommender can be a witness or relaying first-hand information from witnesses, (d) honesty is a term attached to recommenders. A recommender is honest if it gives what it beliefs in (i.e. it does not lie), and (e) if peer x wants to interact with peer y, we say that peer x is a source peer and peer y is a target peer.

2 Related work

A trust model for P2P computing environments is presented in Azzedin *et al.* (2006). This model uses a determined set of recommenders. The recommenders will provide recommendations based on their direct experience with the target peer. An honesty mechanism is also presented to filter the recommenders set. The recommendations are then manipulated based on the recommenders' accuracy. Accuracy is calculated based on the difference between the recommendation and the actual trust level of the target peer in the transaction. The manipulated recommendations are then aggregated using the average of the recommendations.

Gupta *et al.* (2006) propose a reputation system for unstructured P2P file-sharing systems. The reputation is computed with a debit and credit concept with two methods of reputation tracking differing only in the trade-offs between reliability and overhead. Their approach uses a special node, reputation computation agent (RCA), to which peers can enroll to participate in the reputation system. Although the enrollment is voluntary, the RCA will only process transactions between enrolled peers so that a resource provider will prefer to have transactions with enrolled peers in order to claim the upload credit. RCAs keep the debit part of the reputation so that peers cannot discard it.

Another model for unstructured P2P systems is presented in Aringhieri *et al.* (2006). A peer assigns a Boolean satisfaction value for a transaction to indicate the outcome of the transaction. If previous experience exists, the satisfaction value is incorporated with past experiences using freshness value, based on accuracy of current trust compared with latest transaction. To check the target peer's reputation, the

source peer sends a broadcast message to ask for recommendations. Then, the replies are verified to identify fake recommendations. Ordered weighted average (OWA) is used to aggregate the remaining recommendations. In addition to the recommendations, the source peer's own experience with the target peer is used.

To detect dishonest peers, Hughes *et al.* (2006) introduce the concept of suspicious transactions. A suspicious transaction is a transaction whose feedback is different from the one expected based on the target peer's reputation. To enable peers to move to different supernodes, each peer keeps a copy of his reputation. The data are encrypted using the supernode's secret key. Assuming every supernode has the public keys of other supernodes, the new supernode can check if the reputation data has been tampered with.

EigenTrust is proposed in Kamvar *et al.* (2003). It assigns a global trust value to every peer in a structured P2P file-sharing system. The system uses the value to choose the peers to download from. This way, untrustworthy peers are isolated. A peer's global trust value is computed and stored by other peers assigned as its score managers. A distributed hash table (DHT) is used to assign peers to score managers by hashing the peers' unique ID. To get the target peer's global trust value, the source peer contacts the target peer's score managers. If they return with different values, the majority value is chosen. After the transaction, the source peer submits its local trust value to the target peer's score managers. The local trust value is weighted based on the source peer's reputation and aggregated to the reputation.

Hou *et al.* (2005) propose another reputation system using P-Grid (PG) to store the reputation information in a distributed manner. The calculation of reputation is based on the number of successful and unsuccessful transactions. Using confirmation theory, the value of reputation ranges from -1 (no satisfactory transaction) to 1 (no unsatisfactory transaction). After a transaction, the source peer recalculates its own experience with the target peer and submits it to the peers responsible to store the target peer's reputation. In aggregating the target peer's reputation, the source peer's own experience is weighted by the source peer's reputation.

FuzzyTrust, a system to evaluate peer reputation in P2P e-commerce transactions, is proposed in Song *et al.* (2005). For a transaction, the buyer performs fuzzy inference to calculate his local score toward the seller and vice versa. The buyer uses goods quality and delivery time in his inference, while the seller uses payment method and payment time. Global reputation is calculated by aggregating the local scores from all peers who have interacted with the target peer. Three pieces of information are collected from every recommender: the recommender's reputation, transaction date, and the transaction amount. These three variables are used in fuzzy inference rules to compute the weights of the recommendations. Only recommendations whose weights are over certain threshold are aggregated. The aggregation uses weighted average with the weights normalized so that their sum is one.

In Mekouar *et al.* (2006), a reputation system is proposed for P2P file-sharing systems, that uses a hierarchy such as FastTrack. Other than isolating untrustworthy peers, the proposed system also aims to improve peers' satisfaction. A peer's reputation and satisfaction are maintained by its supernode. Before downloading a file, the source peer requests its supernode for the target peer's reputation. The supernode then contacts the target peer's supernode for the information. It is assumed that supernodes know and trust each other. After the download is completed, the source peer submits a feedback about the target peer to the target peer's supernode, indicating the file size and if the file is trustworthy or not.

Reply consistency is used in Azzedin and Maheswaran (2003) to predict honesty; consistent peers are assumed to be honest and vice versa. Each peer has a set of trusted allies through whom a consistency check is performed. The checking is done by asking one or more of the trusted allies to send a recommendation request for the target peer to the recommender. The source peer would compare the recommendation it gets directly with the one received by the trusted allies. Assuming the requests come in a relatively short time, the recommender should give answers with no or little value difference. Therefore, if the difference is more than a certain threshold, the recommender is being inconsistent. The recommender would be replaced from the source peer's recommender list and marked as dishonest so that it would not be included again in the list. However, other peers are not informed about this marking, so the inconsistent peer can still be part of other peers' recommender sets.

A method to filter out dishonest feedbacks for Bayesian reputation systems such as Jøsang and Ismail (2002) is presented in Whitby *et al.* (2005). Honesty checking is performed by identifying outliers in the retrieved recommendations. An outlier is a recommendation, providing the number of trustworthy and

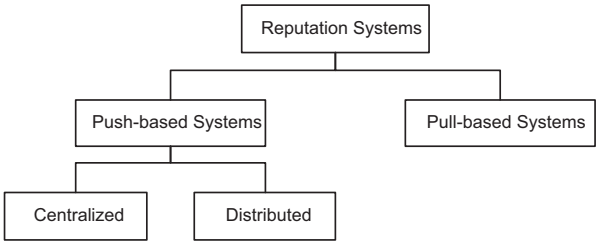

Figure 1 Classification of reputation systems

untrustworthy transactions, whose expected probability is less than a certain quantile, q, or is greater than $(1-q)$ quantile of the β distribution resulting from all of the recommendations. The checking is conducted iteratively until no outliers are found.

Jin *et al.* (2007) propose a filtering method that requires both the source peer and the target peer to provide a feedback on the target peer's behavior in their transaction. A source peer's feedback is considered consistent if it agrees with the target peer's feedback. Assuming most of the peers are trustworthy and honest, most inconsistencies would be in the case that the source peer reports a trustworthy transaction as untrustworthy (badmouthing) or an untrustworthy transaction as trustworthy (collusion to boost the target peer's reputation). Thus, a source peer is considered to be dishonest if the proportion of consistent feedbacks is less than a certain threshold, T_θ. Feedbacks from such peer are no longer accepted.

The reputation system in Selçuk *et al.* (2008) uses a measurement called credibility factor. The credibility factor increases if a recommender provides a recommendation that matches the actual result of the transaction. From the credibility, discredibility factor can be derived. A recommender whose discredibility factor is higher than its credibility factor will be filtered out.

3 Reputation systems taxonomy

The objective of a reputation system is to assess the reputation of a target peer. The assessment result would help peers to decide whether to proceed with a transaction or to cancel it. It adds the dimension of trust in selecting a transaction partner. If the assessment result accurately describes the behavior of the target peer, it would direct transactions to trustworthy peers and save the source peer from the risk of transacting with untrustworthy peers.

Recommendations has an important role in reputation systems and, hence, we base our classification on how are these recomendations injected into the reputation system. Reputation systems can be classified as push-based and pull-based systems. In push-based systems, witnesses would push their feedback to certain storage peers as performed in Aberer and Despotovic (2001), eBay (2013), Kamvar *et al.* (2003), Mengshu *et al.* (2005) and Xiong and Liu (2004). On the other hand, in pull-based reputation systems, recommenders would keep their feedback to themselves until this feedback is pulled, as done in Azzedin *et al.* (2006) and Patel *et al.* (2005). The general classification of reputation systems is shown in Figure 1.

Push-based reputation systems can be further classified as centralized and distributed reputation systems. In centralized push-based reputation systems such as *eBay*, *Amazon*, and *BizRate*, there is a single authority that manages the recommendations. Distributed push-based reputation systems distribute the recommendations utilizing the design of the overlay network.

4 Reputation assessment process

The reputation assessment process is shown in Figure 2. Recommendation retrieval in push-based systems is performed by contacting the nodes responsible to store the feedbacks pushed by other nodes. In pull-based systems, a peer needs to collect the feedback from the recommenders in order to compute the target peer's reputation. The recommender filtering component tries to improve the quality of the recommendations by eliminating recommenders who are considered to be not useful. In the recommendations evaluation component, recommendations are adjusted to improve the accuracy of the recommendations and aggregated

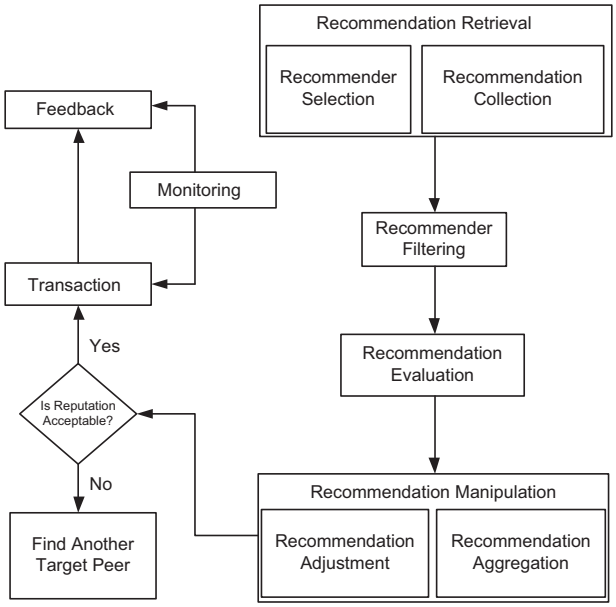

Figure 2 Reputation assessment process

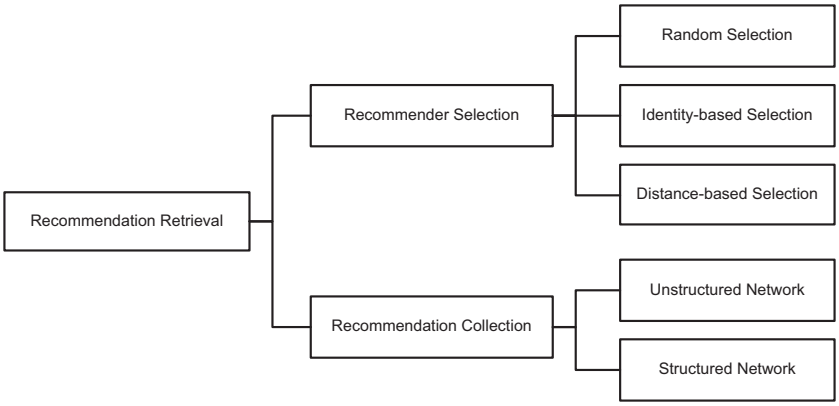

Figure 3 Recommendation retrieval

to obtain a single value predicting the target peer's reputation. If the predicted reputation is above a certain threshold, the transaction is committed.

4.1 Recommendation retrieval

Recommendation retrieval is crucial in the reputation assessment process because the subsequent components rely on it. The efficiency as well as the cost of recommendation retrieval depend on the overlay network. Recommendation retrieval involves two processes: recommender selection and recommendation collection. Recommender selection specifies the peers that should become recommenders. In recommendation collection, recommendation requests are sent to the selected recommenders and replies are returned to the source peer. A reputation system can rely on the resource discovery mechanism for this component as in Selçuk *et al.* (2008). The components of recommendation retrieval and its classification are illustrated in Figure 3.

4.1.1 Recommender selection

Recommender selection can be random, distance based, or identity based. In the random approach, the system determines the number of recommenders that it considers needed to assess the reputation of the target peer. Then, the recommenders are chosen randomly.

Distance-based approaches select the neighboring peers within a certain distance from the source peer. Recommendations are requested from those peers, regardless of their identities or the identity of the source peer and the target peer. The same peers would be selected as recommenders by the source peer in performing reputation assessment as long as they are still within the distance. If the specified distance is less than the network diameter, there is no guarantee the selected recommenders would include all witnesses in the network. Thus, retrieved recommendations may not provide the complete picture of the target peer's behavior. In other words, the predicted reputation from those recommendations may not be the actual global reputation.

On the other hand, identity-based approaches relate the recommender selection to the identities of the recommenders or to the identity of the source peer or the target peer. A source peer may choose to ask recommendations only from certain recommenders, for example, from those who are considered to be honest. This approach does not result in a global reputation value by design. The objective is to get recommendations only from desirable recommenders. Aiming for global reputation, peers may also be selected as recommenders if they have interacted with the target peer. Thus, the selected recommenders would vary depending on the identity of the target peer.

Random selection is used in Sen and Sajja (2002). The number of recommenders is determined such that the majority of the selected recommenders would be honest. The method, however, assumes that the majority of peers are honest and the number of dishonest peers is known, which is impractical.

Distance-based selection approaches are practically the main choice in systems using completely unstructured networks. In those systems, such as Gnutella, peers do not have the routing information to reach other peers. Hence, there is no guarantee for a peer to be able to contact another peer efficiently. This method is used in Aringhieri *et al.* (2006).

An identity-based approach is proposed in Azzedin *et al.* (2006) using a recommender set. The set only changes if a recommender is found to be dishonest. The use of a recommender set involves the following cost: (a) storage cost to keep the list of recommenders; (b) communication cost to check the recommenders' heartbeat; and (c) incentives for recommenders to be part of the recommender set.

Another identity-based approach is to select all peers that have transacted with the target peer. This approach is used in many reputation systems (Aberer & Despotovic, 2001; Kamvar *et al.*, 2003; Bhargava *et al.*, 2004; Xiong & Liu, 2004; Mengshu *et al.*, 2005; Song *et al.*, 2005; Gupta *et al.*, 2006; Mekouar *et al.*, 2006); as it targets the global reputation. Social network analysis is used in Sabater and Sierra (2002) to select the recommenders from peers that have transacted with the target peer. However, this approach requires the information about the peers' connectivity, which is difficult to obtain.

4.1.2 Recommendation collection

After recommender selection determines who should be the recommenders, recommendation collection performs the actual information collection. It involves sending request messages to other peers and receiving the replies. This part is directly tied to the overlay network because the topology dictates how a peer can reach another peer. Hence, it is important to use the appropriate recommendation collection method to obtain the desired recommendations. Likewise, the choice of the overlay network also plays an important role.

In collecting the recommendations from the selected recommenders, some reputation systems do not specify the method as in Bhargava *et al.* (2004) and Gupta *et al.* (2006). They usually rely on the resource discovery mechanisms. For any P2P system, recommendation collection would depend on the overlay network to send and receive messages.

Flooding is used in completely unstructured networks as in Aringhieri *et al.* (2006). Communication is conducted by sending messages to directly connected peers. The messages are then forwarded to other peers until a certain limit is reached in order to avoid infinite loops. The number of hops is tracked using time-to-live (TTL) count, which is subtracted in each forwarding. A peer only forwards a message if the TTL is greater than 0.

A *recommendation tree* approach is described in Azzedin *et al.* (2006). The source peer sends recommendation requests to its set of recommenders. If a recommender has the information, it will provide recommendations based on its past experience with the target peer. Otherwise, it forwards the request to its recommenders until a recommendation is found or a request loop is detected.

Storage peers are used in reputation systems that use all witnesses in the recommender selection. It is the method of choice in push-based systems. Most of such systems rely on a structured network infrastructure in storing the recommendations such as in Aberer and Despotovic (2001), Mengshu *et al.* (2005), Kamvar *et al.* (2003), Song *et al.* (2005), and Xiong and Liu (2004). The structured network enables all peers to be reachable in a bounded number of hops. Many DHT-based networks guarantee $O(\log n)$ hops to any other peer (Wang *et al.*, 2005). However, care must be taken to observe the overhead of keeping the structure stable, especially if the churn rate is high (Qiao & Bustamante, 2005). If replicas are used to store recommendations, they need to be synchronized. This can be difficult when they are connected to the network intermittently.

Storage peers can also be implemented in unstructured networks such as the proposed system in Gupta *et al.* (2006), where peers called RCA are used as storage peers. However, it was not specified how to select the nodes that act as the agents. An unstructured network with a hierarchy, as in *FastTrack* network, is used in Mekouar *et al.* (2006) with the *supernode* acting as a storage peer for regular peers connected to it. The recommendation request from a source peer would be directed to the target peer's supernode. If the transaction takes place, the source pushes the feedback to the target peer's supernode. The approach assumes that the supernodes know each other.

4.2 Recommender filtering

Recommendation-based reputation assessment rely on the recommenders to provide the information about the target peer. Therefore, the attributes of the recommenders can affect the reputation assessment. The objective of recommender filtering is to avoid recommenders with undesirable attributes because contacting them results in the pollution of the reputation assessment result.

In pull-based reputation systems, undesirable recommenders would mean waste of the communication costs incurred to contact them, whereas in push-based reputation systems, recommendations from undesirable recommenders would waste the space and computing power of the storage peers. If the storage peer provides individual recommendations to the source peer, it would also result in waste of bandwidth. These additional benefits are not applicable in systems using distance-based recommender selection as in Aringhieri *et al.* (2006), because the request would be broadcasted indiscriminately. Recommender filtering in such systems can only be used to produce a recommender blacklist to weed out the recommendations after they are collected, but not to avoid the waste of bandwidth.

4.2.1 Recommender attributes affecting the reputation assessment

A recommender may be unwilling to participate in the reputation assessment by not replying to recommendation requests. Unwillingness can be identified by the lack of response to recommendation requests.

Activeness is another attribute that can affect a recommender's recommendations. An inactive recommender is less likely to provide an up-to-date recommendation, so it may not reflect the current behavior of the target peer. Activeness can be measured from the number of transactions.

A very important attribute that affects the reputation assessment is honesty. Dishonest recommendations will make the computed reputation useless. Dishonesty may be in the form of colluding to boost the reputation of an untrustworthy peer. It may also be badmouthing in order to destroy a trustworthy peer's reputation. However, this issue is not handled in some reputation systems (Aberer & Despotovic, 2001; Sen & Sajja, 2002), because they assume that the majority of the peers are honest and, therefore, cancel the effect of dishonest ones on the recommendation network.

The systems that apply recommender filtering focus on the honesty attribute because dishonest recommenders would distort the reputation assessment so that untrustworthy peers are misidentified as trustworthy and vice versa. Such distortion would result in committing transactions with untrustworthy peers and canceling transactions with trustworthy peers.

Many reputation systems relate honesty to trustworthiness (Kamvar *et al.*, 2003; Bhargava *et al.*, 2004). Trustworthy peers are assumed to provide honest recommendations, whereas untrustworthy peers are to be dishonest. This assumption, however, is not necessarily correct. A trustworthy peer still has the motivation

to provide dishonest recommendations for its own interest, for example, stopping other peers from transacting with competing peers in an environment that provides monetary incentives to resource providers. A peer may be marked as untrustworthy because it cannot provide the expected level of service owing to something not under its control, such as unreliable connection. In this case, the peer may be honest in providing recommendations. Therefore, it is necessary to distinguish between trustworthiness and honesty as in Azzedin *et al.* (2006) and Jin *et al.* (2007).

4.3 Recommendation evaluation

Recommendations have to be evaluated in order to obtain a predicted reputation value. The predicted value can be used by the source peer as one of the criteria in deciding whether to have a transaction with the target peer. Recommendation evaluation is composed of two components, namely recommendation adjustment and recommendation aggregation. Recommendation adjustment is performed to improve the accuracy of the recommendation. The adjustment can be performed by assigning weights to indicate the importance of each recommendation in predicting the reputation. Finally, the recommendations are aggregated in order to come up with a single value.

The witness' trustworthiness is often involved in the weighting to measure the reliability of the recommendation. The assumption is that trustworthy peers provide honest and reliable recommendations, whereas untrustworthy peers provide dishonest and unreliable recommendations. However, this assumption is not necessarily true, as discussed in Section 4.2.

Recommendation evaluation can be performed by the source peer, the target peer or a third party such as the storage peer. This classification is shown in Figure 4. In push-based systems, recommendation evaluation is conducted at the storage peer except in Xiong and Liu (2004), whereas in pull-based systems, it is performed by the source peer.

The source peer in Xiong and Liu (2004) uses the weighted average as the aggregation scheme. The aggregation in Mekouar *et al.* (2006) is based on the upload, download, and query forwarding metrics as the system is designed for file sharing. It adjusts the feedback from the source peer based on the difference between the feedback and the expected behavior of the target peer. The upload, download, and query forwarding metrics are aggregated in Gupta *et al.* (2006) using a simple summation. The aggregation in Mengshu *et al.* (2005) uses confirmation theory to combine two recommendations and repeating it until a single value is obtained.

Recommendation adjustment can be performed by assigning weights to the recommendations to indicate their importance. Preference may be given to recommendations from peers known to be trustworthy, based on multiple interactions or to recommendations that are based on recent, frequent, and high-value transactions. The exact values and thresholds of transaction values, transaction frequency, and other weighting factors can vary in different contexts. Zero weights can be assigned to recommendations that do not meet the thresholds as they are regarded to be not useful. Systems that use simple average practically assign equal weights to the recommendations.

In some systems such as in Azzedin *et al.* (2006) and Bhargava *et al.* (2004), recommendation adjustment is performed based on the difference of the recommendation and the actual trust level after a transaction. Recommendation adjustment in Patel *et al.* (2005) is performed using a probabilistic distribution derived from the history of the recommender's accuracy, but this method involves heavy computation.

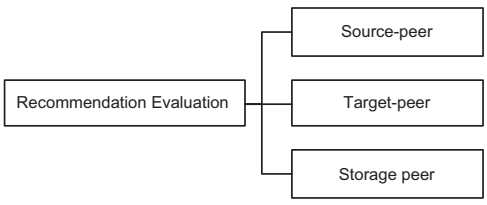

Figure 4 Recommendation evaluation

The recommendations would then have to be aggregated in order to get a value predicting the trustworthiness of the target peer. Voting is used in Sen and Sajja (2002); the target peer is considered as trustworthy if the majority of the selected recommenders considers the target peer to be trustworthy. A voting mechanism is also used in systems using replicas when there are conflicting reputation values from them as in Aberer and Despotovic (2001) and Kamvar *et al.* (2003). A heuristic function is also used in Aberer and Despotovic (2001) to compare the target peer's reputation to the average reputation in the system. Because the reputation system only uses negative recommendations, a peer is considered to be untrustworthy if it receives negative recommendations significantly more than the average.

A simple average is used in Azzedin *et al.* (2006) to aggregate recommendations, whereas a weighted average is used for aggregation in Bhargava *et al.* (2004) and Song *et al.* (2005). An OWA is used in Aringhieri *et al.* (2006) to aggregate recommendations with higher preference is given to recommendations with low trust level and the highest preference to direct trust.

Transaction frequency and timestamp of the recommendation are used in Azzedin *et al.* (2007) as input to fuzzy inference rules in determining the weights, extending the recommendation evaluation in Azzedin *et al.* (2006). The credibility factor in Selçuk *et al*. (2008) is not only used for recommender filtering, but is also used in weighting if the recommender is not filtered out. Fuzzy inference rules are also used in Song *et al.* (2005) with the transaction value and the witness' trustworthiness as weighting factors.

Witness' trustworthiness is used to determine the weight in Mekouar *et al.* (2006), Mengshu *et al.* (2005), Kamvar *et al.* (2003), and Xiong and Liu (2004). The calculation is performed iteratively in Kamvar *et al.* (2003), until the values converge. The evaluation approach in Xiong and Liu (2004) includes a similarity factor to measure the credibility of the witness. The similarity is computed by comparing the values of the witness' direct experiences and the source peer's direct experiences. Higher similarity in the intersection means higher weight to the witness' recommendation. The transaction value and the community context can also be included in the evaluation.

4.4 Recommendation manipulation

This phase calculates the reputation of the target peer based on the recommendation. The calculation involves two parts: adjustment and aggregation. Recommendation adjustment is performed to deal with subjectivity and accuracy problem. Once they are adjusted, there are several aggregation methods available such as average, weighted average, OWA, and so on. The weights from the recommendation evaluation phase are used in the aggregation part.

Recommendation adjustment has been dealt with differently by different researcher groups. Many reputation systems ignore the need of adjustment for recommendation (Aberer & Despotovic, 2004; Song *et al.*, 2005; Aringhieri *et al.*, 2006). The recommendations received from the recommenders need to be adjusted before being used. The reason for this is to enable peer review-based mechanisms to function with imprecise trust metrics, the imprecision is introduced by peers evaluating the same situation differently. In Azzedin *et al.* (2006) and by adjusting the recommendations, it was shown that the reputation-based trust model reaches an acceptable level of capability after a certain number of transactions. However, as the number of dishonest peers increase, the model becomes slow in reaching the acceptable level of capability. The issue of the dishonest recommenders is dealt with in the recommender filtering phase of the reputation assessment process.

Recommendation aggregation is a process that takes as input the individual recommendations from recommenders and their assigned weights of importance from the previous phase of recommendation evaluation. Its output is a single reputation value for the target peer based on the recommendations. There are two main directions of research in this recommendation aggregation process: (a) which aggregation algorithm should be used and (b) where the recommendations should be aggregated.

4.5 Transaction monitoring and feedback

In any transaction between peer x and peer y, there is a trust concern from x as well as from y. Peer x can monitor the transaction using offline, online, or combining offline and online mechanisms. Peer x observes

the transaction or the transaction records to determine whether any abuses have taken place by the target peer. Each peer should have special configuration to define what conditions exactly cause a breach in the transaction contract. Some example breaches include: (a) holding the resources for longer periods that initially requested, (b) trying to access protected local data, (c) instantiating illegal tasks on the resources, (d) reneging on promises to provide resources, and (e) going down during periods of peak usage. One way of monitoring the transaction is to analyze the data gathered from audit data generated by the operating system. Audit trails are particularly useful because they can be used to establish guilt of attackers. An audit trail is a record of activities on a system that are logged to a file in chronologically sorted order. As almost all activities are logged on a system, it is possible to manually inspect these logs and detect untrustworthy target peer.

Using transaction feedback, peer x can update its own local experiences with peer y that it interacted with. This feedback process is done for future references. For example, if peer x wants to interact with peer y that it had interacted with before, then peer x can evaluate its past experience with peer y. The feedback process can be extended to a complaint mechanism in the reputation assessment process. A complaint mechanism is very crucial to post the reputation of target peers in a place where other peers can access. Having a complaint mechanism is motivated by this example. If there is no complaint mechanism, then each peer keeps its direct experiences to itself. This set of peers will not propagate the information about the untrustworthy peer, unless they were asked for recommendations. This way, the untrustworthy peer might be globally undetected for sometime and be given chance to harm other peers.

Based on the reputation assessment presented, we classified the existing reputation systems as illustrated in Table 1.

5 Performance evaluation of recommendation retrieval

5.1 Overview

In the recommendation retrieval component, recommenders are selected and recommendations are collected as discussed in Section 4.1. We evaluate the various recommendation retrieval mechanisms in order to: (a) compare the effectiveness of various recommendation retrieval approaches in obtaining recommendations from witnesses, (b) compare the communication costs incurred by various recommendation retrieval approaches, and (c) analyze the performance of those approaches and provide recommendations in conducting recommendation retrieval.

5.2 Performance metrics

We conducted a series of simulation studies to examine the recommendation retrieval phase of the proposed reputation assessment process. The performance measures used in the simulation are:

- Precision (P). This metric is analogous to precision in information retrieval (Baeza-Yates & Ribeiro-Neto, 1999) and is calculated as follows:

$$P = \frac{C_W}{C} \times 100\,\% \tag{1}$$

  where C_W is the set of contacted witnesses and C the set of contacted peers.

- Recall (R). This metric is also analogous to recall in information retrieval (Baeza-Yates & Ribeiro-Neto, 1999) and is calculated as follows:

$$R = \frac{C_W}{P_W} \times 100\,\% \tag{2}$$

  where P_W is the set of all witnesses.

- F-Measure (F). This metric combines recall and precision using harmonic mean and is calculated as follows:

$$F = \frac{2 \times (R \times P)}{(R + P)} \times 100\,\% \tag{3}$$

Table 1 Classification of existing reputation systems

System	Recommendation retrieval	Recommender filtering	Recommendation evaluation
Aberer and Despotovic (2001)	Selection: all witnesses; collection: storage peer in structured network	N/A	Aggregation: voting and heuristic function
Aringhieri *et al.* (2006)	Selection: distance-based; collection: flooding	N/A	Weighting: trust level; aggregation: ordered weighted average
Azzedin *et al.* (2006)	Selection: recommender set; collection: recommendation tree	Consistency checking	Adjustment: accuracy based; aggregation: weighted average
Bhargava *et al.* (2004)	Selection: all witnesses; collection: unspecified	N/A	Adjustment: accuracy based; aggregation: weighted average
Gupta *et al.* (2006)	Selection: storage peer and target peer; collection: unspecified	N/A	Aggregation: simple summation
Jøsang and Ismail (2002)	Selection: all witnesses; collection: storage peer	N/A	Weighting: witness' reputation; aggregation: β distribution
Kamvar *et al.* (2003)	Selection: all witness; collection: storage peer	N/A	Weighting: witness' reputation; aggregation: weighted average
Mekouar *et al.* (2006)	Selection: all witness; collection: supernodes in unstructured network with hierarchy	N/A	Weighting: witness' reputation; aggregation: weighted average
Mengshu *et al.* (2005)	Selection: all witnesses; collection: storage peer in structured network	N/A	Weighting: witness' reputation; aggregation: confirmation theory
Patel *et al.* (2005)	Unspecified	N/A	Adjustment and weighting: probabilistic; aggregation: β distribution
Selçuk *et al.* (2008)	Unspecified	Discredibility factor	Weighting: credibility factor; aggregation: average
Sen and Sajja (2002)	Selection: random; collection: unspecified	N/A	Aggregation: voting
Song *et al.* (2005)	Selection: all witnesses; collection: storage peer in structured network	N/A	Weighting: transaction value and witness' reputation; aggregation: weighted average
Xiong and Liu (2004)	Selection: all witnesses; collection: storage peer in structured network	N/A	Weighting: transaction and community context, feedback similarity; aggregation: weighted average

- Communication Cost (Γ). This metric measures the cost incurred to collect the recommendation and is calculated as follows:

$$\Gamma = (S_m + R_m) \times D_a \tag{4}$$

where S_m is the number of sent messages, R_m the number of reply messages, and D_a the average depth of the tree.

Notice that $P = 0\%$ means none of the contacted peers are witnesses and $P = 100\%$ means that all of the contacted peers are witnesses. In addition, $R = 0\%$ means none of the witnesses are contacted and $R = 100\%$ means all the witnesses are contacted. Recall and precision are irrelevant for systems that store all recommendations about a peer at one or more storage peers as in structured P2P such as PG or DHT-based systems. If such systems are stable and the network is secure and reliable, both recall and precision for such structured P2P systems are 100%. Therefore, we can measure the performance of

```
1  Generate recommender list (for RT);
2  Construct overlay network;
3  Generate transaction matrix based on the acquaintance rate;
4  for each peer do schedule its first transaction;
5  while maximum number of transactions has not been reached do
6  |  Select the peer with the earliest scheduled transaction as the source peer;
7  |  Determine the target peer randomly;
8  |  Retrieve recommendations;
9  |  Update performance metrics;
10 |  Schedule the next transaction attempt for the source peer;
11 end
```

Figure 5 Flow of recommendation retrieval simulation.

Table 2 Exogenous and design parameters used in the simulation

Symbol	Definition	Values
N	Number of peers	$N = (2048, 4096)$
λ	Mean inter-arrival time	$\lambda = (0.1)$
AR	Acquaintance rate	$AR = (0.0 - 1.0)$
Num_{rec}	Number of recommenders	4
TTL	Time-to-live	7
$Tran_{max}$	Maximum number of transactions	$5N$

structured P2P systems using: (a) hop count: how many peers are contacted before reaching the storage peers and (b) the structure-maintenance overhead: the cost of keeping the structure stable, especially if the churn rate is high.

5.3 Simulation structure

The evaluation is performed using a discrete event simulation. The flow of the simulation is described in Figure 5. The requests initiating inter-peer transactions are assumed to have a Poisson arrival process where the mean inter-arrival time is based on the value used in Schollmeier (2005). The target peer for each transaction is randomly generated from $[1, N]$ using uniform distribution (Line 7). The TTL for flooding is based on the predominant value used in Gnutella (Ripeanu *et al.*, 2002; Ciraci *et al.*, 2005). For recommendation tree, the source peer would contact four recommenders. For PG, the request is sent to the storage peer responsible for the target peer. The parameters used in the simulation are listed in Table 2. Recommendations are sent individually by back-tracking the request path. The performance metrics are calculated for each transaction attempt and the results are the average of all transaction attempts.

We introduce the term *acquaintance rate* (AR), which is the average probability that one peer knows another. Higher AR means a peer has had transactions with more peers. In the simulation, we assume that the AR is the same for all peers. The value ranges from 0 (every peer has no transactions with others) to 1 (every peer has had transactions with all other peers). The transaction matrix is generated randomly based on the AR (Line 3).

5.4 Overlay network construction for flooding

In simulating flooding, we construct an unstructured overlay network using two algorithms. Algorithm I is based on the construction described in Schollmeier (2005). The algorithm is shown in Figure 6.

Algorithm I does not force all peers to form a single network. Instead, it results in multiple islands. For the reputation system simulation, we only take the largest island, which in average, consists of about 75% of the total peers. An example of the largest island in the overlay network generated by Algorithm I is shown in Figure 7. It has 784 peers (out of 1024 total peers) with a diameter of 17 and 57.82% are reachable with $TTL = 7$.

```
1  for each peer do assign maximum nodal degree;
2  repeat
3  |  Randomly select two distinct peers;
4  |  if both are not connected and have not reached their maximum degree then
5  |  |  Connect them to each other
6  |  end
7  until no peer can be connected without exceeding its maximum nodal degree ;
8  if there are peers not reaching their maximum degrees then
9  |  Add connections until all peers reach their maximum degree;
10 end
```

Figure 6 Algorithm I for unstructured overlay network construction

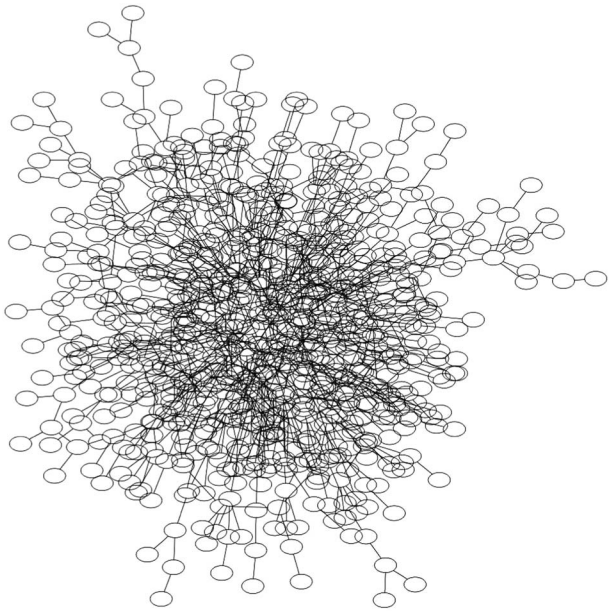

Figure 7 Unstructured overlay network using Algorithm I (largest island)

Comparing the overlay network construction using Algorithm I to the observation of the Gnutella network in Ripeanu *et al.* (2002), the portion of peers reachable with $TTL = 7$ for the largest island ranges from 45% to 64%, inversely related to the network size, whereas in Ripeanu *et al.* (2002) it was reported as more than 95%, regardless of the network size. The difference is because of construction of the overlay network. In the Gnutella network observed in Ripeanu *et al.* (2002), new peers would contact some predefined nodes in order to get the list of connected peers. This mechanism results in peers connecting to more highly connected peers and the overlay network would have relatively small diameter. On the other hand, Algorithm I pairs peers randomly.

In order to simulate the environment observed in Ripeanu *et al.* (2002), we develop Algorithm II to construct the overlay network. We use the average number of connections observed in Ripeanu *et al.* (2002). The algorithm is illustrated in Figure 8.

In the pairing, no peer can be connected to itself and a maximum nodal degree is imposed. Each peer has at least one connection (Lines 3–5). Algorithm II connects all peers to a single island and keeps the diameter relatively small compared with Algorithm I. The portion of peers reachable with $TTL = 7$ with this algorithm ranges from 98.5% to 99.3%, inversely related to the size of the network. An example of an overlay network generated by Algorithm II is shown in Figure 9. It has 1024 peers with a diameter of 11 and 99.29% are reachable with $TTL = 7$. Compared with the example shown in Figure 7, it has smaller diameter and higher reachability with $TTL = 7$, although the island has more peers.

In order to reduce duplicate sent messages in flooding, a peer does not forward a request to the peer from whom the request is received. I addition, if the same request has been received previously through a different path, it would only be forwarded if the TTL of the request is greater than the TTL of the previous

```
Select two peers randomly as the starting pair;
Place the initial pair in the connected peer list and assign equal probability to both;
for each peer do
    Connect it to a peer included in the connected peer list based on the probability;
    Add the peer to the connected peer list;
    Update the probabilities based on the nodal degree;
end
for each remaining connection do
    Select a peer randomly, connect it to another peer based on the probability;
    Update the probabilities based on the nodal degree;
end
```

Figure 8 Algorithm II for unstructured network construction

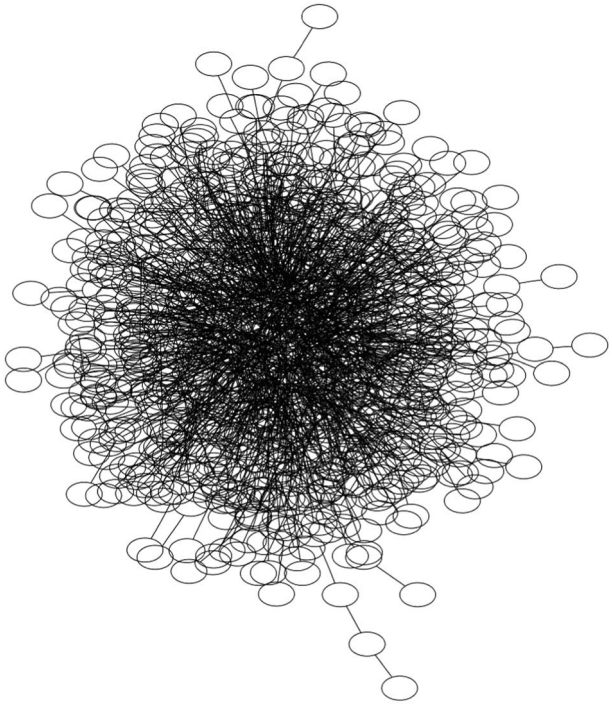

Figure 9 Unstructured overlay network using Algorithm II

request, because in this case, the request may reach peers that have not received the request through another path. If the TTL of the request is less than or equal to the TTL of the previous request, the request would reach peers that have received the request.

5.5 Overlay network construction for recommendation tree

In simulating recommendation retrieval using a recommendation tree, the overlay network is constructed based on the recommender list. Each peer is directly connected to its recommenders. This arrangement minimizes the communication cost in retrieving recommendations as a peer does not need to go through other peers to contact its recommenders. In contacting the recommenders, we simulate two approaches: depth-first search (DFS) and breadth-first search (BFS). Using DFS, the source peer sends a recommendation request to one of its recommenders and not to the other recommenders, until a recommendation is found or a loop is discovered. In this approach, the source peer only needs to open a single connection. On the other hand, using BFS, the source peer connects and sends the requests to four recommenders.

5.6 Overlay network construction for storage peer in structured overlay network

For retrieval in the structured overlay network using storage peers, we select PG (Aberer *et al.*, 2003). We measure the average number of hops needed to reach other nodes with different network sizes and find that

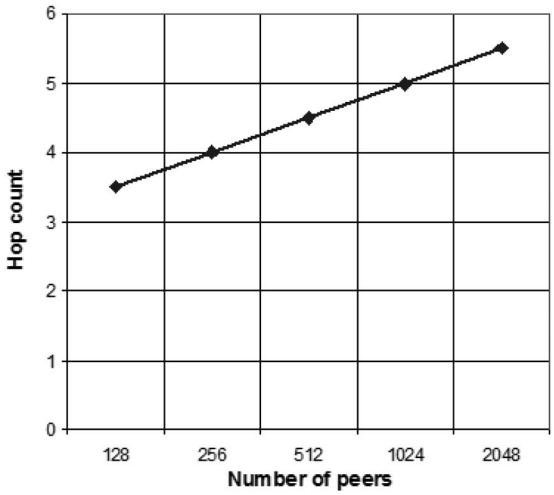

Figure 10 Average hop count in P-Grid

the average number of hops to reach another peer is $\log(N)/2$ where N is the number of peers as illustrated in Figure 10. In measuring the recall and the precision, we consider reaching the storage peer as equivalent to contacting all witnesses in the system.

5.7 Results and discussion

Figure 11 shows the recall values in flooding using Algorithm I (FL1), flooding using Algorithm II (FL2), recommendation tree using DFS (RT1), recommendation tree using BFS (RT2), and PG. The recall in FL1 and FL2 is stable because the number of contacted peers only depends on the network diameter and TTL. However, FL2 has higher recall owing to the overlay network construction algorithm. FL2 has a network diameter average of 11.4, much smaller compared with the network diameter average of 20.1 in FL1. In FL2, the node-to-node shortest paths with length ≤ 7 represent more than 95% of all the paths, whereas in FL1, only 47.83% of the node-to-node shortest paths. Thus, with $TTL = 7$, >95% peers have are reachable in FL2, whereas less than half of the peers in the network are reachable in FL1. The node reachability in FL1 and FL2 is illustrated in Figures 12 and 13.

Compared with FL1, more recommenders are covered in RT1 and RT2 for low AR because recommendation tree is not limited by a TTL. It results in higher recall than FL1. However, at high AR, the search tree in RT1 and RT2 is shrinking because the recommender is more likely to have a recommendation and the path is not expanded. Therefore, the recall plunges as the number of contacted witnesses decreases. The pattern is not affected by the choice of DFS or BFS. PG has a recall of 100% because it uses storage peers that provide the first-hand information from all witnesses.

Figure 14 shows that for FL1, FL2, RT1, and RT2, precision is directly related to the AR and relatively similar for all the methods with RT1 and RT2 performing slightly better at some points. The precision in RT1 and RT2 is slightly higher than flooding at high ARs because the precision is averaged over individual transactions. In some instances, the recommendations are found earlier, resulting in higher precision and affecting the average precision. However, if the precision is calculated from the average reached witnesses and contacted peers over all transactions, the average precision is similar to flooding. PG behaves differently from flooding and recommendation tree because it uses the storage peer approach. The precision in PG is the number of witnesses over the number of witnesses plus the number of hops needed to reach the storage peer.

The F-measure pattern is influenced by the pattern of precision and recall. F-measure in FL1 and FL2 increases if the AR increases owing to the stable recall and the increase in precision as shown in Figure 15. In RT1 and RT2, on the other hand, the F-measure is low for $\text{AR} > 0.6$ as the recall drops.

For the communication cost, the number of sent messages, as shown in Figure 16, is stable in FL1 and FL2, while it is inversely related to the AR in RT1 and RT2 because the tree shrinks as more peers

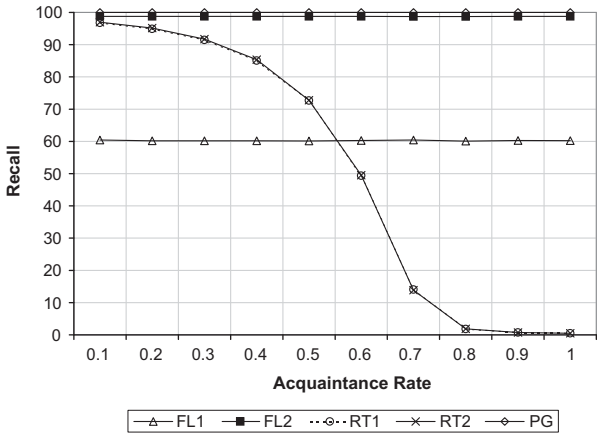

Figure 11 Recall values (1024 peers)

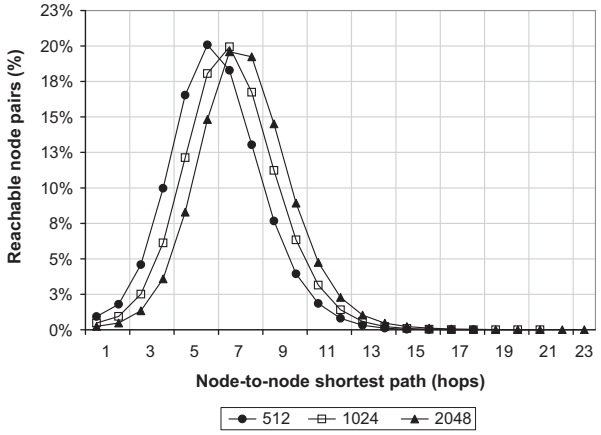

Figure 12 Node reachability (FL1)

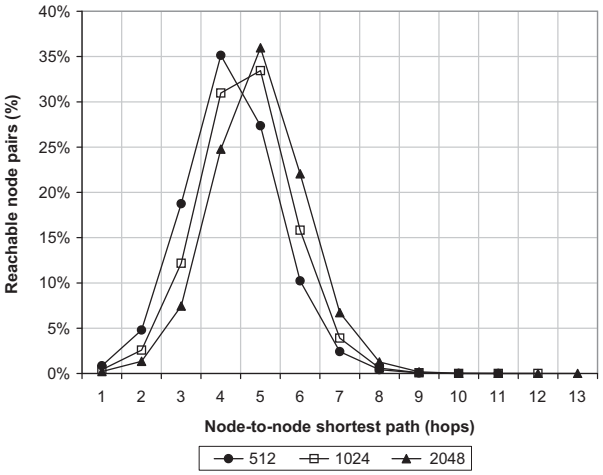

Figure 13 Node reachability (FL2)

have the requested information. The results also show that sending messages in unstructured networks is not efficient. Many peers receive the same request repeatedly showing cycles in the overlay network. The proportion of duplicate sent messages is shown in Figure 17. For PG, the number of sent messages is the

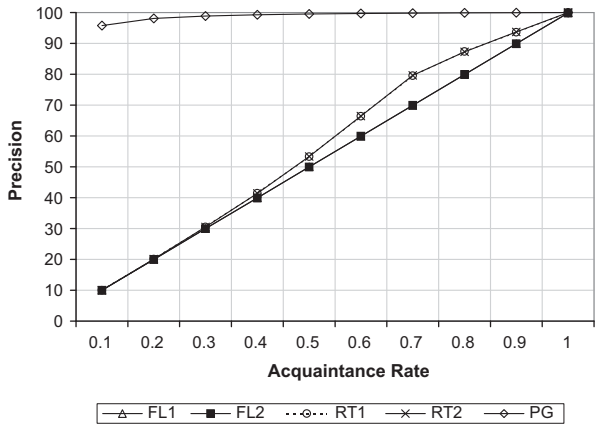

Figure 14 Precision values (1024 peers)

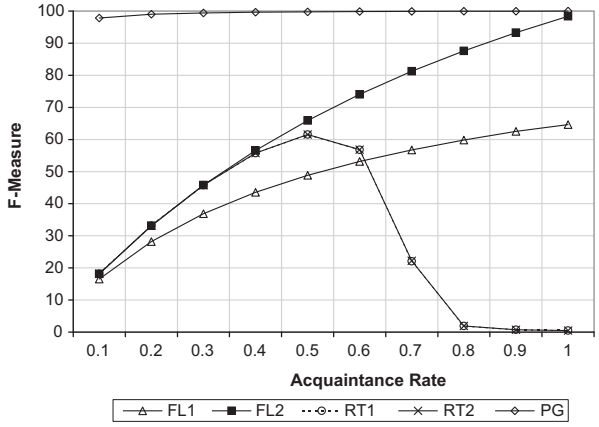

Figure 15 *F*-measure values (1024 peers).

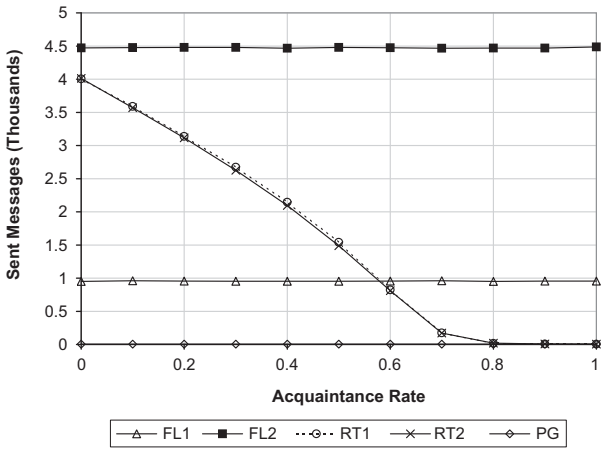

Figure 16 Number of sent messages for 1024 peers (in thousands)

same as the number of hops required to reach the storage peer illustrated in Figure 10. There are no duplicates in PG because the overlay network provides the routing information.

The number of replies is illustrated in Table 3. It is directly related to the AR in FL1, FL2, and PG. The number of replies in PG is the number of hops times the number of witnesses because the storage peer

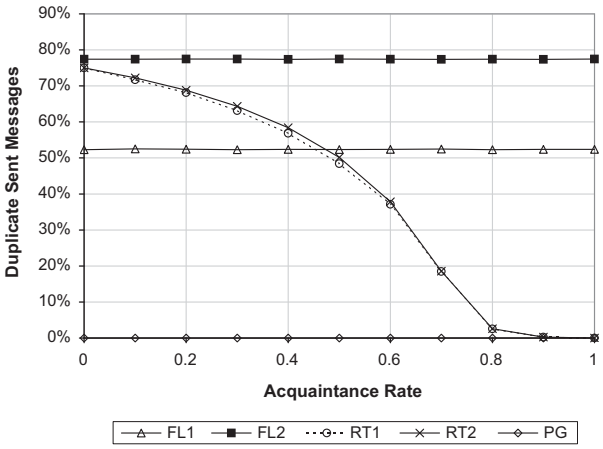

Figure 17 Percentage of duplicate sent messages (1024 peers)

Table 3 Number of reply messages (1024 peers)

AR	FL1	FL2	RT1	RT2	PG
0	0	0	0	0	0
0.1	292.383	667.722	13 223.902	522.590	511.775
0.2	583.846	1335.378	23 499.380	1098.790	1023.809
0.3	875.258	2003.175	29 654.390	1733.555	1534.555
0.4	1166.172	2671.899	30 249.613	2404.131	2046.869
0.5	1458.296	3338.896	23 835.676	2982.937	2560.445
0.6	1752.324	4008.446	11 481.575	3013.501	3070.747
0.7	2050.741	4674.883	1519.963	1038.353	3586.805
0.8	2330.137	5341.625	58.628	59.509	4090.142
0.9	2629.857	6008.429	9.768	9.750	4612.014
1.0	2920.050	6679.465	4	4	5127.588

provides individual recommendations. If the storage peer has aggregated the recommendations, the number of replies would be the same as the number of hops.

RT1 and RT2 have similar number of sent messages, but both have significantly different number of replies. The number of replies in RT1 is much higher than FL1 and FL2, except for high ARs ($\geq$0.7), whereas the number of replies in RT2 is lower than FL2 most of the time. The high number of replies in RT1 is owing to the unbounded depth of the tree and the DFS approach. When the AR is low, the first traversed path goes very deep before finding a witness or a loop. The reply would then have to backtrack the path resulting in a high number of messages for it to arrive at the source peer. This problem is alleviated by using BFS because sending the requests simultaneously increases the chance of a loop to be found. It results in a significant reduction of replies without affecting the number of contacted witnesses, as illustrated by the similar recall and precision results.

The difference between DFS and BFS in the recommendation tree is illustrated in Figure 18. Both trees have the same nodes and edges, so they have the same number of sent messages. However, they have different traversal order in reaching the witnesses. The edges are numbered based on the traversal order. The solid edges are the paths through which the witnesses ($P5$, $P7$, and $P8$) receive the recommendation request from the source peer ($P1$) for first time in the traversal order. Using DFS, the first recommendation request that arrives at $P5$ is through the path: $P1 \rightarrow P2 \rightarrow P6 \rightarrow P9 \rightarrow P5$; so it takes four messages for the recommendation to reach the source peer ($P1$). On the other hand, using BFS, $P5$ is reached earlier through the path: $P1 \rightarrow P5$, which only needs one message for the recommendation to reach the source peer. Overall, in this example, using DFS costs 10 reply messages, whereas using BFS takes only six reply messages.

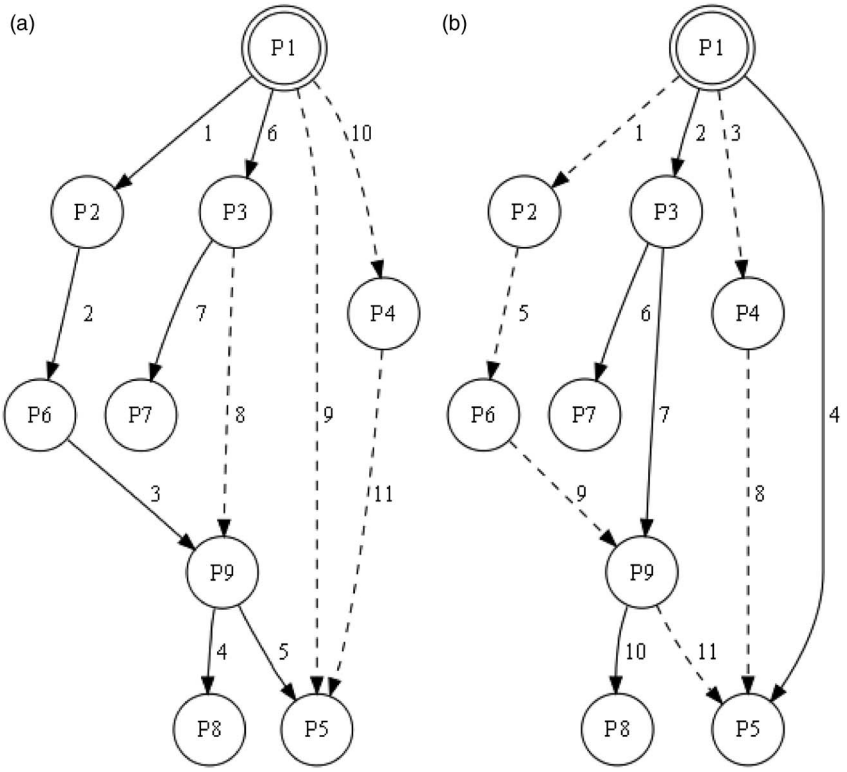

Figure 18 (a) Recommendation tree with depth-first search (DFS), (b) recommendation tree with breadth-first search (BFS)

5.8 Remarks

The simulation results show that the overlay network construction is an important issue in recommendation retrieval. The storage peer approach in PG achieves the best performance compared with other methods utilizing a structured overlay network. The structured overlay network enables a peer to reach other peers efficiently. A structured overlay network, however, assumes that all peers are capable and willing to participate in hosting the recommendations and routing messages. However, P2P networks can consist of heterogeneous nodes with varying capabilities, so nodes with less computing power may become the bottleneck in the network (Lua *et al.*, 2005).

In addition, peers are autonomous, so it is difficult to ensure the availability of the storage peers. If a peer leaves the network without properly handing over the recommendations that it stores to another peer appointed to replace its duty, the information will be lost. It can be abused by peers who want to disrupt the reputation system. Using replicas can help to avoid such problems, but it increases the complexity of recommendation storage. The replicas need to synchronize themselves in order to keep their information up-to-date and consistent. Replicas are also useful in reducing the risk of storage peers tampering or dropping the recommendations, because their response can be compared. However, the source peer would have to contact more than one storage peer, thus, increasing the communication cost. Recommendation retrieval would benefit from structured overlay networks if these problems can be addressed.

The unstructured overlay network, on the other hand, is simple and has been implemented widely in P2P environments. It enables nodes with various capabilities to participate in the network, although not providing the most efficient performance. A recommendation tree using BFS shows interesting results as an alternative to flooding. It even surpasses the retrieval performance of flooding at low ARs with lower communication cost. In a P2P network, the AR starts at 0 and increases as peers start to transact. On the other hand, the AR decreases as more new peers are joining and old peers are leaving. Hence, it is less likely for a peer to know or be known by >60% of other peers in a large and active network.

In order to benefit from the recommendation tree, however, the overlay network has to be constructed based on the recommender set. Each peer has to be connected directly to its recommenders. Otherwise, in a structured network, a peer would be able to reach its recommenders but the number of messages would increase by a factor of log n, owing to the number of hops between the peer and its recommenders. The problem is worse in an unstructured network where the peer may need to flood the network to find its recommenders as there is no routing information. In such a case, it is possible for the recommenders to be unreachable (although they are online) if they are outside the range of the TTL. In addition, the high traffic caused by the flooding would be much less utilized as only few recommenders are sought. This problem would eliminate the cost benefit of the recommendation tree over flooding. It shows the importance of overlay network construction in recommendation retrieval.

The significance of overlay network construction is also shown in the results of flooding. Both FL1 and FL2 use flooding, so their performance has similar pattern but with significantly different values. A network with small diameter as in FL2 would have better performance. Another important issue is how to traverse the overlay network. This is shown by RT2, which has significantly less number of replies compared with RT1, owing to using BFS, instead of DFS.

6 Conclusions and future work

In this paper, we developed a reputation assessment process that can be used as a classification, comparison, and analysis tool for reputation systems. A reputation system consists of many phases including recommendation retrieval (i.e. recommender selection and recommendation collection), recommender filtering, recommendation evaluation, and recommendation manipulation.

The methods in recommender selection and recommendation collection contribute to the effectiveness and efficiency of a reputation assessment process. In this paper, we analyze three different methods to compare their performances. The flooding method is simple and can reach most part of the network, resulting in high and stable coverage. However, the precision depends on the AR. In larger networks, the increase of traffic is not desirable from the scalability point of view. On the other hand, in RT, as AR increases, the precision increases while the recall decreases. If all or most recommenders are active, that is, having recent transactions with the target peers, few recommendations (i.e. low recall) may be sufficient. As the network getting larger, AR is more likely to decrease, which in turn will generate deeper recommendation trees and higher recall but it also means higher traffic. In fact, RT may use more bandwidth compared with FL for low AR and high diameter networks. PG seems to offer a very good solution to retrieve recommendations. Further research should be focused on the cost of maintaining the structure in networks with high churn rate.

Acknowledgments

The authors would like to thank all anonymous reviewers for their valuable and helpful comments, resulting in a significant improvement in the quality of this manuscript. This research is supported by King Fahd University of Petroleum and Minerals under Research Grants IN080430 and FT060028.

References

Aberer, K., Cudré-Mauroux, P., Datta, A., Despotovic, Z., Hauswirth, M., Punceva, M. & Schmidt, R. 2003. P-Grid: a self-organizing structured P2P system. *ACM SIGMOD Record* **32**(3): 29–33.

Aberer, K. & Despotovic, Z. 2001. Managing trust in a peer-2-peer information system. In *10th Int'l Conf. Information and Knowledge Management (CIKM'01)*, November, 310–317.

Aberer, K. & Despotovic, Z. 2004. *Possibilities for Managing Trust in P2P Networks*. EPFL Technical Report IC/2004/84.

Aringhieri, R., Damiani, E., Vimercati, S. D. C. D., Paraboschi, S. & Samarati, P. 2006. Fuzzy techniques for trust and reputation management in anonymous peer-to-peer systems. *Journal of the Association for Information Science and Technology* **57**(4):528–537.

Azzedin, F. & Maheswaran, M. 2003. Trust modeling for peer-to-peer based computing systems. In *International Parallel and Distributed Processing Symposium (IPDPS'03)*, April.

Azzedin, F., Maheswaran, M. & Mitra, A. 2006. Trust brokering and its use for resource matchmaking in public-resource grids. *Journal of Grid Computing* **4**(3):247–263.

Azzedin, F., Ridha, A. & Rizvi, A. 2007. Fuzzy trust for peer-to-peer systems. In *21st International Conference on Computer, Electrical, and Systems Science, and Engineering, CESSE 2007, WASET*, May, 123–127.

Baeza-Yates, R. & Ribeiro-Neto, B. 1999. *Modern Information Retrieval*, Addison-Wesley.

Bhargava, B., Lilien, L., Rosenthal, A., Winslett, M., Sloman, M., Dillon, T., Chang, E., Hussain, F., Nejdl, W., Olmedilla, D. & Kashyap, V. 2004. The pudding of trust. *Intelligent Systems* **19**(5):74–88.

Ciraci, S., Korpeoglu, I. & Ulusoy, Ö. 2005. Characterizing Gnutella network properties for peer-to-peer network simulation. In *Computer and Information Sciences – ISCIS 2005*, Lecture Notes in Computer Science **3733**, 274–283. Springer.

eBay 2013. `http://www.ebay.com`

Gupta, M., Ammar, M. H. & Ahamad, M. 2006. Trade-offs between reliability and overheads in peer-to-peer reputation tracking. *Computer Networks* **50**(4):501–522.

Hou, M., Lu, X., Zhou, X. & Zhan, C. 2005. A trust model of P2P system based on confirmation theory. *ACM SIGOPS Operating Systems Review* **39**(1):56–62.

Hughes, D., Coulson, G. & Walkerdine, J. 2006. Free riding on Gnutella revisited. *IEEE Distributed Systems Online* **6** (6):1–18.

iTrust 2008. `http://www.itrust.uoc.gr/`

Jin, Y., Gu, Z. & Ban, Z. 2007. Restraining false feedbacks in peer-to-peer reputation systems. In *International Conference on Semantic Computing, 2007. ICSC 2007*, 17–19 September, 304–312.

Jøsang, A. & Ismail, R. 2002. The beta reputation system. In *Proceedings of the 15th Bled Electronic Commerce Conference*, June.

Jsang, A., Ismail, R. & Boyd, C. 2006. A survey of trust and reputation systems for online service provision, Decision Support Systems.

Kamvar, S., Schlosser, M. & Garcia-Molina, H. 2003. The EigenTrust algorithm for reputation management in P2P networks. In *12th International World Wide Web Conference*, 640–651.

Liang, J., Kumar, R., Xi, Y. & Ross, K. 2005. Pollution in P2P file sharing systems. In *INFOCOM 2005, 24th Annual Joint Conference of the IEEE Computer and Communications Societies*, **2**, March, 1174–1185.

Lua, E. K., Crowcroft, J., Pias, M., Sharma, R. & Lim, S. 2005. A survey and comparison of peer-to-peer overlay network schemes. *IEEE Communications Surveys & Tutorials* **7**(2):72–93.

Mekouar, L., Iraqi, Y. & Boutaba, R. 2006. Peer-to-peer's most wanted: malicious peers. *Computer Networks* **50**(4):545–562.

Mengshu, H., Xianliang, L., Xu, Z. & Chuan, Z. 2005. A trust model of P2P system based on confirmation theory. *SIGOPS Operating Systems Review* **39**(1):56–62.

Patel, J., Teacy, W., Jennings, N. & Luck, M. 2005. A probabilistic trust model for handling inaccurate reputation sources. In *3rd Int'l Conference on Trust Management*, 193–209.

Qiao, Y. & Bustamante, F. 2005. Elders know best-handling churn in less structured P2P systems. In *Fifth IEEE International Conference on Peerto-Peer Computing, 2005. P2P 2005*, 31 August to 2 September, 77–86.

Quercia, D. 2009. *Trust Models for Mobile Content-Sharing Applications*. PhD thesis, University College London.

Ripeanu, M., Iamnitchi, A. & Foster, I. 2002. Mapping the Gnutella network. *IEEE Internet Computing* **6**(1):50–57.

Rodriguez, P., Tan, S. & Gkantsidis, C. 2006. On the feasibility of commercial, legal P2P content distribution. *ACM SIGCOMM Computer Communication Review* **36**(1):75–78.

Sabater, J. & Sierra, C. 2002. Reputation and social network analysis in multi-agent systems. In *1st Int'l Joint Conf. Autonomous Agents and Multiagent Systems*, 475–482.

Schollmeier, R. 2005. *Signaling and Networking in Unstructured Peer-to-Peer Networks*. PhD thesis, Technische Universität München.

Selçuk, A., Uzun, E. & Pariente, M. 2008. A reputation-based trust management system for P2P networks. *International Journal of Network Security* **6**(3):235–245.

Sen, S. & Sajja, N. 2002. Robustness of reputation-based trust: Boolean case. In *1st Int'l Joint Conf. Autonomous Agents and Multi-Agent Systems (AAMAS-02)*, July, 288–293.

Song, S., Hwang, K., Zhou, R. & Kwok, Y.-K. 2005. Trusted P2P transactions with fuzzy reputation aggregation. *IEEE Internet Computing* **9**(6):24–34.

Wang, H., Zhu, Y. & Hu, Y. 2005. To unify structured and unstructured P2P systems. In *19th IEEE International Parallel and Distributed Processing Symposium*, April.

West, A. G., Lee, I., Kannan, S. & Sokolsky, O. 2010. An evaluation framework for reputation management systems. In *Trust Modeling and Management in Digital Environments: From Social Concept to System Development*. Z. Yan (ed.), IGI Global, 282–308.

Whitby, A., Jøsang, A. & Indulska, J. 2005. Filtering out unfair ratings in Bayesian reputation systems. *The Icfain Journal of Management Research* **4**(2):48–64.

Xiong, L. & Liu, L. 2004. PeerTrust: supporting reputation-based trust for peer-to-peer electronic communities. *IEEE Transactions on Knowledge and Data Engineering* **16**(7):843–857.

Yan, Z. & Prehofer, C. 2010. Autonomic trust management for a component based software system. *IEEE Transactions on Dependable and Secure Computing*, April.