Available online at www.sciencedirect.com

ScienceDirect

Electronic Notes in Theoretical Computer Science 244 (2009) 13–25

Electronic Notes in Theoretical Computer Science

www.elsevier.com/locate/entcs

Recommendation Retrieval in Reputation Assessment for Peer-to-Peer Systems

Farag Azzedin[1] ,[2] Ahmad Ridha[1] ,[3]

Department of Information and Computer Science
King Fahd University of Petroleum and Minerals
Dhahran, Saudi Arabia

Abstract

Recommendation-based reputation assessment in peer-to-peer systems relies on recommendations in predicting the reputation of peers. In this paper, we discuss the effectiveness and cost metrics in the recommendation retrieval. We evaluate the following retrieval methods: flooding, recommendation tree, and the storage peer. The simulation results show that overlay network construction significantly contributes to the performance of recommendation retrieval in terms of effectiveness and cost. Storage peer approach in structured network outperforms the other two approaches as long as the network is stable.

Keywords: Reputation System, Recommendation Retrieval, Peer-to-Peer Systems

1 Introduction

The evolution of reputation is one of the most profound and irreversible changes in *peer-to-peer* (P2P) systems [5,17]. Peers with different motivations and malicious behavior can cause harm in such systems. This implies that mechanisms to manage interactions on the basis of trust are fundamental and are integral part of the success of any online community [3,19]. Reputation systems are widely used in the literature to assess the trustworthiness of a given peer. A reputation system retrieves, filters, and evaluates the received recommendations about the past behavior of a given peer. The derived reputation score is used to help peers decide whether a future interaction with a given peer is conceivable or not. Recommendation retrieval provides the input to the reputation system, while the filtering and evaluation components process the input.

[1] The authors wish to thank King Fahd University of Petroleum and Minerals and its Information Technology Center for the facilities utilized to perform the present work and for their support.

[2] Email:fazzedin@kfupm.edu.sa

[3] Email:aridha@kfupm.edu.sa

doi:10.1016/j.entcs.2009.07.035

In order to retrieve recommendations, recommenders have to be selected and the recommendations have to be collected from these selected recommenders. In selecting the recommenders and collecting the recommendations, the objective is to obtain as much first hand information as possible with minimum cost. Once the recommendations are received, the objective of the filtering component is to avoid undesirable recommenders. Contacting undesirable recommenders will result in waste of bandwidth and processing power as well as polluting the reputation system. Undesirable recommenders include dishonest recommenders, inactive recommenders that give stale recommendation, or unwilling recommenders that do not even reply. The filtering component feeds back the recommendation retrieval component so that the reputation system will contact only honest, active, and willing recommenders.

In the evaluation stage, the recommendations: (a) can be adjusted to improve the accuracy of the recommendations [5], (b) are assigned weights indicating the importance of the recommendation, and (c) are aggregated in order to come up with the derived reputation score. If the score vouches that the interaction with the given peer is conceivable, then the interaction can take place. Afterwards, the reputation of the given peer can be disseminated on demand or voluntarily.

In this paper, we focus on the performance issues of the recommendation retrieval component. We were motivated by the fact that the recommendation retrieval component controls the effectiveness and the cost of the information flow and consequently contributes to the accuracy of the derived reputation score. Section 2 discusses the existing recommendation retrieval mechanisms and proposes a taxonomy of recommendation retrieval. A two-dimensional evaluation process for the retrieval mechanisms, namely effectiveness and cost, is specified in Section 3. It also discusses the performance of different recommendation retrieval mechanisms. We conclude the paper in Section 4. For the rest of this paper, we will use the following terminology: (a) a peer that has the first hand information is called a witness, (b) 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, and (c) the peer that wants to assess a reputation is called a source peer, whereas the peer whose reputation is assessed is called a target peer.

2 Recommendation Retrieval Mechanisms

2.1 Taxonomy of Recommendation Retrieval

Recommendation retrieval involves two processes, namely, 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. The taxonomy is illustrated in Figure 1.

Recommender selection can be distance-based or identity-based. Distance-based approaches select the neighboring peers within 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

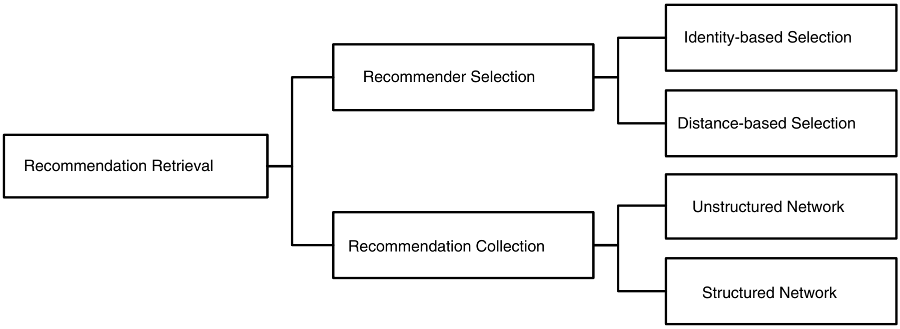


Fig. 1. Taxonomy of recommendation retrieval.

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 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.

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 overlay network also plays an important role.

2.2 Classification of Existing Reputation Systems

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 [4,11].

An identity-based approach is proposed in [5] using a recommender set. The set only changes if a recommender is found to be dishonest. A similar approach is used in [12] to search for negative feedbacks about another peer. The recommender set consists of peers that are considered trustworthy by the source peer. Another identity-based approach is to select all peers that have transacted with the target peer. This approach is used in many reputation systems [2,7,9,10,15,16,22,24] as it targets the global reputation. Social network analysis is used in [20] to select the recommenders from peers that have transacted with the target peer. This approach requires the information about the peers' connectivity, which is difficult to obtain.

In collecting the recommendations from the selected recommenders, some reputation systems do not specify the method as in [7,9]. 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 [4]. 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. Similar method is used in [11], but it builds different overlay networks for different contexts.

Recommendation tree is described in [5]. 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. It was not specified whether the mechanism is for unstructured or structured networks.

Storage peers are used in reputation systems that use all witnesses in the recommender selection. Most such systems rely on structured network infrastructure in storing the recommendations such as in [2,10,16,22,24]. The structured network enables all peers to be reachable in a bounded number of hops. Many *Distributed Hash Table* (DHT)-based networks guarantee $O(log\ n)$ hops to a destination [23].

Storage peers can also be implemented in unstructured network such as the proposed system in [9], where peers called *Reputation Computation Agents* (RCA) are used as storage peers. However, it was not specified how to select the nodes that act as the agents. Unstructured network with a hierarchy, as in *FastTrack* network, is used in [15]. A peer is connected to a *supernode* and recommenders push their recommendations to the supernode. The *supernode* acts 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. The approach assumes that the supernodes know each other. In Table 1, we summarize the existing reputation systems based on our recommendation retrieval taxonomy.

3 Performance Evaluation

3.1 Performance Metrics

We use Precision (P), Recall (R), and F-measure (F) metrics to evaluate the effectiveness of a recommendation retrieval mechanism. These metrics are defined as:

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

Table 1
Classification of Existing Reputation Systems

System	**Recommender Selection**	**Recommendation Collection**
[4,11]	Distance-based: Peers within TTL	Flooding
[5,12]	Identity-based: Recommender set	Recommendation tree
[7,9]	Identity-based: All witnesses	Unspecified
[10,16,22]	Identity-based: All witnesses	P-Grid/DHT
[15]	Identity-based: All witnesses	Supernodes

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

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

where P_W is the set of all witnesses.

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

The effectiveness metrics are adopted from information retrieval [6]. Using only recall or precision can give the wrong picture about the performance of recommendation retrieval. Perfect recall can be achieved by simply contacting all peers, but it would lower the precision. For precision, perfect performance can be obtained by contacting a single witness, but recall would be low if there are more other witnesses. Therefore, F-measure combines both metrics into a single value giving the same importance to both precision and recall. Perfect F-measure is only achievable with perfect precision and recall. We also evaluate the cost of recommendation retrieval mechanism. This cost is measured as the number of messages sent to other peers and the number of replies, i.e., the messages needed for the recommendations to reach the source peer.

3.2 Simulation Setup

The evaluation is developed as a discrete event simulation. We use 1024 and 2048 peers in constructing the overlay network and the number of transactions is five times the number of peers. For recommendation tree, the recommender set for each peer is constructed by randomly choosing four other members. In flooding, we use $TTL = 7$ based on the predominantly used value in Gnutella [8,18]. Recommendations are sent individually by back-tracking the request path. The source and the target peers for each transaction were randomly generated from [1,N].

In order to reduce duplicate requests in flooding, a peer does not forward the request to the peer from whom it is received. Also, if the same request has been received previously, it would only be forwarded if its TTL is larger than the TTL of the previous request.

We introduce the term *acquaintance rate* (AR), which is the average probability that one peer knows another. Higher acquaintance rate means a peer has had transactions with more peers. In the simulation, we assume that the acquaintance rate 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 acquaintance rate.

3.3 Overlay Network Construction

3.3.1 Overlay Network Construction for Flooding

In simulating flooding, we build the unstructured overlay network using two algorithms. **Algorithm 1** is based on the construction described in [21]. It results in multiple islands instead of forcing all peers to form a single network. For the simulation, we only take the largest island which in average consists of about 75% of the total peers. For example, the largest island in one of the generated networks using 1024 peers has 784 peers with diameter 17 and 57.82% peers are reachable with $TTL = 7$.

```
Algorithm 1 :
Step 1: Assign maximum nodal degree for each peer
Step 2: Until no peer can be connected without exceeding its max. nodal degree do:
Step 2a: Randomly select two peers
Step 2b: If both are not connected and have not reached their maximum degree then connect them to each other
Step 3: If there are peers not reaching their maximum degrees then add connections until all peers reach their maximum degree
```

Algorithm 1. Algorithm 1 for unstructured overlay network construction.

Comparing the overlay network construction using **Algorithm 1** to the observation of Gnutella network in [18], the portion of peers reachable with $TTL = 7$ for the largest island is less than 60% whereas in [18], it was more than 95%. The difference is due to the construction of the overlay network. In Gnutella network observed in [18], new peers would contact some predefined nodes 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 1** pairs peers randomly.

In order to simulate the environment observed in [18], we develop **Algorithm 2** to construct the overlay network. The maximum number of connections is based on the average connections observed in [18]. In the pairing (*Step 2* and *Step 4a*), no peer can be connected to itself and a maximum nodal degree is imposed. *Step 3* ensures that all peers have at least one connection. This algorithm connects all peers into a single island and keeps the diameter relatively small compared to **Algorithm 1**. In one of the generated networks using the second algorithm with 1024 peers, there is only one island with diameter 11 and 99.29% peers are reachable with $TTL = 7$. Compared to **Algorithm 1**, **Algorithm 2** generates an overlay network with a smaller diameter and higher reachability with $TTL = 7$.

```
Algorithm 2 :
Step 1: Select two peers randomly as the starting pair
Step 2: Place the initial pair in the connected peer list and assign equal probability to both
Step 3: For each peer do:
    Step 3a: Connect it to a peer in the connected peer list based on the probability
    Step 3b: Add the peer to the connected peer list
    Step 3c: Update the probabilities based on the nodal degree
Step 4: For each remaining connection do:
    Step 4a: Select a peer randomly, connect it to another peer based on the probability
    Step 4b: Update the probabilities based on the nodal degree
```

Algorithm 2. Algorithm 2 for unstructured overlay network construction.

Table 2
Average number of hops for P-Grid.

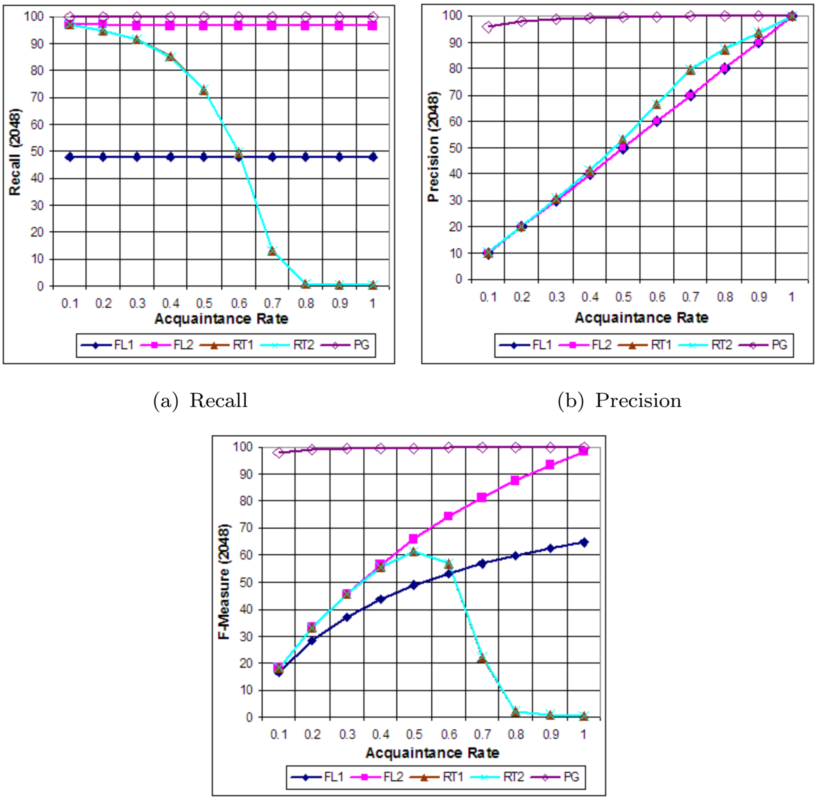
# of peers	Run 1	2	3	4	5	Avg.	$\frac{log_2(N)}{2}$
128	3.49	3.49	3.45	3.55	3.40	3.47	3.5
256	4.10	4.07	3.99	4.03	4.03	4.03	4.0
512	4.56	4.54	4.49	4.49	4.49	4.52	4.5
1024	5.01	4.97	5.02	5.03	5.01	5.01	5.0
2048	5.50	5.50	5.50	5.50	5.50	5.50	5.5

3.3.2 Overlay Network Construction for Recommendation Tree

In simulating recommendation retrieval using 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 the retrieval process 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 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 all of its recommenders.

3.3.3 Overlay Network Construction for Storage Peer

For retrieval in the structured network using storage peers, we select P-Grid [1]. We measure the average number of hops needed to reach other nodes with different network sizes and find that the average number of hops to reach another peer is $log(N)/2$ where N is the number of peers as illustrated in Table 2. In measuring the recall and the precision, we consider reaching the storage peer as equivalent to contacting all witnesses in the system.



(a) Recall

(b) Precision

(c) F-Measure

Fig. 2. Effectiveness of recommendation retrieval.

3.4 Simulation Results and Discussion

Figure 2(a) shows the recall values in flooding using **Algorithm 1** (FL1), flooding using **Algorithm 2** (FL2), recommendation tree using DFS (RT1), recommendation tree using BFS (RT2), and P-Grid (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 due to the overlay network construction algorithm. FL2 has a network diameter average of 11.4, much smaller compared to the network diameter average of 20.1 in FL1. With TTL = 7, more than 95% peers are reachable in FL2, whereas only 47.83% peers are reachable in FL1. The node reachability in FL1 and FL2 is illustrated in Figure 3 and Figure 4, respectively. Compared to FL1, more recommenders are covered in RT1 and RT2 for low acquaintance rate because recommendation tree is not limited by a TTL. It results in a higher recall than FL1. However, at high acquaintance rates, the search tree in RT1 and RT2 shrinks and recall plunges as the number of contacted witnesses decreases. The pattern is not affected by the choice of DFS or BFS. PG has recall of 100% because it uses storage peers that ensures the availabiliy of the first hand information from all witnesses.

Figure 2(b) shows that for FL1, FL2, RT1, and RT2, precision is directly related

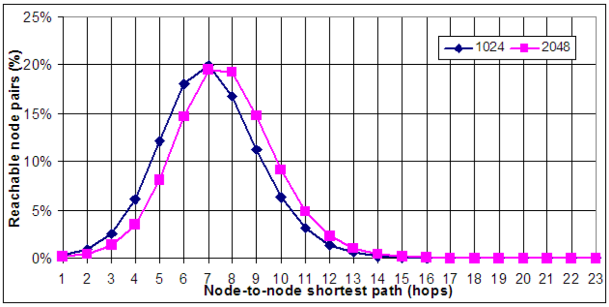


Fig. 3. Node reachability (FL1).

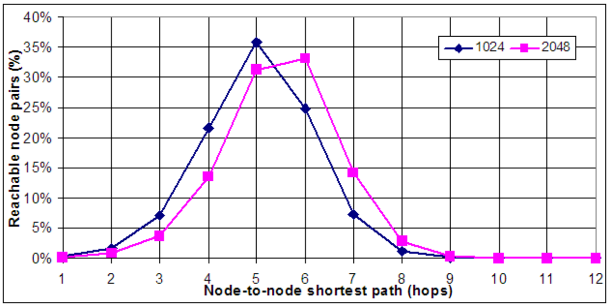


Fig. 4. Node reachability (FL2).

to the acquaintance rate 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 that flooding at high acquaintance rates because in some instances, the recommendations are found earlier, resulting in higher precision and affecting the average precision. P-Grid behaves differently from flooding and recommendation tree because it uses the storage peer approach. The precision in P-Grid is approximately the ratio between the number of witnesses and the number of hops plus the number of witnesses because individual recommendations are returned.

The F-measure pattern is influenced by the pattern of precision and recall. F-measure in FL1 and Fl2 increases if the acquaintance rate increases due to the stable recall and the increase in precision as shown in Figure 2(c). In RT1 and RT2, on the other hand, the F-measure is low for acquaintance rate > 0.6 as the recall drops.

For the communication cost, the number of sent messages, as shown in Figure 5(a), is stable in FL1 and FL2, whereas it is inversely related to the acquaintance rate in RT1 and RT2 because the tree shrinks as more peers 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 5(b). For PG, the number of sent messages would be the same as the number of hops required to reach the storage peer without any duplicates because the overlay network provides routing information.

As for the number of replies, it is directly related to the acquaintance rate in FL1

Table 3
Number of reply messages (2048 peers)

AR	FL1	FL2	RT1	RT2	PG
0	0	0	0	0	0
0.1	471.812	1319.282	53091.430	1154.378	1126.185
0.2	947.098	2634.826	94204.35	2429.634	2255.295
0.3	1419.433	3956.042	118761.729	3848.491	3376.357
0.4	1891.429	5273.086	120644.859	5357.783	4506.515
0.5	2360.072	6591.188	94376.261	6736.522	5621.09
0.6	2830.015	7912.965	44196.176	7018.634	6764.524
0.7	3303.805	9232.640	4311.552	2617.211	7881.491
0.8	3777.46	10551.583	66.776	65.411	9020.237
0.9	4262.464	11868.245	9.934	9.954	10134.696
1.0	4722.145	13186.938	4	4	11265.197

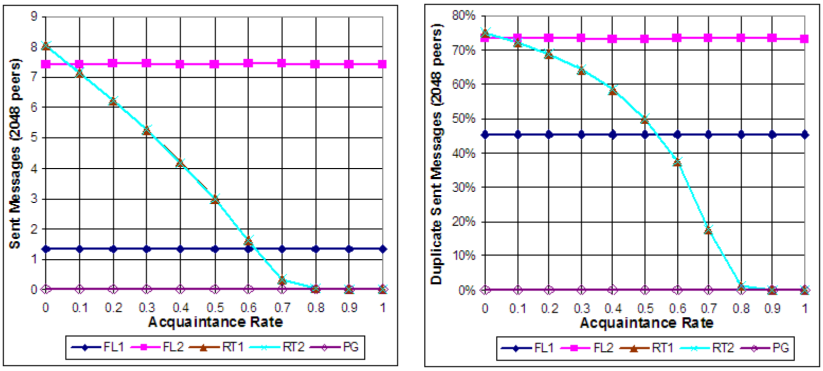


Fig. 5. (a) Sent messages (in thousands), (b) Percentage of duplicate sent messages.

and FL2. RT1 and RT2 show significantly different number of replies. For RT1, the number of replies is much higher than FL1 and FL2 except for high acquaintance rates (≥ 0.7) as shown in Table 3. This is due to the unbounded depth of the tree and the DFS approach. Due to the low acquaintance rate, the first traversed path is going very deep before finding a witness or a loop. The reply would then have to backtrack the path resulting in 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 significant reduction of replies although the number of contacted witnesses is the same as illustrated by the similar recall and precision results. Compared to FL2, the number of replies in RT2 is lower most of the time. The number of replies in PG is the number of hops times the number of witnesses which would depend on the acquaintance rate. If the storage peer has aggregated the recommendations, the number of replies would also be the number of hops.

The best retrieval performance is given by the storage peer approach utilizing a structured overlay network. However, structured overlay networks assume that all peers are capable and willing to participate in hosting the recommendations and routing messages. In reality, peer-to-peer networks can consist of heterogeneous nodes with varying capability so that nodes with less computing power may become the bottleneck in the network [14].

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

In order to benefit from recommendation tree, however, the overlay network has to be constructed based on the recommender set. Using recommendation tree in a structured overlay network without directly connecting each peer to its recommenders would increase the number of messages by a factor of *log n*. The communication cost would be worse to apply recommendation tree in a pure unstructured network without adjusting the overlay network to the recommender set. This eliminates the benefit of recommendation tree over flooding.

Our simulation also shows the significant impact of overlay network construction. Both FL1 and FL2 use flooding, so their performance has similar pattern, but with significantly different values. A network with a small diameter as in FL2 has 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 to RT1 due to using BFS, instead of DFS.

4 Conclusions and Future Work

Recommendation retrieval is an important component of a reputation system as the results of this step would be the main ingredients for recommendation-based reputation systems. This component also contributes significantly to the cost of the reputation system due to the communication cost. Therefore, the measurement of this component is critical in evaluating a reputation system.

We evaluated the effectiveness of recommendation retrieval by measuring the recall and precision. We found that the recall is 100% for structured overlay networks, whereas this is not true for unstructured overlay network. For flooding, recall is constant and depends on the construction of the overlay network. For recommendation tree, recall is inversely proportional to the acquaintance rate. On the other hand, we found that precision is highly correlated with the acquaintance rate except in structured overlay networks using storage peers, where a source peer does not need to contact all other peers in retrieving recommendations.

We also evaluated the cost of recommendation retrieval. From the simulation results, we concluded that the average number of reply massages are highly correlated with the acquaintance rate with the exception of the structured overlay network using storage peer, where the storage peer aggregates the recommendations and replies with the derived reputation score. On the other hand, the average number of sent messages is constant for structured overlay networks. For unstructured overlay networks using flooding, the number of sent messages is constant and depends on the TTL and the nodal degree. For unstructured overlay networks using recommendation tree, the number of sent messages is highly correlated with the acquaintance rate.

Recommendation retrieval in structured networks using storage peer approach can theoretically achieve the best performance in terms of effectiveness and cost because a peer can reach any other peer efficiently. However, the network has to be stable in order to reach such performance. There is also a hidden cost of maintaining the overlay network structure.

In the future work, recommendation retrieval in different overlay networks needs to be studied in detail. We have not covered unstructured network with hierarchy. Further experiments need to be conducted to incorporate churn rate especially in structured overlay networks. Also, we need to see the relationship between optimizing the overlay network for recommendation retrieval and other topology optimization approaches such as location aware topology proposed in [13].

References

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

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

[3] Anceaume, E. and A. Ravoaja, *Incentive-based robust reputation mechanism for P2P services*, Principles of Distributed Systems, Lecture Notes in Computer Science **4305** (2006), pp. 305–319.

[4] Arienghieri, R., E. Damiani, S. Vimercati, S. Paraboschi and P. Samarati, *Fuzzy techniques for trust and reputation management in anonymous peer-to-peer systems*, JASIST **57** (2006), pp. 528–537.

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

[6] Baeza-Yates, R. and B. Ribeiro-Neto, "Modern Information Retrieval," Addison-Wesley, 1999.

[7] Bhargava, B., L. Lilien, A. Rosenthal, M. Winslett, M. Sloman, T. Dillon, E. Chang, F. Hussain, W. Nejdl, D. Olmedilla and V. Kashyap, *The pudding of trust*, Intelligent Systems, IEEE **19** (Sept.-Oct. 2004), pp. 74–88.

[8] Ciraci, S., I. Korpeoglu and Ö. Ulusoy, *Characterizing Gnutella network properties for peer-to-peer network simulation*, Computer and Information Sciences - ISCIS 2005, Lecture Notes in Computer Science **3733** (2005), pp. 274–283.

[9] Gupta, M., M. Ammar and M. Ahamad, *Trade-offs between reliability and overheads in peer-to-peer reputation tracking*, Computer Networks: The International Journal of Computer and Telecommunications Networking **50** (2006), pp. 501–522.

[10] Kamvar, S., M. Schlosser and H. Garcia-Molina, *The EigenTrust algorithm for reputation management in P2P networks*, in: *12th International World Wide Web Conference*, 2003, pp. 640–651.

[11] Kinateder, M. and K. Rothermel, *Architecture and algorithms for a distributed reputation system*, in: *Trust Management: First International Conference, iTrust*, 2003.

[12] Lee, S., R. Sherwood and B. Bhattacharjee, *Cooperative peer groups in NICE*, in: *INFOCOM 2003, 22nd Annual Joint Conference of the IEEE Computer and Communications Societies*, 2003, pp. 1272–1282.

[13] Liu, Y., L. Xiao, X. Liu, L. Ni and X. Zhang, *Location awareness in unstructured peer-to-peer systems*, Parallel and Distributed Systems, IEEE Transactions on **16** (Feb. 2005), pp. 163–174.

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

[15] Mekouar, L., Y. Iraqi and R. Boutaba, *Peer-to-peerś most wanted: Malicious peers*, Computer Networks: The International Journal of Computer and Telecommunications Networking **50** (2006), pp. 545–562.

[16] Mengshu, H., L. Xianliang, Z. Xu and Z. Chuan, *A trust model of P2P system based on confirmation theory*, SIGOPS Oper. Syst. Rev. **39** (2005), pp. 56–62.

[17] Ravoaja, A. and E. Anceaume, *STORM: A secure overlay for P2P reputation management*, in: *Self-Adaptive and Self-Organizing Systems, 2007. SASO '07. First International Conference on*, 9-11 July 2007, pp. 247–256.

[18] Ripeanu, M., A. Iamnitchi and I. Foster, *Mapping the Gnutella network*, IEEE Internet Computing **6** (2002), pp. 50–57.

[19] Rodriguez, P., S. Tan and C. Gkantsidis, *On the feasibility of commercial, legal P2P content distribution*, ACM SIGCOMM Computer Communication Review **36** (2006), pp. 75–78.

[20] Sabater, J. and C. Sierra, *Reputation and social network analysis in multi-agent systems*, in: *1st Int'l Joint Conf. Autonomous agents and multiagent systems*, 2002, pp. 475–482.

[21] Schollmeier, R., “Signaling and Networking in Unstructured Peer-to-Peer Networks,” Ph.D. thesis, Technische Universität München (2005).

[22] Song, S., K. Hwang, R. Zhou and Y. Kwok, *Trusted P2P transactions with fuzzy reputation aggregation*, IEEE Internet Computing **9** (2005), pp. 24–34.

[23] Wang, H., Y. Zhu and Y. Hu, *To unify structured and unstructured P2P systems*, in: *19th IEEE International Parallel and Distributed Processing Symposium*, 2005.

[24] Xiong, L. and L. Liu, *PeerTrust: Supporting reputation-based trust for peer-to-peer electronic communities*, IEEE Trans. Knowledge & Data Engineering **16** (2004), pp. 843–857.