Using Reputation Measurement to Defend Mobile Social Networks against Malicious Feedback Ratings

Lin Huang[1], Shangguang Wang[1], Ching-Hsien Hsu[2], Juanjuan Zhang[1], Fangchun Yang[1]

[1]State Key Laboratory of Networking and Switching Technology; [2]Department of Computer Science and Information Engineering

[1]Beijing University of Posts and Telecommunications; [2]Chung Hua University

[1]Beijing 100876, China; [2]Hsinchu 707, Taiwan

E-mail: linhuang.bupt@gmail.com; chh@chu.edu.tw; {sgwang; jjzhang;fcyang}@bupt.edu.cn

Abstract

The reputation of a particular node/service is determined by the collective feedback ratings obtained from past users, and services' reputation is vital to service recommendation in mobile social networks. However, existing malicious feedback ratings complicate the accurate measurement of nodes' reputation scores. In this paper, we introduce an accurate reputation measurement approach, which uses both virgin and non-virgin reputation scores to shield services against malicious feedback ratings. We implement our approach based on the NetLogo simulation environment, and the simulation results show that our approach is capable of measuring node's reputation more effectively when suffering from malicious feedback ratings compared with other approaches.

1. Introduction

A social network is composed of a set of social actors (such as individuals, groups, organizations, or even entire societies) and the interactive relations between these actors [1]. Social networks have recently received much attention on the mobile Internet. In turn, mobile social networks have been created like Foursquare, Instagram, Path, and communities which are built around mobile functionality[2]. Mobile social network is used for individuals with similar interests to converse and connect with one another through theirs netbooks, smart phones, laptops, sensors, wireless headsets, etc. The mobile social networks provide a powerful means for users to share, organize, and locate interesting nodes with services based on the reputation of the node.

A reputation is an expectation about a node's behavior based on the information or observations of its past actions. For mobile social network, a reputation represents users' collective perception of a node. It is based on the collective feedback rating provided by users that have interacted with the node in the past[3].The feedback rating, that could be either a single value or a vector, is each user's perception of the performance of the invoked node in terms of response time, reliability, and availability. The reputation score obtained for each node, which can subsequently be determined by aggregating users' feedback ratings. Reputation as an important

metric often influences a user’s decision of selecting nodes. Hence, an accurate measurement of node reputation in mobile social networks is very important to assist users to identify nodes and connections with a proven performance such as response time, reliability, availability, etc.[4].

However, as reputation is able to influence users’ tendency to recommend nodes, some dishonest service providers (nodes) misuse mobile social networks. These service providers are intended for improving the chance of a certain node as a potential candidate for selection. Alternatively, they diminish the chances of other nodes by spreading malicious feedback ratings. Therefore, the main challenge is to address the behavior of nodes that attempt to provide malicious feedback ratings. Generally, the malicious feedback rating contains positive malicious feedback rating and negative malicious feedback rating. For positive malicious feedback rating, a service provider colludes with a group of users in giving unfairly high feedback ratings. This scenario has the effect of inflating a node's reputation. For negative malicious feedback rating, service providers can collude with nodes in giving "bad-mouth" to competitors. In such a situation, conspiring nodes provide unfairly negative feedback ratings to the targeted node, thus lower their reputation [3]. This paper mainly focuses on the malicious feedback rating.

The investigation of malicious feedback rating behavior in reputation measurement has resulted in the proposal of various notable reputation measurement schemes. Kamvar et al.[5] proposed a global trust model named EigenTrust, a file sharing system in the distributed environment. EigenTrust holds the opinion that the global reputation of a node depends on the global aggregation of its partial evaluations by all the nodes with which it has ever interacted. EigenTrust has a good inhibition effect against simple malicious attacks, collusion attacks, and attacks by traitors. However, EigenTrust only evaluates the trustworthiness of a node, which means that global iterative algorithms with a high complexity reduce the system feasibility. Li et al.[6] proposed a PeerTrust trust management model based on reputation in a Peer-to-Peer network. It considers the influence of different feedback to the trust model. The PeerTrust model reflects the context correlation of trust through transaction context factors. It reduces the effect of malicious feedback on the performance by using a feedback credibility factor. Moreover, it implements an incentive mechanism to motivate nodes to provide feedback by rating services. Ruidong et al.[7] hold the view that the emergence of selective misbehavior attacks is mainly due to the exclusive reliance of nodes on their own direct observations in the process of evaluating others. However, their own observations consider insufficiently comprehensive to enable them to evaluate others objectively.

Although the experimental evaluation described above appear to be effective for identifying malicious nodes, the network nodes have to maintain a large amount of historical transaction information during the process of similarity evaluation. Furthermore, the time complexity that is required to compute the feedback credibility is high, which places a higher demand on storage capacity and computing power of nodes.

The approach proposed in this paper attempts to calculate nodes' reputation by combining its virgin and non-virgin reputation scores, which differs from the reputation measurement models mentioned above. The combination of scores measure the reputation of a node more objectively and accurately. It effectively detect and combat the negative effect of malicious feedback ratings, result in assisting users (called deciding nodes) to make a more informed choice of selecting a node or service based on objective reputation measurement. The approach first calculates the non-virgin reputation score of target node by using recommendation data from other nodes, and

then updates the virgin reputation score of the node by using Bayesian reference. Finally, we measure the reputation score of the target node by combining the virgin and non-virgin reputation scores. We implemented our approach on the NetLogo platform[1] with extensive simulations. The results show that compared with other reputation measurement approaches, our approach can inhibit malicious feedback ratings effectively and provide an accurate reputation score of each node for users.

The remainder of the paper is organized as follows. We introduce our proposed approach, including non-virgin reputation computing, virgin reputation update, and reputation measurement, in Section 2. Section 3 proves the validity of our approach through comparative experiments between our approach and others. Finally, the conclusions are summarized in Section 4.

2 Our Reputation Measurement Approach

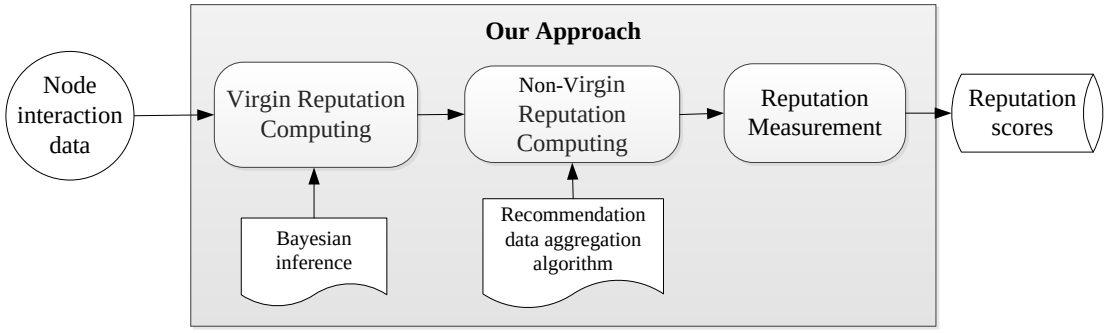

Figure 1. Flow diagram showing the different stages of our proposed approach

In existing schemes, deciding nodes exclusively rely on their own feedback rating of the target node when evaluating other nodes; however, their own feedback ratings are not sufficiently comprehensive to allow for an objective evaluation of the other nodes. Hence, it is necessary for a node to evaluate another node. The aiming is to combine its own feedback ratings for this node with those provided by other nodes while calculating the reputation of the target node. As shown in Figure 1, our approach is composed of Virgin Reputation Computing, Non-virgin Reputation Computing, and Reputation Measurement stages, and it is a process that eliminates malicious feedback ratings, and enhances the accuracy of the reputation measurement.

2.1Virgin Reputation Computing

Generally, the reputation score of a node is based on two parameters, i.e., the normal feedback rating α and the malicious feedback rating β, and it can be expressed as a vector (α, β).

Definition 1 (*Virgin Reputation*) Virgin Reputation refers to the process by which a target node uses the simple Bayesian reference to calculate the reputation score of the node. The node has had direct interactions with the target node, for which it has already generated feedback[8].

For example, two nodes, i and j, have the possibility to interact with each other, and the virgin reputation computation from node i to node j is initialized as $(\alpha = 1, \beta = 1)$. Subject node i utilizes the observed first-hand feedback ratings obtained as a result of the feedback from its

[1]https://ccl.northwestern.edu/netlogo/

last interaction with object node j to update its local evaluation of node j. Suppose the virgin reputation computation vector from i to j after *n-1* initial interactions is $(\alpha_{n-1}, \beta_{n-1})$. Thus, after the n-th interaction, this vector should be updated as follows:

$$\alpha_n = u \times \alpha_{n-1} + s \tag{1}$$

$$\beta_n = u \times \beta_{n-1} + 1 - j \tag{2}$$

where s denotes the feedback rating from i to j for the n-th provided service ($s = 1$ represents high praise and $s = 0$ otherwise), and u denotes the time discounting factor and is used to control the effect of past experience on posterior distribution.

2.2 Non-virgin Reputation Computing

Definition 2 *(Non-Virgin Reputation*) Non-Virgin Reputation refers to the process by which a target node aggregates the feedback ratings from other nodes to calculate the reputation score of the target node.

In a reputation system, selecting an approach for aggregating the feedback ratings obtained from other nodes in an effective way presents a primary problem. Most existing reputation systems search for feedback ratings based on establishing a trust chain. By broadcasting the request across the whole system, they involve agents' traversing the whole network system through connections between adjacent entities. An example illustrating the specific principle is shown in Figure 2.

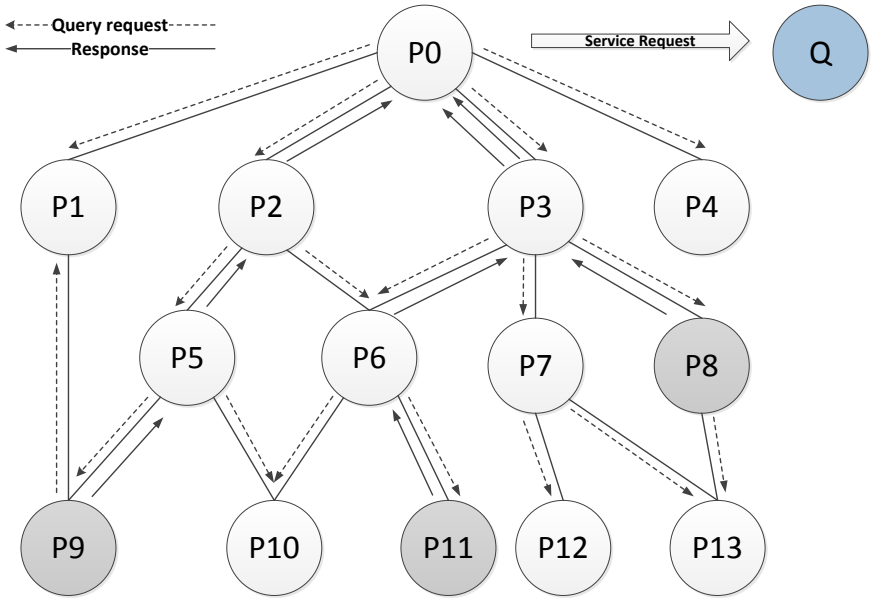

Figure 2. Example of a flooding query requesting a feedback rating from other nodes

As shown in Figure 2, if a node P0 in the mobile social network decides to send a service request to service provider Q, and then, to determine the feedback rating of the target node Q, the source node P0 sends a target node feedback rating query request packet to all its neighborhood peers. The nodes that receive this packet check whether they already have local feedback that conforms to the query request. If so, the node is respond to the request by sending a query response packet to the source node along the sending path, as shown for nodes P8, P9, and P11 in the figure. Upon receipt of the query request packet, irrespective of whether a node is able to

respond to the query, it continues to forward the query packet to its neighbors in the network until the feedback rating query depth (Query_depth) has been satisfied.

However, the source node P0 may receive the response packet several times from the same node, since the query request packet was originally forwarded in the form of a flooding request. Therefore, it is necessary to compare the feedback rating contained in each received response packet with the local feedback at the responding node. The response packet is discarded if the difference were found to be greater than the default threshold. This is similar to the model proposed by Buchegger[9], and the verification of information similarity is shown as follows:

$$|E(Beta(\alpha_i,\beta_i))-E(Beta(\alpha_d,\beta_d))|\le d \tag{3}$$

where d as a threshold value denotes the similarity between virgin reputation and non-virgin reputation and it should be setting by users. We denote with $E(Beta(\alpha,\beta))$ the expectation of the distribution $Beta(\alpha,\beta)$. We consider the non-virgin and virgin reputation computation vectors to be (α_i,β_i) and (α_d,β_d), respectively. If the condition in (3) is satisfied, the feedback rating contained in the response is accepted; otherwise, dropping. Algorithm 1 includes the detailed process of searching and aggregating relevant recommended information for a node that receives published second-hand information, and then name it as a current node in the algorithm.

Algorithm 1 Searching and aggregating relevant feedback ratings.

```
CD=Current_Query_depth
IF (CD <= Query_depth)
{
The current node flooding release query request packet for the target node;
    IF ( receive new query response packet)
    {
        Receive the packet;
        Verify and process such packet by Buchegger algorithm;
        IF ( the packet passes validation )
Update recommend information table;
        ELSE    Discard the packet;
}
ELSE
   Discard the packet;
}
```

As shown in Algorithm 1, nodes that receive the query request packet, namely, the current node will check whether there are local data conforming to the query requests. If so, the node responds to the request by sending a response packet to the source node along the sending path. The current node will continue to forward the query packets until it satisfies *Query_depth*. After receiving response packets, the current node will check whether receiving it before. If it is yes, and then drop it. Otherwise, it will verify the recommendation information by Buchegger algorithm, and processes the recommendation feedback ratings passed the verification.

Based on an aggregation of the relevant feedback ratings, we assume that node *u* has released the response to the query containing the feedback rating for node *v*. When the *Query_depth* is satisfied, there are *s* different nodes responding to the query. The feedback received by node *u* can

be ordered according to time, with the most recent row at the top, thus populate the list as follows: $\{(\alpha_1,\beta_1),(\alpha_2,\beta_2),...,(\alpha_i,\beta_i),...,(\alpha_s,\beta_s)\}$.

Then the computation of the non-virgin reputation becomes a process of combining multiple sources of evaluation parameters into a single parameter as follows:

$$\alpha=\sum_{i=1}^{s}w_i a_i \ , \quad \beta=\sum_{i=1}^{s}w_i\beta_i \tag{4}$$

where $w(0\le w_s\le w_i\ldots w_1\le 1)$ is a time-related weighting factor and the newer evaluation parameter corresponds to the larger factor. In this way, it is possible to obtain the non-virgin reputation score.

2.3 Reputation Measurement

Monitoring the responses to the requests for feedback ratings enables both the virgin and non-virgin reputations to be computed, and provides access to the latest feedback information. The virgin reputation computation vector (α_D,β_D) is integrated with the non-virgin reputation computation vector (α_I,β_I) , forming the final reputation computation vector as follows:

$$(\alpha,\beta)=\gamma(\alpha_D,\beta_D)+(1-\gamma)(\alpha_I,\beta_I) \tag{5}$$

where γ $(0\le\gamma\le 1)$ is the virgin reputation weighting factor. With respect to the observation type, first-hand information is usually weighted more heavily than second-hand information. Because the confidence level for observations made by the node itself is higher than that for observations communicated by others. It means that virgin reputation carries more weight in the comprehensive reputation score.

After updating the distribution parameters of the reputation measurement, the node *i* obtains a Beta posterior distribution (α,β) with respect to the reputation measurement of node *j*. One node evaluates another by utilizing the posterior distribution information in two respects: the reputation score t^{ij} and confidence value c^{ij} .

1) Reputation score t^{ij} . Reputation score can be calculated as the expectation value of the beta distribution[10]. Hence, the reputation score from node *i* to node *j* can be expressed as t^{ij} with the range 0 to 1 as follows:

$$t^{ij}=\mathrm{E}(Beta(\alpha,\beta))=\frac{\alpha}{\alpha+\beta} \tag{6}$$

2) Confidence value c^{ij} . Confidence value c^{ij} is used to measure the credibility of the reputation score in the range of 0 to 1 as follows:

$$\mathrm{c^{ij}}=1-\sqrt{12}\sigma\left(\mathrm{B}eta(\alpha,\beta)\right)=1-\sqrt{\frac{12\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)}} \tag{7}$$

with

$$\sigma^{\mathrm{ij}}=\sigma(\mathrm{Beta}(x,\alpha,\beta))=\sqrt{\frac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)}}$$

where the larger the value of *c*, the more credible the reputation score from node *i* to node *j*.

By combining the reputation score with the confidence value, we can obtain the final reputation score from the source node to the target node as follows[8]:

$$T^{ij} = 1 - \frac{\sqrt{\frac{(t-1)^2}{x^2} + \frac{(c-1)^2}{y^2}}}{\sqrt{\frac{1}{x^2} + \frac{1}{y^2}}} \quad (8)$$

where $0 \le T^{ij} \le 1$ and constants *x*, *y* are used to adjust the weight of the reputation and confidence in the trustworthiness, respectively.

3 Performance Evaluation

We implemented the proposed reputation measurement approach on the NetLogo platform[11] (a programmable modeling environment produced by Northwestern University). NetLogo enables the simulation of a reputation model by providing a practical application environment. It runs the model that offers features such as controlling the network scale, defining users' behavioral patterns,specifying the number of interactions between nodes and so on. We also compare our approach with other reputation measurement approaches[7,8]

3.1 Simulation Configuration

The number of entities in the network can be adjusted by changing the parameter *PeerNumber* arbitrarily within the scale limit. Two types of entities are defined in our simulated network, namely, honest entities and malicious entities. The percentage of malicious entities is denoted by *maliciousNum*. An honest entity always cooperates in transactions and subsequently provides honest feedback in terms of the provided service, whereas a malicious entity behaves fraudulently.

In the online reputation feedback system, an entity rates another entity by awarding a score of either 0 or 1 based on a binary feedback system, where a feedback of 1 corresponds to satisfaction in terms of the service, otherwise the feedback score is 0[12]. The average number of transactions for each entity is denoted by *trade_num*. Selected parameters used in our simulation are listed in Table 1.

Table 1 Simulation Parameters

Parameter	Description	Default
PeerNumber	Number of entities in the network	100
maliciousNum	Percentage of malicious entities	0.10
trade_num	Average number of transactions of an entity	100
Mrate	Percentage malicious transactions by malicious entities	100%
Query_depth	Query depth of flooding" here to improve the clarity	4.0
d_i	Weighting factor of Direct trust	0.60
Buchegger-factor	Threshold value in Buchegger algorithm	0.335
time-factor	Time discount factor	0.70

We compare our approach with two other reputation measurement approaches[7,8]. For a traditional reputation measurement approach (called TRAD)[8], an average of the feedback ratings is used to measure the reputation score of a peer without taking into account the credibility factor. Another approach called OMTF[7] mainly aims at defending selective misbehavior attacks, and

holds the view that the emergence of these attacks is mainly due to nodes exclusively relying on their own direct observations in the process of evaluating others. However, their own observations are not considered sufficiently comprehensive to objectively evaluate other nodes.

An honest node, which judged honest after the experimental process, indicates the accuracy of the assessment to the user. Similarly, the assessment is considered accurate when a malicious node is classified as malicious by the evaluation mechanism. Apart from the above-mentioned situations, the inappropriate classification of nodes indicates that the system has made an evaluation error.

We consider the reputation computation error (RCE) to be an evaluation of the accuracy of different approaches to reputation measurement, with a smaller error corresponding to improved accuracy. Within a given period, the RCE can be obtained as follows[6]:

$$RCE = \sqrt{\frac{\sum_{i \in U} [\tau_t(i) - p_t(i)]^2}{|U|}} \tag{9}$$

where |U| denotes the number of nodes in the network, $\tau_t(i)$ corresponds to the trust value of node *i* at the time *t*., and $p_t(i)$ denotes the possibility for node *i* to submit truthful feedback (i.e., $p_t(i)=1$ indicates that node *i* acts honestly at time *t*, otherwise $p_t(i)=0$).

3.3 Accuracy Comparisons on Reputation Measurement

We used an alterable number of entities, namely, *PeerNumber* for the three approaches we implemented. The parameter *PeerNumber* was initially set to 100 and each node was allowed to perform random transactions with each other, which meant that there are an average of 100 transactions for each node (*trade_num=100*). Of course, there are two kinds of behavioral patterns in the system, honest and malicious. After each transaction, the honest node submits its feedback according to its perception of the service provided. However, the malicious node carries out attacks by submitting incorrect feedback irrespective of the actual service quality experienced. Therefore, the malicious node provides different feedback ratings to different entities regardless of the actual qualities of the services the entity provided. For example, it always submits a negative feedback rating to a service provider without regard for the quality and submit a good feedback rating when benefit from it. TEC is used as the accuracy evaluation in all approaches with a varying percentage of malicious nodes (*maliciousNum*). An honest node is selected to evaluate the reputation of all the other nodes.

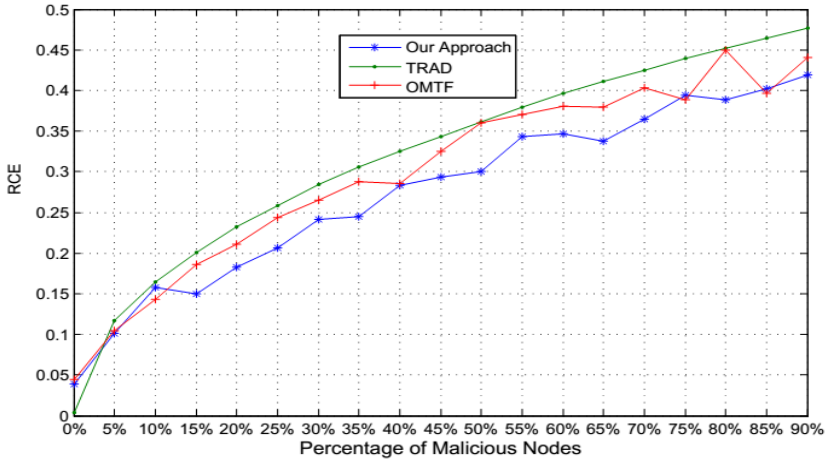

Figure 3. Accuracy of reputation measurement with respect to the percentage of malicious nodes. The experimental results showed our approach to outperform the other approaches in terms of protection against malicious feedback ratings

The results are shown in Figure 3,a plot of the accuracy of all the approaches which was determined by varying the percentage of malicious nodes (*maliciousNum*) with the malicious rate set to 1 (*mrate* = 1).

Figure 3 shows that as the number of malicious nodes increases, the REC value of TRAD and OMTF increases near linearly, and the TEC of our approach is much smaller. Compared with TRAD and OMTF, our approach includes a confidence value parameter in the reputation measurement, thus inhibit the malicious feedback rating more effectively.

Accuracy of the three approaches is compared in Figure 4, in which we vary the malicious rate (*mrate*) and set the percentage of malicious nodes at 50% (*maliciousNum* = 0.5).

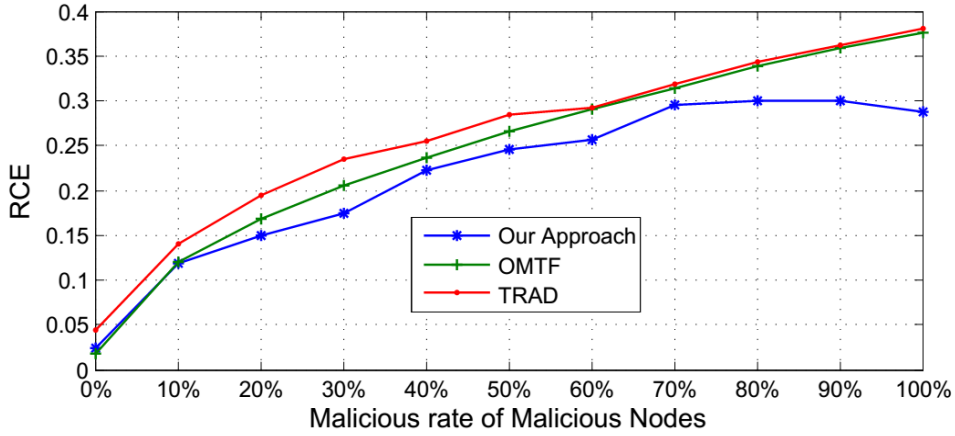

Figure 4. Accuracy of reputation measurement with respect to the malicious rate of malicious nodes. As expected, the RCE of all the approaches becomes large as *mrate* increases. However, RAD and OMTF perform worse compared to our approach, which results in lower RCE values.

As shown in Figure 4, we used *mrate* to model the possibility of a malicious node behaving maliciously. As *mrate* increases, our approach always maintain a smaller RCE value than the other two approaches.

In a word, the reputation of a node is calculated by combining its virgin and non-virgin reputation scores. Our approach effectively detects and combats the negative effect of malicious

feedback ratings, thus assist deciding nodes to make a more informed choice when selecting a node or service based on objective reputation measurement. Hence, our approach can measure the reputation of a node more objectively and accurately.

3.4 Study of Parameters

According to the working principles of our approach, there are several main parameters which affect the accuracy of our approach. Hence, in this section, we vary the value of the following three parameters: *trade_num*, *Query_depth*, and *PeerNumber*, to further analyze the corresponding performance of our approach.

3.4.1 Parameter *trade_num*

This experiment analyzed the effect of increasing the value of *trade_num* (the average number of interactions of each node) from 20 to 300. In addition, we set *PeerNumber* (the number of nodes in the network) as 100, and *maliciousnum as 0.2*.The remaining parameters were assigned the default values listed in Table 1. Figure 5shows the result of the experiment.

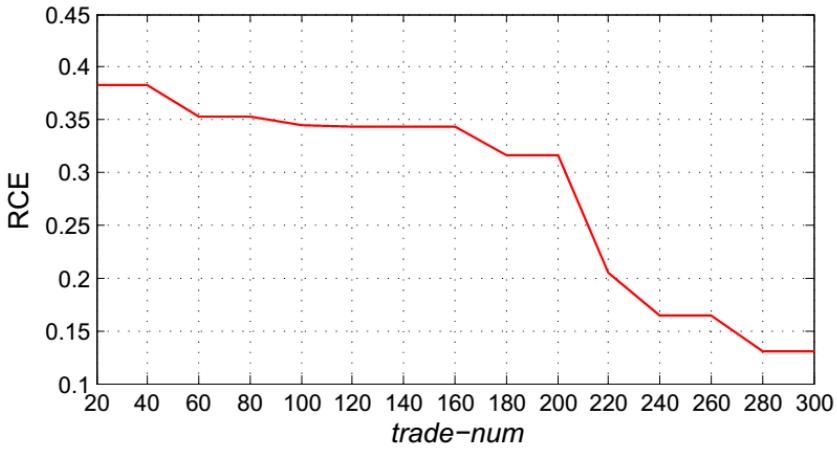

Figure 5. Experimental results obtained by varying the number of interactions per node

As shown in Figure 5, an increase in in*trade_num* leads to a decrease in RCE. Specifically, when *trade_num* is less than 200, the trust computation error is about 0.35, fluctuating within a narrow range. That is because the value of *trade_num* is not sufficiently large to lead one node to fully understand other nodes in a network where *PeerNumber* equals 100; however, once the value of *trade_num* exceeds 200, the RCE displays a significant reduction, indicating the obvious ability of our approach to withstand malicious behavior.

When a node attempts to find the optimal services, its reputation evaluations for the service providers are codetermined by the feedback ratings from both the other nodes in the network and

itself. As the value of *trade_num* increases, the source node is bound to receive a larger number of feedback responses from each individual node in the network. For the entire network, the deciding node is able to collect additional feedback ratings from other nodes during the evaluation process. Should this be the case, the RCE of our approach will decrease even further as the value of *trade_num* increases.

3.4.2 Parameter *Query_depth*

The search for feedback ratings from other nodes involves the adoption of a recursive query method in which *Query_depth* is an important factor, because it is capable of influencing the query range of the non-virgin reputation computation vector. The experiment analyzes the extent to which the performance of our approach is affected by the value of *Query_depth* (i.e., the query depth of searching by way of flooding), which was increased from 0 to 6. The value of *PeerNumber* (the number of nodes in the network) was specified as 100, and *maliciousnum* was 0.2. In the last experiment, it was determined that our approach is able to resist malicious behavior effectively when the parameter *trade_num* is set to 200; hence, this value of *trade_num* was again used in the current experiment. The remaining parameters were assigned with the default values listed in Table 1. Figure 6 shows the result of the experiment.

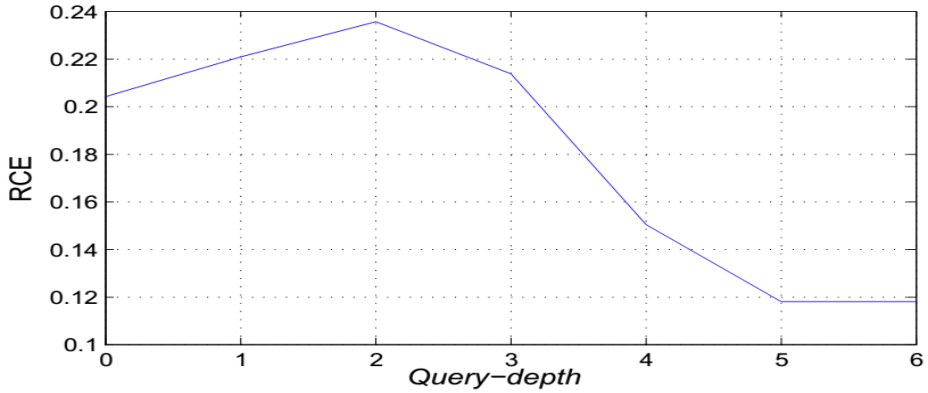

Figure 6. Experimental results obtained by varying the query depth among nodes

In Figure 6, we can see that an increase in the value of *Query_depth* initially leads to an increase in RCE. However, RCE won't increase when the *Query_depth* reaches 5, indicating that a *Query_depth* value of 5 is sufficiently large to satisfy the service-requesting node in a small-scale network with 100 nodes. According to the Six Degrees of Separation[2], it requires as many as six steps for one node to search for relevant information about anyone in the world. Hence, as the value of *Query_depth* was close to 6, the growing awareness of the service requesting node in terms of potential services providers could lead it to develop a stronger resistance to malicious feedback behavior.

[2]http://en.wikipedia.org/wiki/Six_degrees_of_separation

3.4.3 Parameter *PeerNumber*

Finally, the size of network was varied in this experiment, namely, the value of *PeerNumber* (the number of nodes in the network) which was varied from 100 to 1000 with a step of 100.This required the value of *trade_num* to be set equal to that of *PeerNumber*. The remaining parameters were assigned with the default values as listed in Table 1. Figure 7 shows the result of the experiment.

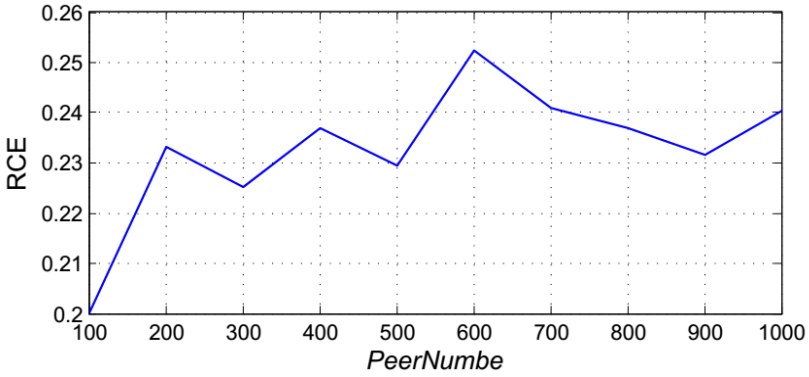

Figure 7. Experimental results on the parameter *PeerNumber*

As shown in Figure 7, increasing the network size from 100 to 1000 does not markedly affect the RCE, which retains a value of approximately 0.2 with no obvious floating. This reflects the stability of our approach under different network scale.

4 Conclusions

A service-oriented environment implementing our approach between service providers and clients are able to enhance its performance. Then the mobile social network should be exposed to malicious feedback ratings[13-15]. Compared with traditional approaches, our approach contains three phases, i.e., virgin reputation computing, non-virgin reputation computing, and reputation measurement. Extensive simulation results showed that our approach is capable of measuring the reputation of a single node effectively when the node suffers from malicious feedback ratings.

In our approach, honest nodes with a higher reputation have a clear advantage to be selected as interactive objects, and then they are more likely to be chosen for providing services, result in a significant overall improvement in the performance of mobile social networks. In future work, we will compare our approach with the famous EigenTrust approach[5] to further discuss the advantage and disadvantage of our approach.

Acknowledgements

The work presented in this study is supported by NSFC (61202435), the Natural Science Foundation of Beijing under Grant No.4132048, and NSFC (61472047).

Reference

[1] J. Yichuan and J. C. Jiang, "Understanding Social Networks From a Multiagent Perspective," *IEEE Transactions on Parallel and Distributed Systems*, no. 10, vol. 25, pp. 2743-2759, 2014.

[2] L. Eng Keong, C. Ruichuan, and C. Zhuhua, "Social Trust and Reputation in Online Social N*etworks," Proceedings of IEEE 17th International Conference on in Parallel and Distributed Systems (ICPADS 2011*), 2011, pp. 811-816.

[3] S. Wang, Z. Zheng, Z. Wu, F. Yang, and M. R. Lyu., "Reputation Measurement and Malicious Feedback Rating Prevention in Web Service Recommendation Systems," *IEEE Transactions on Services Computing*, no. 99, vol. pp. 1-14, 2014.

[4] L. JooYoung and J. C. Oh, "A model for recursive propagations of reputations in social networks," *Proceeding of 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2013)*, 2013, pp. 666-670.

[5] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina, "The Eigentrust algorithm for reputation management in P2P networks," *Proceedings of the 12th international conference on World Wide Web (WWW 2003)*, 2003, pp. 640-651.

[6] X. Li and L. Ling, "PeerTrust: supporting reputation-based trust for peer-to-peer electronic communities," *IEEE Transactions on Knowledge and Data Engineering*, no. 7, vol. 16, pp. 843-857, 2004.

[7] L. Ruidong, L. Jie, L. Peng, and C. Hsiao-Hwa, "An Objective Trust Management Framework for Mobile Ad Hoc Networks," *Proceedings of IEEE 65th Vehicular Technology Conference (VTC2007-Spring)*, 2007, pp. 56-60.

[8] P. Resnick, K. Kuwabara, R. Zeckhauser, and E. Friedman, "Reputation systems," *Communications of the ACM*, no. 12, vol. 43, pp. 45-48, 2000.

[9] S. Buchegger and J. Y. Le Boudec, "A Robust Reputation System for P2P and Mobile Ad-hoc Networks,"Proceedings of P2PEcon, 2004.

[10] C. Zouridaki, B. L. Mark, M. Hejmo, and R. K. Thomas, "A quantitative trust establishment framework for reliable data packet delivery in MANETs," *Proceedings of the 3rd ACM workshop on Security of ad hoc and sensor networks*, 2005, pp. 1-10.

[11] Z. Liang and W. Shi, "Enforcing Cooperative Resource Sharing in Untrusted P2P Computing Environments," *Mobile Networks and Applications*, no. 6, vol. 10, pp. 971-983, 2005.

[12] K. Hoffman, D. Zage, and C. Nita-Rotaru, "A survey of attack and defense techniques for reputation systems," *ACM Computing Surveys*, no. 1, vol. 42, pp. 1-31, 2009.

[13] S. Wang, Z. Zheng, Z. Wu, Q. Sun, H. Zou, and F. Yang, "Context-aware mobile service adaptation via a Co-evolution eXtended Classifier System in mobile network environments," *Mobile Information Systems*, no. 2, vol. 10, pp. 197-215, 2014.

[14] D. T. Tran, T. T. M. Truong, and T. G. Le, "A routing strategy for non-cooperation wireless multi-hop ad hoc networks," *Mobile Information Systems*, no. 4, vol. 8, pp. 333-349, 2012.

[15] J. Ahn and R. Han, "Personalized behavior pattern recognition and unusual event detection for mobile users," *Mobile Information Systems*, no. 2, vol. 9, pp. 99-122, 2013.