TTEM: An Effective Trust-Based Topology Evolution Mechanism for P2P Networks

Jianli Hu[1]
Institute of Networks & Information Security, School of Computer,
National University of Defense Technology, Changsha, China
Email: lxman82@gmail.com

Quanyuan Wu, Bin Zhou
Institute of Networks & Information Security, School of Computer,
National University of Defense Technology, Changsha, China
Email: {quanyuan, binzhou}@nudt.edu.cn

***Abstract*—Current unstructured peer-to-peer (P2P) systems lack fair topology structures, and take no consideration for malicious behaviors of peers. The main reason is that the topology is not sensitive to peer's trust, and cannot accommodate heterogeneity of peers over the network. Thus, a feedback credibility based global trust model is presented in this paper. Then, based on the trust model, an adaptive topology evolution mechanism for unstructured P2P networks is proposed. Through this mechanism, trusted peers can migrate to the centric position, while untrusted peers to the edge of the topology, guaranteeing fairness during topology evolution. On the other hand, the mechanism can effectively counter the malicious behaviors of peers, and also has the incentive functionality, which incents peers to provide more high-quality services in order to get more return on services. Analysis and simulations show that, compared with the current topologies, the resulting topology mechanism demonstrates more effectiveness and robustness in combating the selfish or malicious behaviors of peers.**

***Index Terms*—P2P, topology evolution, trust, incentive mechanism**

I. Introduction

In recent years, pee-to-peer (P2P) computing has achieved its popularity in many distributed applications, including file-sharing, digital content delivery, and P2P Grid computing [1]. However, current deployed systems lack any "viable" incentive structures for encouraging users to behave in the best interest of the community. As a result, various forms of abuse and attack have been observed in practice [2]. The most common ones are free riders [3] and the attacks from some types of malicious peers. For example, some peers in P2P networks only provide inauthentic services (for example, bogus files), or use collusive strategy to attack the trust system itself. A free rider is a user who only reduces services from a P2P network while never sharing any files. These users become leeches and drain resources from the community.

Thereby, how to improve the availability of the system, compel rational and selfish participants to share their resources actively, and punish malicious peers, becomes a challenging task for the healthy development of P2P networks.1

As for the traditional resolution, some researchers have proposed some trust system for combating these problems [4-7]. In such a system, each user is assigned a trust value by the community that reflects its contribution to and its participation in the community. In contrast to building a trust system with reputations or economics that requires users to cooperate or some central authorities, we offer a trust based topology evolution mechanism to warrant the availability, effectiveness and fairness of P2P networks.

The availability of P2P networks has a close relation with P2P topology evolution, which is tightly related to the heterogeneity of peers. The heterogeneity of the peer in P2P network involves such features as the maximum connection number, computation capacity and honesty, etc. Hence, the peer's heterogeneity is largely attributed to the trust heterogeneity of this peer. Until now, in designing the topology evolution mechanism, the popular distributed P2P networks [8-10] don't take into account the factor of heterogeneity, and use the same metric for the importance of all peers, which cannot ensure the fairness of topology, and cannot prompt peers to share more authentic resources actively. Thus, referring to the concept of trust in the social network, we propose a trust-based P2P topology evolution mechanism (TTEM), which can make the normal peers take up good position, and obtain authentic services more effectively, while the selfish or malicious peers are ostracized to the fringe of the network, more difficult to gain high-quality services. Analysis and simulation experiments show that TTEM has advantages in the effectiveness of encouraging the normal peers and punishing the selfish or malicious peers

[1] 1. Corresponding author, Jianli Hu (Email: lxman82@gmail.com)

over the existing topology mechanism.

The remaining parts of the paper are organized as follows: Section II reviews the related work. Section III describes the global trust computing model. Section IV demonstrates TTEM concretely. Section V simulates and discusses TTEM. Finally, we conclude the paper make suggestions for further research work.

II. Literature Review

The related work about the adaptive P2P topology mainly includes several aspects as follow:

(1) Peer capacity-based topology evolution. Cooper BF [11] presents a topology evolution mechanism, in which peers can be self-organized into relatively efficient networks to cope with the over-loaded problems. In terms of this mechanism, the connection between peers can be divided into two categories, including the search connection (sending search messages) and the index connection (sending index messages). When subjecting to the overloaded problem, the peer drops some connections, according to the quantity of messages submitted by the neighboring peers. Similar mechanisms are also put forward by Lv Q, et al. [12] and Chawathe Y, et al. [13]. These mechanisms only conceive the factor of search connections among peers. Each peer evaluates the request-processing capacity of neighboring peers, and computes the satisfactory degree for the neighboring peers in terms of the peer's own request-processing capacity and neighboring peers' capacity. Finally, this peer establishes connection with peers with high processing capacity to improve the corresponding satisfactory degree, until the current set of neighboring peers have met the demands for request-processing capacity.

(2) Peer physical location-based topology evolution. Liu Y, et al. [14,15] give an adaptive unstructured topology evolution mechanism, which solves the marching problem between the P2P topology and the underlying physical network by choosing the nearer peer in real distance as some peer's neighbor, to improve the performance of P2P networks.

(3) Peer trust-based adaptive topology evolution. Condie T [16] provides an adaptive P2P topology (APT). The basic idea in APT is that after each transaction, the peer (supposed *i*) calculates the trust value of the corresponding peer (supposed *j*), and compares the result with its neighboring peers, to determine whether there exists some peer whose trust value is lower than that of peer *j*. If yes, peer *i* will send the connection request to peer *j*. When peer *j* receives the request, it will use the same principle to decide whether to receive this request.

Notably, the topology evolution mechanisms provided in (1) and (2) take no consideration for free-riding or malicious behaviors of peers, while APT in (3) can suppress the free-riding or malicious behaviors to a certain degree.

The differences between TTEM and APT are described as follow:

(1) As for the trust value computation, the local trust model provide by APT is a little rougher, but the feedback credibility based global trust model proposed in this paper can identify and restrain more effectively the attacks from broad malicious peers.

(2) When dealing with the process of topology evolution, APT only pays attention to the process of the trust value comparison between the current transaction peer and neighbors to decide which link to be dropped or displaced. However, in TTEM, the peer (supposed *i*) not only takes into account the current peer, but also considers the peers, which have ever transacted with peer *i*. In other words, peer *i* can pick up the peer which has the largest trust value from the whole transaction history, to make a contrast with its neighbors.

III. Peer Trust Model

A. Definition of the Model

Firstly, the definitions of the satisfactory degree valuation function and the local trust value are given. Secondly we define feedback credibility (FC), and then the global trust value (GTV).

Definition 1 Satisfactory degree valuation function. After transacting with each other, one peer *i* (the service consumer) will submit its ratings of satisfactory degree to the other peer *j* (the service provider), which can be defined as the following map function $f(i,j)$:

$$f(i,j)=\begin{cases} 1, & \textit{fully}\quad \textit{satisfactory} \\ 0, & \textit{fully}\quad \textit{unsatisfactory} \\ e(\in(0,1)), & \textit{else} \end{cases} \tag{1}$$

in which, we use the method of probability to distinguish the different QoS provided by different peers. The number 1 denotes peer *i* feels fully satisfactory to the service provided by peer *j*, while zero means for the opposite rating, and the larger the value is, the more satisfactory the peer feels.

Definition 2 Local trust value (namely, the normalized local satisfactory degree feedback). In the time fraction *t* (*t* is decided by the concrete application. For example, six months), supposing *m* denotes the number for with peer *i* has interacted with peer *j*. Thus, the local trust value peer *i* puts to peer *j* can be defined:

$$D_{ij}=\begin{cases} \dfrac{\sum_{k=1}^{m} f(i,j)}{m}, & m\neq 0 \\ 0, & m=0 \end{cases} \tag{2}$$

where, *m* = 0, means there are no transaction records between peer *i* and peer *j*. Thereby, we set 0 to the corresponding local trust value.

Definition 3 FC is used to measure the degree of accuracy of the feedback information the feedback peers (service consumers) provide to the trustee (service provider). During the trust evaluation, the feedbacks provided by the peer with higher FC are trustworthier, and are weighted more than those from the peer with the lower FC, and vice versa. Normally, FC is relevant to the following factors:

(1) The transaction frequency between peers. Normally, the higher the frequency, the higher FC is. (2) The consistency of rating behaviors between peers. The more the similarity of ratings between peer *i* and the reference peer *j* is, the more consistently they rate other peers in the network.

We introduce the transaction density factor $TNum_{ij}$ to describe the transaction frequency between peer *i* and peer *j*, which can be defined in (3).

$$TNum_{ij} = \frac{m}{n} * \beta^{\frac{1}{m}} \qquad \beta \in (0,1) \cap m \neq 0 \tag{3}$$

where, *m* denotes the transaction number between peer *i* and peer *j*, and *n* denotes the total transaction number of peer *i* with other peers. When the value of *m* equals 0, we set 0 to $TNum_{ij}$; β, being transaction density regulatory constant, is used to portray precisely the actual state of the transaction frequency, and reflect exactly the difference of the transaction density of peers.

We take the index for depicting the similarity of ratings between peers as $TSim_{ij}$, used for representing the consistency of peers' actions. Assuming the set of common interaction peers between peer *i* and peer *j* as *CSet*(*i*,*j*). Therefore, the difference of ratings between them $TDif_{ij}$ can be defined as follows:

$$TDif_{ij} = \frac{\sum_{k \in CSet(i,j)} \left| D_{ik} - D_{jk} \right|}{\left| CSet(i,j) \right|} \tag{4}$$

Supposing θ denotes the maximum deviation peer *i* can allow for peer *j*, so we can define $TSim_{ij}$ as:

$$TSim_{ij} = \begin{cases} TSim_{ij} + \frac{(1 - TSim_{ij})}{2} * \left(1 - \frac{TDif_{ij}}{\theta}\right), & TDif_{ij} < \theta \\ TSim_{ij} - \frac{TSim_{ij}}{2} * \left(1 - \frac{\theta}{TDif_{ij}}\right), & else \end{cases} \tag{5}$$

Synthesizing the above two factors, we can give the computation formula of FC (denoted Cr_{ij}) as follows:

$$Cr_{ij} = TNum_{ij} * TSim_{ij} \tag{6}$$

Thus, based on the above analysis, we can conclude that the bigger the transaction number of feedback peers is, and the more consistent the rating actions are, and so is FC.

Definition 4 We can define the feedback quality matrix as $R = (R_{ij})$, in which, $R_{ij} = D_{ij}*Cr_{ij}$. Different from the normal matrix of the network trust relationship D_{ij}, the feedback quality matrix not only allow for the peers' local trust ratings, but also consider the FC of these peers themselves. The approach of the convergence of these two kinds of information better paints the real trust level of the feedback information.

Definition 5 In the network *N*, the GTV of an arbitrary peer *i* (denoted T_i) is defined in (7).

$$T_i = \sum_{j \in K} D_{ji} * Cr_{ji} * T_j \tag{7}$$

where, *K* denotes the peer set which consists of peers, who have ever interacted with peer *i*, and offered feedbacks to it. We use the FC peer *j* puts to peer *i* and the GTV of peer *j* to weight the local trust information provided by peer *j*.

Assuming the GTV vector is $T = [T_1, T_2, \cdots, T_n]^T$, then the matrix form of (7) is as follows:

$$T = R^T * T \tag{8}$$

in which, *R* is the feedback quality matrix given in Definition 4.

B. The Convergence Analysis of GTV Iterative Computation

The iterative convergence feature of (8) determines whether we can get the computation results of the GTV vector *T*. In the following, we use the proposition that the norm of the iterative matrix R^T is less than 1, to proof the convergence of (8).

Theorem 1 For an arbitrary initial vector $T^{(0)}$, we can conclude that the simple iterative formula $T^{(k+1)} = R^T * T^{(k)}$ of (8) converges.

Proof The sufficient condition of the convergence for the above formula is the norm of the matrix R^T meets the limitations: $\left\| R^T \right\|_1 < 1$ [17].

For the computation relationship:

$$\left\| R^T \right\|_1 = \max_i \sum_j \left| D_{ij} * Cr_{ij} \right| \le \max_{i,j} \left| Cr_{ij} \right| * \max_i \sum_j D_{ij} \le \max_{i,j} \left| Cr_{ij} \right|$$

we can obtain from (6):

$$\max_{i,j} \left| Cr_{ij} \right| = \max_{i,j} \left| TNum_{ij} * TSim_{ij} \right| < \max_{i,j} \left| TNum_{ij} \right| * \max_{i,j} \left| TSim_{ij} \right| < \max_{i,j} \left| TSim_{ij} \right| < 1$$

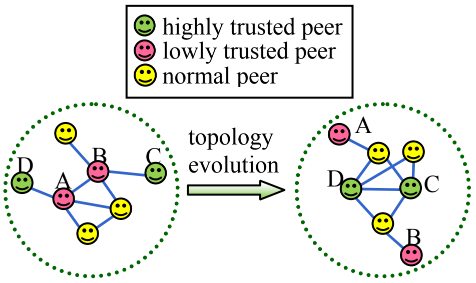
Therefore, the proposition of $\left\| R^T \right\|_1 < 1$ has been proofed.

IV. Trust Based Topology Evolution Mechanism

The peer's trust value represents the possibility of collaborating with each other in the future. Therefore, intuitively, evolving the P2P topology, in terms of the peer's trust value, can keep the connections for the collaborative peers, exclude the non-collaborative peers to the fringe of the network, and reach the objective of incentive functionality to encourage the collaborative peers and to punish the non-collaborative peers.

As illustrated in Fig. 1, the green, red and yellow small circle denote the highly trusted peer (C and D), the lowly trusted peer (A and B), and the normal peer. from the initial stage (*see* 1(a)), where peer A and peer B are lying in the center of the topology, while peer C and peer D are lying in the fringe of the topology. Through the topology evolution, we can get a possible equilibrium topology (*see* 1(b)) that the highly trusted peers move to the center, and the lowly trusted peers to the fringe.

Trust based topology evolution is implemented by the request-response mechanism between peers. A peer can locate the needed service in P2P networks by sending query requests. However, the real locating effect for this peer is largely determined by its message forwarding mechanism. Therefore, in this section, we offer a trust oriented message forwarding mechanism (TMFM) to incent peers to enhance its trust level, and reach a better



(a) Random topology (b) Equilibrium topology

Figure 1. A possible topology evolution diagram

search effect.

Before introducing the topology evolution mechanism, we first define some notations used later on.

N(*i*): The set composed of peer *i*'s neighboring peers;

M(*i*): The set composed of peers peer *i* has ever transacted with;

$TCN_{min}(i)$: The least trust threshold peer *i* can allow to its neighbors. In other words, if the trust value of a neighboring peer is less than $TCN_{min}(i)$, peer *i* will disconnect with it;

$TCR_{min}(i)$: The least trust threshold other peers have to possess, when their connection requests are accepted by peer *i*; if the trust value of the requesting peer is less than $TCR_{min}(i)$, peer *i* will refuse it;

$Fv(i)=\{j\mid T_j\geq TCR_{\min}(i),\ j\in M(i)-N(i)\}$: The set of peers peer *i* wishes to, but has not built connections with;

$SNS(i)=\{j\mid T_j\geq TCN_{\min}(i),\ j\in N(i)\}$: The set of peers whose trust value is no less than $TCN_{min}(i)$;

$Fv(i)_{\max}=\{j\mid j\in Fv(i),\forall k\in Fv(i)\ \ and\ \ k\neq j\ \ T_j>T_k\}$: The peer whose trust value is the maximum in *Fv*(*i*);

$N(i)_{\min}=\{j\mid j\in N(i),\forall k\in N(i)\ \ and\ \ k\neq j\ \ T_k>T_j\}$: The peer whose trust value is the minimum in *N*(*i*);

$\tau^i_{\min}$: The minimum connection number peer *i* can maintain;

$\tau^i_{\max}$: The maximum connection number peer *i* can maintain.

A. Message Forwarding Mechanism

Query messages are propagated via flooding-based broadcast. A search query can be initiated by any node in the network by first broadcasting to all peers in its neighbor set. Each node, upon receiving a propagated search query, will examine its local service system for a match. Any matches are returned directly to the query initiator. The peer may then propagate the query to all its neighbors except for the node from which it received the query. Each query maintains a time-to-live (TTL) field to limit the scope of the query flooding. At query time, the issuing peer will set the TTL field to some default value, which is then decremented by one at each propagation. A node receiving a query with TTL=0 will not forward the query.

Fig. 2 illustrates how a query propagates through the P2P network. In the example, peer *k* initiates the search query, with TTL=1, by broadcasting to all of its neighbors, in this case peer *j*. Peer *j* decides not to respond to the query and forwards the query to its neighbors, except to peer *k*. Upon receiving the query, peer *i* directly responds to peer *k* with a match. The query flooding terminates at peer *i* since the TTL is now 0.

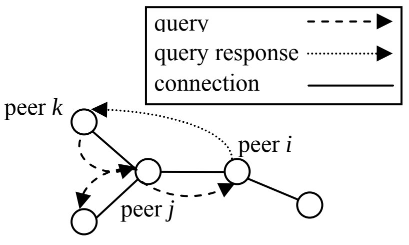


Figure 2. Query propagation model

To incent the peer to provide more authentic services actively, and increase its trust level, we relate the propagation scope of the peer with its trust value. The peers with different trust values have different propagation effect. In other words, the peer with higher trust value will be given a larger search scope, and can go across many hops to reach more peers. Thus, these peers can have more options of services offered by different providers, and vice versa. As for the lowly trusted peers, in order to gain more high-quality services, they have to improve the quality of services and feedbacks they provide to others, to increase their own trust values.

Concretely, the design idea of TMFM is as follows:

(1) The probability that a peer forwards the query request of the highly trusted peer is always higher than that does for the lowly trusted peer.

(2) The query request messages sent by a peer are largely forwarded by the peer whose trust value is less than itself.

In which, the design principle of (1) can assure that the highly trusted peer can get a larger search scope, and (2) can not only provide enough chance for the lowly trusted peer to provide good services, but also restrict the opportunity for the lowly trusted peers to gain the services from the highly trusted peer. This mechanism can prompt the lowly trusted peer to offer more contributions to others to improve its trust rank.

In the light of the above design principle, taking advantage of the design method of the trust-critical broadcast search mechanism (TBS) [2], we give a novel forwarding probability algorithm as followed:

$$Pf_{ij}=\begin{cases}T_j^r, & T_j\leq T_i\\ T_j^{1-\frac{1}{r}}, & T_j>T_i\end{cases}\tag{9}$$

in which, *r* denotes the current broadcast radius, and T_i, T_j denote the trust value of peer *i* and peer *j*, respectively. We can get the fact from (9), that for an arbitrary *r*, the formula $T_j^r<T_j^{1-\frac{1}{r}}$ is always correct. In other words, as to a peer *j*, the probability for it to obtain services from the highly trusted peer is always more than that does from the lowly trusted peer. The conclusion has proved our above design idea in TMFM.

B. Sending Connection Request

Peer *i* can update the topology links every some time, and when the time is up, it will process as the following steps:

Firstly, when peer *i* meets the condition of $Fv(i)\neq\varnothing\ \cap\ |N(i)|<\tau^i_{\max}$, it will send connection request to $Fv(i)_{\max}$, which shows the fact that peer *i* wishes to establish a desirable connection with $Fv(i)_{\max}$, but the final result is determined by the negotiation process with peer $Fv(i)_{\max}$. As to peer $Fv(i)_{\max}$, it will decide whether to accept this request or not. When the negotiation is successful, the state of *i.negotiation*($Fv(i)_{\max}$) returns the boolean value *true*, showing both participants hope to establish the connection. After building the connection, peer *i* will remove peer $Fv(i)_{\max}$ from *Fv*(*i*).

Secondly, when $Fv(i)\neq\varnothing\ \cap\ |N(i)|=\tau^i_{\max}$, the process is same with the above way, and the unique difference is that peer *i* will drop the connection with peer $N(i)_{min}$, when its current connection number has reached the specified threshold.

Finally, when $Fv(i)=\varnothing \cap |SNS(i)|<\tau^i_{\min}$, peer *i* will submit request to a random peer over the network. If the request is accepted, and the condition $N(i)>\tau^i_{\max}$ is satisfied, peer *i* will drop the connection with peer $N(i)_{min}$. Peer *i* will try to keep the connections with the $\tau^i_{\min}$ neighbors, whose trust values are no less than $N(i)_{min}$. However, when peer *i* has tried for many times without any response, it will give up.

As the connection demander, the topology evolution algorithm for peer *i* is described as follow:

```
Procedure connect_demander(i) {
    if ( $Fv(i)\neq\varnothing \cap |N(i)|<\tau^i_{\max}$ )
        if (i.negotiation(Fv(i)_max)=true)
        {
            addConncetion(Fv(i)_max);
            remove Fv(i)_max from Fv(i);
        }
    if ( $Fv(i)\neq\varnothing \cap |N(i)|=\tau^i_{\max}$ )
        if (i.negotiation(Fv(i)_max)=true)
        {
            addConncetion(Fv(i)_max);
            remove(N(i)_min);
            remove Fv(i)_max from Fv(i);
        }
    if ( $Fv(i)=\varnothing \cap |SNS(i)|<\tau^i_{\min}$ ) {
        negotiate with a random peer, and establish
        connection with it;
        if ( $N(i)>\tau^i_{\max}$ )
            remove(N(i)_min);
    }
}
```

C. Receiving Connection Request

As the connection receiver peer *j*, when it meets the condition of $T_i>TCR_{\min}(j) \cap |N(i)|<\tau^j_{\max}$ or $T_i>TCR_{\min}(j) \cap |N(i)|=\tau^j_{\max}$, it will negotiate with peer *i* to determine whether to accept peer *i*'s request or not. The following process is similar to that in Section *B*, and the corresponding algorithm is given as follow:

```
Procedure connect_receiver(i) {
    if ( $T_i>TCR_{\min}(j) \cap |N(i)|<\tau^j_{\max}$ )
        if (j.negotiation(i)=true)
            addConncetion(i);
    if ( $T_i>TCR_{\min}(j) \cap |N(i)|=\tau^j_{\max}$ )
        if (j.negotiation(i)=true)
        {
            addConncetion(i);
            remove(N(j)_min);
        }
}
```

In order to fulfill the above operations, peer *i* need maintain two tables as follow: one is the neighboring peer list (*see* 3(a)) and the other is the transaction record list (*see* 3(b)), as shown in Fig. 3.

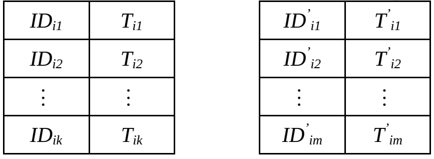
In Fig. 3, $ID_{i1},\cdots,ID_{ik}$ and $ID'_{i1},\cdots,ID'_{im}$ are the identifier sequences of peer *i*'s neighboring peers and peers who have ever transacted with peer *i*, respectively; $T_{i1},\cdots,T_{ik}$ and $T'_{i1},\cdots,T'_{im}$ the corresponding trust value sequences of these peers, respectively.

Assuming the average maximum connection number of peer *i* is *k*. We can know from the above topology evolution algorithm, that, under the extreme circumstances, the communication overhead maintained by this algorithm is $O(k)$. Due to the limited connection quantity with peer *i*, the real communication overhead is not very high. In terms of the data structure of peer *i*, we can easily know its storage overhead is $O(k\times(p+q))+O(m\times(p+q))\approx O(k)+O(m)$, in which, *p* denotes the byte number used for storing the peer's identifier, *q* does for the peer's trust value, and *m* represents the number of transaction record entry, stored in peer *i*. In addition, we can get some storage space by deleting the outdated information and the transaction records of peers with low trust value periodically.

V. Experiments And Analysis

A. Simulation Setup

The peers we simulate are categorized into three types, including (1) the normal peer, marked with capital *N*, which always provide authentic services and honest recommendations to others, (2) the free-riding peer, marked with capital *F*, which doesn't offer any service and recommendation to others, and (3) the malicious peer, marked with capital *V*, which always gives low-quality or fraudulent services and dishonest recommendations to others. In the simulation experiments, the proportion of these three types of peers is 13:5:2, respectively. The related simulation parameter settings are given in Table I. The hardware platform of simulation consists of CPU for AMD Athlon™ 64 X2 Dual 1.9GHZ, and the memory of 1GMB, and the simulation software is developed in Java.



ID_{i1}	T_{i1}
ID_{i2}	T_{i2}
⋮	⋮
ID_{ik}	T_{ik}

(a)

ID'_{i1}	T'_{i1}
ID'_{i2}	T'_{i2}
⋮	⋮
ID'_{im}	T'_{im}

(b)

Figure 3. The data structure of peer *i*

TABLE I. Simulation Parameters Settings

N	# of the total number of peers in community	1000
$\tau_{\min}$	# of the minimum neighboring connection number	3
$\tau_{\max}$	# of the maximum neighboring connection number	8
TCN_{min}	#of the least trust threshold to disconnect from its neighboring peers	0.3
TCR_{min}	#of the least trust threshold to be accepted for other peers' connect requests	0.5
β	% of the transaction density regulatory constant	0.5
θ	# of the maximum deviation between peers	0.1
P_{res}	% of the probability in response to query requests	1

B. Effectiveness against Free-Riding Peers

This experiment is aimed to verify whether TTEM can effectively banish the *F* peers to the brim of the network or not. After being banished to the brim, *F* peers have a further topology distance to *N* peers, and it is difficult for them to obtain the services provided by *N* peers. We have a comparison simulation for TTEM and APT. Here, we apply the shortest path length (SPL) as the metric to measure the location change during the process of P2P topology evolution. As to an arbitrary peer *i*, we can make use of the following formula to compute the SPL from peer *i* to the rest peers:

$$spl_i = \frac{1}{|N \setminus i|} \sum_{j \in N \setminus i} ShortestPath(i, j) \tag{10}$$

in which, *N* denotes the set of *N* peers, and $|N \setminus i|$ denotes the set of the remaining peers except peer *i*.

As demonstrated in Fig. 4, for TTEM, after 40 simulation cycles, the average SPL from *F* peers to *N* peers nears a large value (as for APT, more *F* peers have disconnected from the network in TTEM); the average SPL between *N* peers is close to a constant, roughly 2.8. However, to APT, after the same cycle, the corresponding values are 4.83 and 4.2, respectively. Obviously, APT cannot effectively drive *F* peers to the brim of the network. The main reason is that APT only takes advantage of the current local feedbacks to evolve the P2P topology, and neglect the real trust level in the whole transaction history, which results in the situation that *F* peers cannot be clearly recognized and distinguished, and difficult to be excluded to the brim of the network. Furthermore, The average SPL between *N* peers in APT is longer than that in TTEM, which proves TTEM can make *N* peers get closer together to form a cluster, to gain a higher search efficiency than APT does.

C. Effectiveness against Malicious Peers

This simulation is used to test whether the topology evolution mechanism can effectively banish *V* peers to the brim of the network. Excluding *V* peers to the brim can decrease the negative effect of malicious peers. Due to the long distance between the peers lying in the edge and those in the center, the service requests from *N* peers are difficult to reach *V* peers by setting a less TTL value. We simulate TTEM and APT at the same time, and use the same SPL formula as (10).

As shown in Fig. 5, so far as TTEM is concerned, after 55 simulation cycles, the average SPL from *V* peers to *N* peers approaches a large value. This result shows TTEM can banish *V* peers to the brim of the network. In addition, the average SPL between *N* peers in TTEM is around 2.6. Thus, we set a less value (for example, less than 3) to TTL, preventing *N* peers from receiving the service lookups. In APT, after the same cycle, the corresponding values are 5.2 and 4, respectively. Thereby, APT cannot effectively exclude *V* peers to the brim of the network. Similar to the results in Section *B*, the average SPL between *N* peers in APT is much more than that in TTEM, leading to lower search efficiency and higher communication overhead in APT.

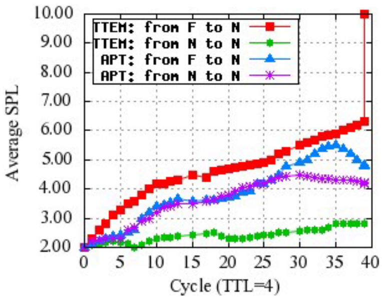


Figure 4. Comparison of average SPLs of both F peers and N peers

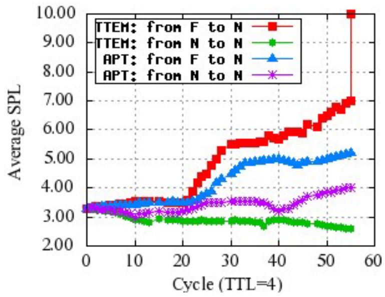


Figure 5. Comparison of average SPLs of both V peers and N peers

D. Effectiveness for Trust Based Search Mechanism

Assuming the P2P network is composed of a scale of peers, based on which, the topology evolution is executed. The trust values of peers are modeled by random distribution, whose scope are confined to (0,1). To simplify the simulation process, we choose such three trust value regions, as (0.4,0.5], (0.6,0.7] and (0.9,1.0], named S_1, S_2 and S_3, respectively. Here, we give the concept of search effect.

Definition 6 Supposing the peer set peer *i* has travelled across when it sends a search request through TMFM is B_i. For an arbitrary peer *j* ($\in B_i$), assuming the distance between peer *j* and peer *i* is d_j, we define $V_j = \frac{T_j}{d_j}$ as the search merit peer *j* put to peer *i*.

Definition 7 As for B_i defined in Definition 6, we define $E_j = \sum_{j \in B_i} V_j$ as the search effect of peer *i* at this time.

Definition 8 With respect to peer set S_i, we define $R_i = \frac{\sum_{j \in B_i} V_j}{|S_i|}$ as the average search effect (ASE) of peer S_i.

The varying tendency of ASE of the three types of peers with different trust value regions with topology evolution is shown in Fig. 6. With topology evolution, peers with similar trust values get together to form a cluster. TMFM has a stronger effect on peers with higher trust values in the search effect. Additionally, an implicit fact is that, due to the immoderately search mechanism in classic unstructured P2P applications, the topology imposes no limitations and restrictions on the highly trusted peers and the lowly trusted peers, resulting in severe tragedy of commons. On the contrary, in terms of the topology evolution mechanism in TTEM, with the lowly trusted peers diverting away from the highly trusted peers, and the effect of TMFM, the probability that the lowly

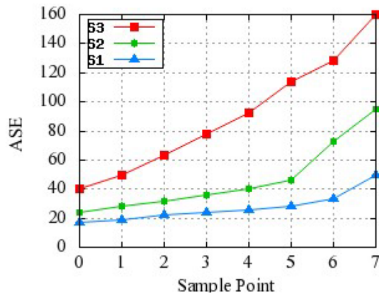


Figure 6. The varying tendency of ASE for three types of peers with topology evolution

trusted peer reach the highly trusted peer is becoming smaller gradually. Thus, TTEM can restrict tragedy of commons at the cost of decreasing the search effect of lowly trusted peers.

E. Effectiveness for Incenting Well-Behaved Peers

The index of the network effectiveness for *N* peers is used to describe how the collaborative peers obtain trusted resources. Here, we utilize the effective response rate (ERR) to measure the network effectiveness.

ERR: Supposing V_{good}^{r} represents the set of the collaborative peers who initiate search requests during some cycle, and receive the corresponding responses. As for the request submitted by peer *i*, r_i responses are received, r_i^a of which are provided by the collaborative peers. Thus, we can define ERR as follow:

$ERR_i = {r_i^a}/{r_i}$. Thus, we can get: $ERR = \sum_{i \in V_{good}^{r}} {ERR_i}/{|V_{good}^{r}|}$.

In Fig. 7, we compare the ERR for TTEM with that for APT, when the proportion between *F* peers and *N* peers is 1:4. Since APT utilizes the concept of the connection trust, the ERR for APT increases fast at the beginning. However, with the simulation cycles growing, the *V* peers in TTEM are identified rapidly to be banished to the brim of the network. Therefore, the ERR for TTEM exceeds that for APT after the 110th cycle, and reach 1 after the 290th cycle, while the ERR for APT is only about 0.84, after 800 simulation cycles.

We can see from Fig. 7, that TTEM show more effectiveness than APT. Since TTEM can converge the well-behaved peers with similar interests into a cluster more effectively, these peers in the cluster can gain high ERR more easily. However, APT regulates P2P topology only based on the current transaction records of neighboring peers, and cannot reflect the real trust level of peers. Thus, whether the well-behaved peers or the malicious peers cannot be accurately evaluated and identified, which causes a negative effect on the topology evolution mechanism.

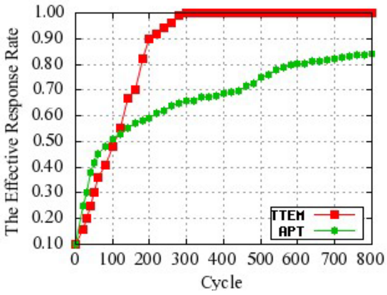


Figure 7. Comparison of the ERRs

VI. Conclusions And Future Work

In this paper, we provide a FC-based global trust model. Based on this, a trust-based adaptive topology evolution mechanism for P2P networks is given. TTEM can regulate P2P topology in light of the peer's global trust level, ensuring the fairness of the topology evolution. The mechanism can suppress the malicious behaviors of peers effectively, and also has the incentive effect on all peers. Our preliminary experiments show that TTEM show more effectiveness and robustness in incentive effect than the existing topology evolution mechanism.

However, we have not discussed the communication overhead. This index is one of the key factors for the successful deployment and effective implementation in the real engineering environment for TTEM. In our future works, we will present concrete experimental simulation and theoretical analysis for the communication overhead of TTEM with topology evaluation, and make tests in real local applications. To this end, we should make further improvements for TTEM in the mechanism performance.

Acknowledgment

The authors acknowledge the useful comments from the anonymous reviewers. This research was supported by the National Grand Fundamental Research Program (973 Program) of China under Grand No. 2005CB321800, the National High Technology Research and Development Program (863 Program) of China under Grant No. 2007AA010301, and the National Science Foundation for Distinguished Young Scholars of China under Grant No. 60625203.

References

[1] Bawa M, Cooper BF, Crespo A, Daswani N, Ganesan P, Garcia-Molina H, Kamvar S, Marti S, Schlosser M, Sun Q, Vinograd P, Yang B. *Peer-to-Peer research at Stanford. ACM SIGMOD Record*, 2003,32(3):23−28.

[2] Dou W. *The research on trust-aware P2P topologies and constructing technologies* [Ph.D. Thesis]. Changsha: National University of Defense Technology, 2003 (in Chinese with English abstract).

[3] E. Adar and B. Huberman. "Free riding on gnutella". First Monday, 5(10), October 2000.

[4] F. Cornelli, E. Damiani, S. D. C. di Vimercati, S. Paraboschi, and P. Samarati. "Choosing reputable servents in a p2p network". In *11th International WWW Conference*, 2002.

[5] R. Dingledine, M. J. Freedman, and D. Molnar. "The free haven project: Distributed anonymous storage service". In Workshop on *Design Issues in Anonymity and Unobservability*, 2000.

[6] Xiong L, Liu L. PeerTrust: "Supporting reputation-based trust in peer-to-peer communities". *IEEE Transactions on Data and Knowledge Engineering*, Special Issue on Peer-to-Peer Based Data Management, 2004, 16(7): 843-857.

[7] Kamwar S. D, Schlosser M. T, Hector Garcia-Molina. "The eigenTrust algorithm for reputation management in P2P networks". In : Proceedings of *the 12th International*

Conference on World Wide Web, Budapest, Hungary, 2003 , 640-651.

[8] Stoica I, Morris R, Karger D, Kaashoek MF, Balakrishnan H. “Chord: A scalable peer-to-peer lookup service for Internet applications”. Technical Rept :TR-819, MIT, 2001.3

[9] S. Ratnasamy et al. “A Scalable Content-Addressable Network”. Proceeding of *ACM SIGCOMM*,ACM Press, NewYork, 2001.8, pp.161-172.

[10] A.Rowstron ,P.Druschel, “Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems”. *IFIP/ACM International Conference on Distributed Systems Platforms*, Kluwer Academic Press, 2001.11, pp.329-350.

[11] Cooper BF, Garcia-Molina H. “Ad hoc, self-supervising peer-to-peer search networks”. Technical Report, Stanford University, 2003.

[12] Lv Q, Ratsnasamy S, Shenker S. “Can heterogeneity make Gnutella scalable?” In: Druschel P, Kaashoek M F, Rowstron AIT, eds. Proc. of *the 1st Int’l Workshop on P2P Systems. Berlin*: Springer-Verlag, 2002. 94−103.

[13] Chawathe Y, Ratnasamy S, Breslau L, Shenker S. “Making Gnutella-like P2P systems scalable”. In: Crowcroft J, Wetherall D, eds. Proc. of *the Conf. on Applications, Technologies, Architectures, and Protocols for Computer Communications*. New York: ACM Press, 2003. 407−418.

[14] Liu Y, Zhuang Zh, Xiao L, Ni LM. “AOTO: Adaptive overlay topology optimization in unstructured P2P systems”. In: Proc. of *the IEEE GLOBECOM 2003*. San Francisco, 2003.

[15] Xiao L, Liu Y, Ni LM. “Improving unstructured peer-to-peer systems by adaptive connection establishment”. *IEEE Trans. on Computers*, 2005, 54(9):1091−1103.

[16] Condie T, Kamvar SD, Garcia-Molina H. “Adaptive peer-to-peer topologies”. In: Lambrix P, Duma C, eds. Proc. of *the 4th Int’l Conf. on Peer-to-Peer Computing*. New York: IEEE Press, 2004. 53−62.

[17] J.Altman. “PKI Security for JXTA Overlay Networks”. Sun Microsystem, Palo Alto, Tech Rept: TR-I2-03-06,2003.

Jianli Hu was born in Wuhan, Hubei, China, in January, 9th, 1976. He received his B. E. degree in computer science in 1999, from the Mechanical Engineering College, Shijiazhuang, China. In 2003, he received his M. E. degree in military commend from Military Command Institute, Zhangjiakou, China, and in 2006, he received his Ph.D. degree in computer science from the Mechanical Engineering College, Shijiazhuang, China.

He has attended many project research, many of which were in part supported by the National Grand Fundamental Research Program (973 Program) of China and National High Technology Research and Development Program (863 Program) of China. He currently concentrates on P2P and trust research in the Computer School of the National University of Defense Technology, Changsha, Hunan, China, as a postdoctoral researcher. He has published many articles in domestic and overseas core journals or academic conferences, three of which are including: Jianli Hu et al. “FCTrust: a robust and efficient feedback credibility-based distributed trust model for p2p networks”. In: Proceedings of *the 2008 International Symposium on Trusted Computing*. Zhangjiajie, China, November 18-21, 2008. Jianli Hu et al. “Distributed and effective reputation mechanism in p2p systems”. In: Proceedings of *the 2008 International Conference on Computer Science and Software Engineering*. Wuhan, China, December 12-14, 2008. Jianli Hu et al. “RBTrust: a recommendation belief based distributed trust management model for p2p networks”. In: Proceedings of *the 2008 International Workshop on Massive Network Storage Systems and Technologies*. Dalian, China, September, 25-17, 2008. His current research interests include mobile agent computing, P2P computing, grid computing, electronic commerce, and network security.

Dr. Hu is a member of CCF. He was awarded the Mechanical Engineering College Special Research Prize in Applied Science in 2006 as the highest standing graduate in the faculty of Applied Science.

Bin Zhou is an associate professor in the School of Computer of the National University of Defense Technology (NUDT), Changsha, Hunan, China. He received his BS, MS and Ph.D. degrees in computer science from NUDT, in 1994, in 1997, and in 2000, respectively. His interests are in distributed computing technology and data mining technology.

Quanyuan Wu, as a doctoral supervisor, is a professor in computer science in the School of Computer of the National University of Defense Technology. His research interests are in artificial intelligence, distributed computing, middleware application.