

FSDRep: A Trust Model for Favorable Services Discovery in Peer-to-Peer Networks

K. H. Tsai T. K. Chiu T. I. Wang
Department of Engineering Science
National Cheng Kung University, Tainan City, Taiwan
TEL: +886-6-275-7575 ext.63338
tsaikunhua@msn.com
birdstar@msn.com
wti535@mail.ncku.edu.tw

Abstract

Recently, P2P applications are becoming more and more popular. There are a variety of services available across P2P networks. However, too many similar services may result in confusing users in selecting favorable services. In addition, every peer in a P2P network is anonymous, and users have no ability to nose out whether services offered by a peer are malicious or friendly. This paper proposes a reputation model – the **FSDRep model**. First, this model describes each service from manifold reputation aspects, using which, users can find out the status of the services that he has discovered, and can also obtain completely the historic behavior of the service and peer which provides this service. The model assesses the reputation and the favorite of each service. Then, from the favorite point of view, it can consider the current status of services such as the network distances or network bandwidths. Besides, the **FSDRep model** uses a mechanism for revising each usage recommendation by taking the recommender's attitude into account and filtering out malicious services and recommenders. In such a way, through simulation, **FSDRep model** can provide users to discover not only trustable but also favorable services.

Keywords: Reputation, Favorable Services, Malicious Services, P2P Networks.

1. Introduction

P2P network is emerging as a new network application architecture in which a variety of services are scattered and shared. Using P2P technologies [8, 9, 10], peers over internet can be connected with each other to form groups called *overlay networks* or *communities*. Intrinsicly, a P2P network can aggregate a large number of services which are spread across the network and can be utilized and accessed conveniently by each of its peers. Because each peer manages and maintains their own services, the supervising of the services is easy and uncomplicated. No peer needs to manage services other than their own.

With P2P applications prevailing, the number and the types of services increase rapidly and becomes diverse in P2P networks (Although for the time being most of the P2P applications provide file-sharing services only, we optimistically believe that more and more types of services, such as entertainment service, financial service, and etc. will be appearing). While services can be located quickly by P2P applications, there is hardly any way to judge both the reliability and the trust of these services and their owner peers. Too many services available for users to select without a fair mechanism for evaluating their qualities results in confusing the users from selecting the suitable one. Hence, several important problems have to be solved before P2P systems can be used widely in the future service network (Here, wide application can be regarded as paying type services or value-added services). The first one is *how to estimate the favorite of a service*. Although a user can find out the credibility of a specific service of a specific peer by asking other peers' opinion or consulting his own record of the service, the information he get may not describe the whole current behavior and state of the service and the peer. The second problem is *how to figure out the attitude of the recommender who provides opinions*. This is a very important issue because a user may not trust an opinion unless he knows the attitude of the recommender, i.e. for a benevolent recommender, he usually returns better feedback than true representation. The third issue is *how to filter out malicious services/recommenders*. All peers are heterogeneous and anonymous in a P2P network. Malicious peers may destroy or attack other peers by releasing bad services or disseminating malicious files such as viruses and false opinions. Similarly, malicious recommenders may respond incorrect opinions. Then this will result in wrong decisions. So, there must be a mechanism established to tell good peers from bad ones.

In this paper, the **FSDRep-model** is proposed to support favorable service discovery. The model is

composite of the **Reputation model** and the **Favorite model**, and is with the intension of achieving the following goals.

- **Describe completely the Reputation and the Favorite of services and peers.**
- **Provide an adjustable recommendation mechanism.**
- **Filter out malicious services and recommenders.**
- **Discover favorable services.**

2. Related work

For the time being, some systems like eBay, Amazon, Yahoo Auction in the e-commerce domain have adopted some mechanisms to assess the reputations of both their providers and their consumers. The reputation information in these mechanisms usually consists of the evaluation of the buyer and the vendor for every single transaction. Every evaluation has just simple contents, for example, the past transaction scores that are accumulated one by one with the range in between -1 and +1 value per each transaction. Because all these systems are centralized, users can obtain all the evaluations easily, which will not be the case in decentralized systems. Other researches like [1, 2, 3, 5] also support reputation metrics to determine either peers/services are malicious or good. Each P2P system has different environment and constraints, so the specific reputation metric may not be applicable to all systems. Observing from the above referred systems [3, 4, 5, 6, 7], some issues can be addressed. They are described as follows.

- **Incorrect or malicious opinion feedbacks.** For the time being, not any reputation system can easily determine whether a returned opinion is appropriate or not. Malicious feedbacks may fabricate bad reputation for a good service/peer but outstanding reputation for a bad one.
- **Incentive of feedback.** Many reputation systems [1, 4, 5] assume that users will energetically provide their feedbacks, but in fact, only a few users are willing to return thoughtful opinions because they will not obtain any benefits. When they are unwilling but have to return feedbacks, they might just decide to return some cursory or arbitrary opinions. If a reputation system depends on such imprecise, even malicious, feedbacks to make a decision, then the users of the system will have higher probabilities to choose bad or malicious services/peers.
- **Single Transaction evaluation.** Despite the size or the significance of every transactions are different, many reputation systems regard them equally. It is easy to calculate though, but is not so reasonable. A user who downloads one music file and one movie clip should result in different degrees of evaluation for the services

and the peers, if these two services are both fulfilled successfully. The transaction volume of these two services may not be the same; Service fulfilled with higher transaction volume should get higher evaluation.

- **Pendulous service/recommender.** Some of the peers may be capricious. They may provide good services or honest recommendations at one time but malicious ones at the others. In some other situations, services or recommenders behave well in the beginning, but later, after having earned high reputation values, begin to give malicious services and to response with incorrect recommendations. When reputation values they own begin to decrease, they will resume good services and feedbacks again. Cases like this result in the confusion of estimations. Some reputation models assume that malicious peers/services will show malicious behavior in all times. But sometimes this is just not the case.
- **Tampered reputation values.** In centralized reputation systems, reputation values associated with peers and services are stored in some righteous servers which prevent these values from being tampered easily. But the risks will exist in P2P network. When a peer asks some recommendations from other peers, during the delivery of these recommendations, malicious peers may intercept and tamper these values. The oppressive consequence is that the requestor receives the faulty feedbacks about specific peers or services.

This paper focus on first four problems mentioned above. The security problem, i.e. reputation tampering problem, is not considered in the paper. We assume that the message deliveries are secure.

3. FSDRep Model

Reputation describes the long-term behavior of a service or a peer. In this paper, both two factors history (long-term status) and current state (short-term status) will be considered in order to set up a model for selecting favorable services.

3.1 Terminology definition

This subsection gives the two terminologies, the *trustable service* and the *favorable service* a clear definition. Fig. 1 shows the classifications of services.



Fig. 1 Classification of Services

- *Trustable service.* Its trustable score is estimated from two sources - the reputations of both the service and the peer that offers the service. If the score is higher than a threshold value, the low bound of the trustiness value, beyond which a user will treat this service as a trustable service and may consider using the service.
- *Favorable service.* It is not only a trustable service but also a suitable service for a user under a certain circumstance. A service, after being evaluated by the reputation model, should be estimated by the favorite model. By this favorite model evaluation, different clients could set their own risk-tolerating margins and personalized service discovery could be achieved.

3.2 Reputation model

The reputation model uses a user's own estimation and the recommendations of other users to a service to count the reputation score of the service. The formula of the *Reputation Score* of a service, $RS(p,r)$, is defined as follows:

$$RS(p,r) = \alpha \times e(p,r) + (1 - \alpha) \times R_{all}(p,r) \quad (1)$$

Where $e(p,r)$ represents a user's estimation on the service r provided by the peer p . If a user never use this service r before, $e(p,r)$ is set to zero. And $R_{all}(p,r)$ is the summation of *all* the recommendation values on the service from other users. The alpha α holds a weight value, which is defaulted to be 0.5. Some other details are described as follows:

- *Service Estimation.* Each peer should evaluate every service after using it. The estimation is expressed in one value, the $e(p,r)$. This value judges the quality of the service r provided by peer p , and its range is set to between 0 (Bad service) and 10 (Good service).

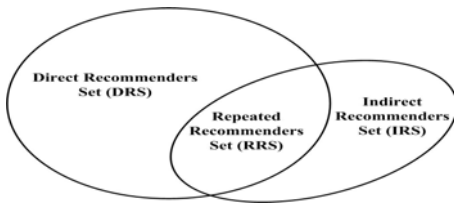


Fig. 2 Recommenders' classification

- *The summation of Recommendations.* This summation is divided into two parts. The first is the evaluation of all the *direct recommendations*, these direct feedbacks from part of the all recommenders, that, as shown in fig. 2, is represented as a set DRS. The evaluation of this part is done by using formula (2):

$$R_{all}^{\Gamma}(p,r) = \frac{1}{|DRS|} \times \sum_{i \in DRS} (e_i(p,r) - PAE(i)) \quad (2)$$

The $R_{all}^{\Gamma}(p,r)$ is the average of the recommendations returned by the peers that reply their feedbacks $e_i(p,r)$ directly on a specific service r provided by peer p . It is identified by the Γ superscription. Every peer feedback is also adjusted by a *Peer Attitude Estimation* $PAE(i)$ to be described late. When a peer asks other peers for the recommendations on a service r provided by a peer p , two sets of values may be returned. The first set of values is called the *direct recommendation set* (DRS), shown as the left oval in figure 2, and is composed of the evaluations on a service made *directly* by the peers themselves. These peers may also return some recorded evaluations that they received from other peers when they asked for the recommendations on the same service before and these recommendations on that service are later proved to be trustworthy. For this second set consists of values only passed on by these peers, it is hence called the *indirect recommendation set* (IRS), shown as the right oval in fig. 2 In the proposed model, a peer may be asked for evaluations on a same service at different time by different peers and may reply with a same value for the same service every time. The value, as described above, could be recorded by several peers. These peers may all pass the value on to a same peer that is asking for a recommendation on the service while the original issuer of the value sends the same value again to the peer. In such a case, a *direct* recommendation value has several duplicates in the IRS. All these duplicated values are collected as a RRS set, as show in fig. 2. Formula (2) uses only direct recommendation values for computation; duplicated values are excluded. The second part consists of the evaluations of all the *indirect recommendations*. The reason of using indirect recommendations is that a user can obtain more comprehensive opinions and estimations. Because a P2P network changes its state all the time, some of the peers that used a service may now have leave the P2P network when a user asks for a recommendation on the service. However, his evaluation on the service can still be used to enhance the precision of calculation. Formula (3) below shows how to calculate the value of indirect recommendations:

$$\begin{cases} R_{all}^{\theta}(p,r) = \frac{1}{|IRS - RRS|} \times \sum_{j \in IRS - RRS} R_j^{\theta}(p,r), \\ R_j^{\theta}(p,r) = \frac{1}{w} \times \sum_{k \in DRS} (e_{k,j}(p,r) + PAE(k)) \quad (3) \end{cases}$$

Formula (3) does not include the duplicated values in the RRS set in the computation of the second part of *the summation of Recommendations* because they have been used in the computation of the first part. The term

$R_{all}^{\theta}(p, r)$ is the *average* value of *all* the indirect recommendations on a service r provided by peer p . It is identified by the θ superscription. It is calculated by summing all the indirect recommendations returned from all the peers, which belongs to IRS-RRS, and dividing the summation by the total number of these peers. The *indirect* recommendation received from a peer j on a service r provided by peer p is represented as $R_j^{\theta}(p, r)$. Since peer j may send several recorded direct evaluations that he received from other peers when he asked before for the recommendations on the service r , these evaluations must be averaged and be adjusted accordingly. The term $e_{k,j}(p, r)$ represents a recorded evaluation on a service r provided by peer p that peer j received from some peer k before. Such an evaluation must be adjusted, when it is added to the summation, by the *Peer Attitude Estimation, PAE(k)* to be described later. Finally, the total recommendation value is calculated by formula (4) as shown below, where γ is a weight to balance between direct and indirect recommendation significance.

$$R_{all}(p, r) = (1 - \gamma) \times R_{all}^{\Gamma}(p, r) + \gamma \times R_{all}^{\theta}(p, r) \quad (4)$$

- *Peer Attitude Estimation (PAE)*: The purpose of PAE is to normalize a peer's attitude toward evaluating any other peers and services. In the realistic human society, while using a service, every client may have different standards for appraising the quality of the service. A user with a stricter standard might mark other services with scores lower than what they deserve. On the other hand users with a tolerant standard might give scores much higher than what the services actually perform. To counteract the effects due to different personalities, this model incorporate this *PAE* in calculating values of both direct and indirect recommendations.

3.3 The Favorite Model

Through the reputation model, a set of trustable services is established. Next in the Favorite model, there are two aspects to be taken into account. The first is the short-term behavior of a peer that provides a service and the second is the network status of the peer. These notions are described as follows.

- *Peer's Network Status (PNS)*. In a P2P network, when accessing a service, a user might come across many network problems. It is perhaps a better strategy to get know the current connecting status of every peer that provides services before deciding on which service to use. Here, PNS is further divided into two factors as follows and their weights are also shown as in table 1.

1. *Bandwidth Factor (BF)*. It measure how much the network bandwidth is when connected to a peer for the time being.
 2. *Distance Factor (DF)*. This minor factor takes into account the network distance between a specific service and the user. A simple method to measure distances is to use number of hops or the respond time between a specific peer and a service.
- *Time window (TW)*. This time window is used as the duration in which the short-term behavior of a service is observed. A peer can decide the period of watching the behavior of a service by adjusting the size of the TW, in which all the transactions will be judged and classified as favorable or unfavorable ones.

Table1: The PNS weight of the network status

| | BF : KB/sec, DF : number of hops | | |
|------------|----------------------------------|------------|---------|
| | DF<50 | 50<=DF<100 | DF>=100 |
| BF>=100 | 2 | 1 | 0.5 |
| 50<=BF<100 | 1 | 0.5 | 0.25 |
| BF<50 | 0.5 | 0.25 | 0.125 |

Using some of the factors, the favorite of each service can be calculated by formula (5), in which, TWS is the set which collects a peer's estimations and recommendations that have been adjusted by *PAE* during the TW . The R_t is a threshold value. During the TW , each pair, (estimation, recommendation), will be recorded. The value, ex is the estimation or recommendation in the TWS during TW . The favorite of a service will be high, if the value favorite_i(p, r) is high.

$$\text{favorite}(p, r) = \frac{1}{|TWS|^2} \times \sum_{ex \in TWS} ex \times \sum_{ex \in TWS} f(ex) \times PNS_p$$

, where $f(ex) = \begin{cases} 1 & , \text{ if } ex \geq R_t \\ 0 & , \text{ otherwise} \end{cases} \quad (5)$

4. Favorable Services Evaluation

After using FSDRep model to estimate these candidate services, some favorable services will be selected by the favorite scores. In this section, it describes how to update estimation and other related information for the favorable services or recommenders.

4.1 Update Estimation

After using a service, a peer should evaluate the service and its providing peer. Several processes are involved and they are described below.

- *Service estimation update*. After using a service r , the peer records the evaluation of the service for this time as $e^{new}(p, r)$. The estimation of service r is then revised by all the old ($n-1$) estimations and the new estimation.

It is the average of all the historical estimations, and is calculated by formula (6) below. The reason of using all the historical estimations in calculating a peer's estimation is that a peer's behavior should be observed over a long period instead of a single transaction.

$$e(p, r) = \frac{1}{n} \times \left(\sum_{j=1}^{j=n-1} e^j(p, r) + e^{new}(p, r) \right) \quad (6)$$

- **PAE update.** After using a service r , a peer may have an evaluation for the service different from the feedbacks of other recommenders. A peer is subjective enough to the calculation of PAE . For the long run, it is necessary for a peer to collect and revise the PAE of each known recommender. The formula of updating the PAE is shown as below in (7) where $e_k(p, r)$ represents a recommendation provided by peer k . As can be seen, the recommendation history of peer k is also used in updating its own PAE .

$$\begin{cases} PAE^{new}(k) = e^{new}(p, r) - e_k(p, r) \\ PAE(k) = \frac{1}{n} \times \left(\sum_{j=1}^{n-1} PAE^j(k) + PAE^{new}(k) \right) \end{cases} \quad (7)$$

4.2 Malicious recommenders filter

After evaluating a service, a peer can record the $PAEs$ of all the recommending peers in their associated PAE lists. The *Recommendation Standard Deviation* (RSD) of a peer can then be calculated by its PAE list recorded, which is shown as in formula (8).

$$RSD(k) = \sqrt{\frac{\sum_{j=1}^{j=n} (PAE^j(k) - \overline{PAE(k)})^2}{n}} \quad (8)$$

In the formula, $PAE(k)$ is the attitude value recorded by a peer for a recommending peer k while $\overline{PAE(k)}$ is the average of all $PAE^j(k)$ in the list. The size of the $PAE(k)$ list is assumed to be n . Each $RSD(k)$ shows the standard deviation of the recommendation values of peer k . A large value suggests that peer k has being unstable in providing recommendation values while a small value implies the other way around. Using the RSD and the PAE , a peer can easily decide whether another peer is providing valuable recommendations or not. In doing this, a peer can setup a threshold, called the rsd , to filter out those unstable peers. A peer with a RSD larger than the rsd will be classified as a malicious recommender. Such peers may be pendulous and their recommendations will be ignored. If the RSD of a peer is closed to zero, the recommending attitude of this peer can be seen as quite consistency.

In addition to the RSD and rsd , this model also defines another parameter, the *range*, to help further classifying those peers whose RSD is less than the rsd , but their $PAEs$

are slightly larger than a peer expects. One classification is shown below in Table 2.

Table2: The classification of peers

| | RSD | Range |
|----------------|----------------|----------------|
| Valuable peer | $RSD \leq rsd$ | $ PAE \leq 2$ |
| Uncertain peer | $RSD < rsd$ | $ PAE > 2$ |
| Malicious peer | $RSD > rsd$ | ----- |

This paper assumes the recommendation of peer k is valuable if $|PAE^{new}(k)| \leq 2$. So if the RSD of a peer is less than or equal to the rsd and the $|PAE|$ is less than or equal to 2, it is classified as a valuable peer, while if the $|PAE|$ of a peer is larger than 2, it is considered to be an uncertain peer. Their recommendations are considered only if a user can not get enough feedbacks for choosing favorable services. For those peers whose $RSDs$ are larger than the rsd , their recommendations are ignored.

5. Simulation Evaluation

In this paper, the Gnutella-like protocol is used to build one virtual P2P network. In this section, some experiment setting will be described first and several select strategies will be introduced and compared in different settings next. Finally, the experiment results are discussed.

5.1 Simulation Settings

- **Network Topology** - In the simulations, Gnutella-like topology is adopted because of its popularity in building P2P systems. In the simulation environment, the number of peers is set to 10000 and each similar service can be provided by 800 peers at most. Most of the settings are listed in Table 3.
- **Quality of services and users** - The quality of services uses three different types of service distributions. Each distribution has five different services and their scores are 10, 8, 6, 4 and 2, respectively. QoR1 is the best service distribution, QoR2 is for general service distribution, and QoR3 is the abominable distribution. The quality of users sets the percentage of valuable recommenders to in valuable ones. For example, QoU1 has 80% good recommenders and 20% bad ones.
- **Malicious behavior of users** - This setting represents the behavior of malicious recommenders. If this value is 100%, it means that a peer is completely malicious and it will provide the invaluable recommendations all the time. A value likes 80% means a peer has 80% probability to provide the invaluable recommendations and 20% probability to provide good recommendations and so on.

- **Rate of message responding** - In the simulation, three different rates are used. It sets how many peers reply a service request or how many recommenders reply the recommendations for a special service. R1 is set to the lowest rate 5% while R3 is set to the highest rate 40%.
- **Other Settings** - The **RS** weight is the weight of computing reputation score. For example, the RS1 (0.7, 0.3) means that the weight of self-estimation is 0.7 and the weight of recommendation is 0.3. In the **Recommendation weight**, the first value is the weight of direct-recommendation and the other is the weight of indirect-recommendation. The rests are **Thresholds**, **PAE**, **rsd** and **Favorite and PAE window** is set in the different functions.

Table3: The setting of experiment parameters

| | |
|-----------------------------------|---------------------------------|
| Number of Peers | 10000peers (service peers=800) |
| Quality of services | QoR1 (80%,5%,5%,5%,5%) |
| | QoR2 (50%,15%,15%,10%,10%) |
| | QoR3 (40%,0%,10%,25%,25%) |
| Quality of users | QoU1 (80%,20%) |
| | QoU2 (50%,50%) |
| | QoU3 (20%,80%) |
| Malicious behavior of users | MB1 (100%) |
| | MB2 (80%) |
| | MB3 (60%) |
| | MB4 (40%) |
| | MB5 (20%) |
| Rate of services or users replied | R1 (5% - Low) |
| | R2 (15% - Middle) |
| | R3 (40% - High) |
| RS weight | RS1 (0.7,0.3) |
| | RS2 (0.5,0.5) |
| Recommendation weight | REC1 (0.7,0.3) |
| | REC2 (0.5,0.5) |
| Threshold (St , Rt) | T1 (7,7) |
| | T2 (7,9) |
| PAE | YES / NO |
| rsd | 0.5, 1, 1.5 |
| Favorite and PAE window | 15 |

5.2 Simulate Scenarios and results

In the proposed reputation systems based on P2P networks, to discover a favorite service is to compare the reputation values of every similar service to decide favorable one or ones. A reputation value consists of the estimation of the peer itself and recommendations from other peers. The recommendations will become very important if the peer never ever access a service before. So how to filter out and update these recommendations is one big challenge. The experiments focus on two aspects. The first is to show how to use **FSDRep-model** to update the effects of recommendations. The second aspect is to show how to build the reputation information including services and recommenders of each peer fast in the different network environments that have various qualities of services and types of recommenders.

In Fig. 3 and 4, the experiment setting uses the qualities of services (QoR2) and rate of service replied 15%. Figure 3 shows a terrible result. In the QoU3 case, only 0.5% rate of success is obtained. It is because that too many peers provide invaluable recommendations, including malicious ones, tolerant ones, and harsh ones.

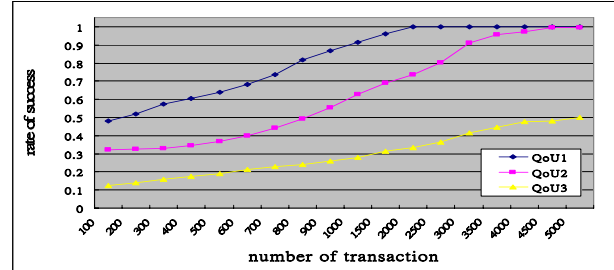


Fig.3 No update PAE function - Using the quality of service QoR2 and rate 15% to evaluate the rate of success

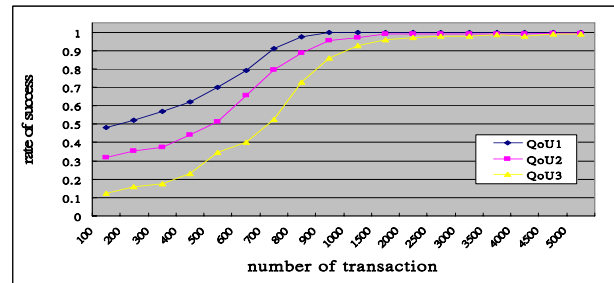


Fig.4 Update PAE function - Using the quality of service QoR2 and rate 15% to evaluate the rate of success

In Fig. 4, the *PAE* update is added to the simulation. After a user selected and accessed one service, the *PAEs* of those peers who gave recommendations about the service will be updated accordingly. As can be seen, the number of transactions in all the cases is reduced to 900 to 1500. Although case QoU3 still costs about 1500 transactions, its rate of success is raised rapidly. The other cases also cost only 900 transactions or so. Hence, the *PAE* update function is useful to normalize the attitudes of recommending peers and to make the recommendations more accurate.

In the Fig. 5 and 6, the simulation is to observe the rate of success in different malicious behavior of recommending peers. Fig. 5 shows the result of the case where QoU1 in Malicious (100%, 80%) and the results are similar to that in Fig. 4. It gets better results in the case QoU1 in Malicious 20%. The reason can be very simple, because almost all peers reply valuable recommendations. But in the case QoU1 in Malicious 40% and 60%, the change of behaviors seems affecting the service estimation and also increasing the number of transactions. In our analysis, the recommending peers change their attitude quickly because of the 40% and 60%, so the *PAE* update function does not shift their recommendations properly. To solve this problem, the recommendation filtering

mechanism (8) is incorporated, which uses a recommendation standard deviation threshold rsd to filters out those malicious peers by considering a sequence of $PAES$ of each peer. The result shows in the Fig. 6 in which QoU1 in Malicious 60% and 40%, as can be seen, obtains remarkable improvements. A new peer can build one reliable reputation table to discovery services in just a few numbers of transactions.

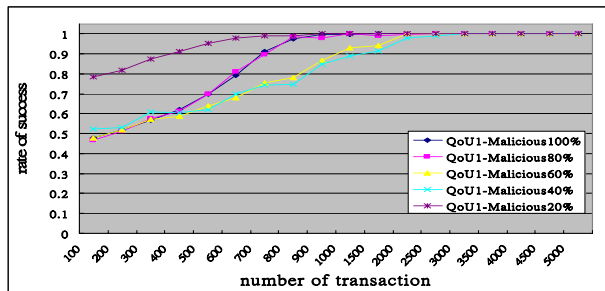


Fig.5. No filter recommender mechanism - Using the different malicious behavior of recommenders with QoU1 and rate of recommendations replied 15% in the quality of service QoR2 and rate 15% to evaluate the rate of success

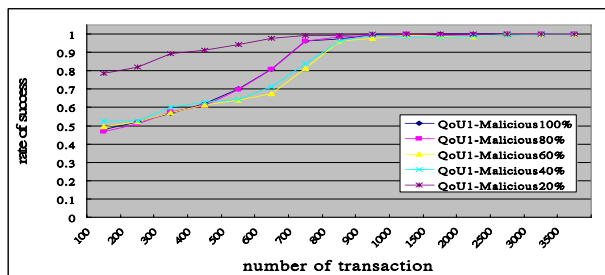


Fig.6. Filter recommender mechanism - Use different malicious behavior of recommenders with QoU1 and rate of recommendations replied 15% in the quality of service QoR2 and rate 15% to evaluate the rate of success

6. Conclusion

In this paper, the trust model –**FSDRep-model** is proposed. This model adopts many different reputation concepts. From the reputation point of view, peers can evaluate the reputation of one service by using its past history, direct recommendations and indirect recommendations. From the favorite point of view, users can obtain current network state of resources. By using these observations, a peer can select a suitable and favorable service rather than just a trustable service. As can be seen from above simulations, it is very important to eliminate malicious recommending peers. Considering these valuable feedbacks can help peers estimate and decide favorable services. And the **FSDRep-model** can also help peers build precise reputation table fast. This ensures that peers can correctly select favorite services in a very short time.

References

- [1]. Damiani, E.; De Capitani Di Vimercati, S.; Paraboschi, S.; Samarati, P.; “Managing and sharing servants' reputations in P2P systems”, Knowledge and Data Engineering, IEEE Transactions on , Volume: 15 , Issue: 4 , July-Aug. 2003 , Pages:840 – 854
- [2]. Wongrujira, K.; Hsin-ting, T.; Seneviratne, A.; “Avoidance routing to misbehaving nodes in P2P by using reputation and variance”, Advanced Communication Technology, 2004. The 6th International Conference on , Volume: 2 , 2004, Pages:1035 – 1039
- [3]. Selcuk, A.A.; Uzun, E.; Pariente, M.R.; “A reputation-based trust management system for p2p networks”, Cluster Computing and the Grid, 2004. CCGrid 2004. IEEE International Symposium on , April 19-22, 2004 , Pages:251 – 258
- [4]. Iguchi, M.; Terada, M.; Ko Fujimura; “Managing service and servant reputation in P2P networks”, System Sciences, 2004. Proceedings of the 37th Annual Hawaii International Conference on , 5-8 Jan. 2004 , Pages:200 - 208
- [5]. Wang, Y.; Vassileva, J.; ”Trust and reputation model in peer-to-peer networks”, Peer-to-Peer Computing, 2003. (P2P 2003). Proceedings. Third International Conference on , 1-3 Sept. 2003 , Pages:150 – 157
- [6]. Aameek Singh; Ling Liu; “TrustMe: anonymous management of trust relationships in decentralized P2P systems”, Peer-to-Peer Computing, 2003. (P2P 2003). Proceedings. Third International Conference on , 1-3 Sept. 2003 , Pages:142 – 149
- [7]. Hu, T.H.; Wongrujira, K.; Sereviratne, A.; “Reputation in peer-to-peer networks”, Communications, 2004 IEEE International Conference on , Volume: 3 , 20-24 June 2004, Pages:1411 - 1415 Vol.
- [8]. Stoica, I.; Morris, R.; Liben-Nowell, D.; Karger, D.R.; Kaashoek, M.F.; Dabek, F.; Balakrishnan, H . “Chord: a scalable peer-to-peer lookup protocol for Internet applications“, Networking, IEEE/ACM Transactions on , Volume: 11 Issue: 1 , Feb. 2003 Page(s): 17 -32.
- [9]. Sylvia Ratnasamy; Paul Francis; Mark Handley; Richard Karp; Scott Shenker;”A Scalable Content-Addressable Network”, Proceedings of ACM SIGCOMM 2001
- [10]. Rowstron, A.; Druschel, P.; "Pastry: Scalable, distributed object location and routing for largescale peer-to-peer systems," in Proc. 18th IFIP/ACM Int'l. Conf. Distributed Systems Platforms (Middleware), 2001.