An Incentive Scheme Based on Contribution for Peer-to-Peer File Sharing Systems

Jinfa YAO, Teng WANG, Baoqun YIN

***Abstract*— In peer-to-peer (P2P) file sharing systems, the behaviour of free riders and malicious peers causes negative impact on the robustness and availability of the networks. In this paper, we propose an incentive scheme based on contribution for P2P file sharing systems to motivate cooperation among peers. Our framework consists of a contribution management for completely distributed P2P file sharing systems, service policy based on contribution and server selection policy. Performance evaluations confirm the ability of our proposed scheme to effectively identify both free riders and malicious peers, and punish them by reducing the service provided to them. On the contrary, those good contributors are rewarded better service. Simulation results also show that based on rational behaviour, peers are motivated to increase their contribution in order to receive better service.**

***Index Term*— File sharing systems, Free riders, Incentive scheme, P2P.**

I. INTRODUCTION

In peer-to-peer (P2P) file sharing systems, peers communicate directly with each other to share files and exchange information. Peers of the same system provide files to the other peers, while obtaining files that they desire from the community; in other words, each peer acts at the same time as server and client. The effectiveness of existing P2P file sharing systems relies on the cooperation of users and the contributions of their file resources. However, although cooperation is of utmost importance to P2P file sharing systems, the open and anonymous nature of P2P systems open the door to misuse by malicious peers, i.e., provide unsatisfied files, and abuse by free riders, i.e., consume resource without contributing to the community. Several studies have shown that most of users in P2P systems are free riders. In Gnutella, for example, a report in 2005 indicated that 85% of users are free riders [1]. This is also known as a social phenomenon reported as "the tragedy of the common" [2] that most of the users in the system are reluctant to cooperate and only a small number of them are willing to share their resources.

Obviously, the tension between the maximization of individual utility and global optimality calls for a mechanism to encourage cooperation among autonomous nodes. Various incentive schemes have been used in current literatures to motivate cooperation, which can be roughly divided as micro -payment-based schemes [3] and no-monetary reciprocity ones. The latter includes direct reciprocal-based models and reputation-based models [4-7]. In addition, the characteristic that peers in P2P systems are treated as rational, strategic players inspired the application of game theory in modelling the interaction of peers [8-10]. Generally, the micro-payment model takes the virtual currency, which seed nodes charge for services they provided from a central server, as an intermediary to measure the contributions of nodes during various resource transactions. While the monetary schemes provide the clean economic models, they seem highly impractical because they require an accounting infrastructure to track the transactions of peers. The core idea of the incentive model of direct reciprocity is that each node can gain the same return in real time after they provided resource or service to others. Real-time is the most important feature of this kind of model. That is to say, historical information of each node involved in a transaction is only exist in one session. Thus, the application of this model has met with restriction. The concept of grade is introduced by reputation -based models to classify services provided for each node according to their credit value. However, differential service schemes require large communication overheads to determine and announce the ratings of peers. As for game theoretical method, the assumption, which it relies on, that all peers are aware of some system information, like the link capacity of all peers is unrealistic and harsh. In short, although the existing work solves, to some degree, some key problems that confront today's P2P systems, the use of simpler and more practical methods is necessitated to motivate users to cooperate. In this paper, we mainly focus on the reputation-based incentive mechanisms for completely distributed P2P file sharing systems.

II. RELATED WORKS

Trust and reputation systems have been extensively investigated in P2P file sharing systems to enhance the performances of existing systems. Current literatures propose several solutions for trust management and reputation computation. Due to space constraints, we only discuss prior art that is more germane to our work.

Ersin et. al., [5] proposed a reputation-based distributed trust architecture for P2P networks to identify malicious peers and to prevent the spread of malicious content. The protocol makes use of two kinds of different rating system (credibility rating system and trust rating system) to prevent the system from coordinated attacks. However, the authors fail to give more discussion on how to choose the values of the thresholds defined in the paper. In addition, the protocol neither distinguishes between malicious peers and careless peers who

Jinfa YAO, Student, Department of Automation, University of Science and Technology of China, Hefei, China

Teng WANG, Student, Department of Automation, University of Science and Technology of China, Hefei, China

Baoqun YIN, Professor, Department of Automation, University of Science and Technology of China, Hefei, China

spread malicious content, nor provides mechanisms to encourage those peers who have built sufficient reputation to continue taking an active part in file sharing.

The model presented in [7], called History-based Reputation System (HRS) achieves better effectiveness in restraining the phenomenon of free-riding by monitoring the file sharing behaviour of peers in the system and providing the proper preferential treatment for peers on the basis of the observed behaviour. Furthermore, HRS augments the local reputation score into a global-like reputation score without the need to disseminate scores between peers, which improves detection and control of free-riders without the risk of the Sybil and whitewashing attacks. However, at the trust evaluation process, every peer is only simply classified as either trustworthy (altruistic behaviour) or untrustworthy (free-riding). Moreover, maliciousness of peers is not taken into consideration.

Ref. [11] developed a reputation system adaptable to dynamics and robustness through taking the age of transaction into consideration. To select service provider more effectively, they introduces the concept of similarity between the requester and the recommender. Although the system could be able to identify and exclude malicious peers in some degree, the mechanism seems weak and insufficient in coping with these problems. Besides, the proposed model lacks punishment and incentive mechanisms to motive malicious peers to participate in the cooperation.

The authors in [12] analyzed the basic characteristics of some typical reputation models and proposed a possible scenario of integration of several existing reputation management techniques and routing mechanisms for individual reputation evaluation and global trust value calculation. However, the reliability of reputation feedback remains controversial because of its privacy. Moreover, some assumptions the paper based on seem unrealistic, e.i., the score manager is responsible for passing all of its stored trust values to its neighbour peer when leaving the network.

A contribution-based service differentiation [17] at super -node level, along with the reputation and the credibility schemes, has been proposed to provide the right incentives for peers to achieve better cooperation in partially decentralized P2P systems. Performance evaluations confirm the ability of the proposed contribution scheme to effectively identify both free riders and malicious peers and reduce significantly the milking phenomenon. However, the incentive mechanisms were not suitable for completely decentralized P2P systems directly.

In short, a number of existing reputation-based systems [5, 7, 14] were proposed to build trust by using peer reputation values as selection criteria to distinguish malicious peers. Nevertheless, these mechanisms lack of incentives for peers to strive for higher reputation as they don't provide differentiated service to peers with different behaviour. Obviously, this is necessary in order for the right incentives regarding performance for service provision to be provided to peers and for fairness reasons.

In addition, some other reputation-based P2P systems consider the peer's reputation as a guideline for service differentiation and reputation is computed on the basis of the number of satisfied and/or unsatisfied transaction [11, 12, 15]. That is to say that a peer with a high reputation will receive better service than peers with a lower reputation. However, these schemes can't effectively identify free riders and punish them, for free riders can obtain a high reputation by only uploading few authentic files and then take advantage of the system resources as the good participating peers.

In this paper, we enhance the application of the protocol proposed in [17] to completely decentralized P2P file sharing systems. As mentioned above, the P2P network is almost made up of self-interested (rational) peers. Thus, the service policy we presented aims to overcome peers' selfish (rational) behaviours and achieve the optimal balance between self-interest and the good of the whole community. In the light of the lack of central management of super-node which is the essential characteristic of completely decentralized systems, we introduce a proper server selection policy to achieve better efficiency of resource transactions for traffic overhead and load balance reasons.

Our main contributions to the literature are summarized as follows. Firstly, a contribution management scheme for completely decentralized P2P file sharing systems is introduced to dynamically reflect the contribution behaviour of peers. Secondly, a service policy based on contribution is proposed to provide service differentiation to peers in order to motivate cooperation among peers. Furthermore, the proposed service policy help peers of the same P2P systems create a competitive environment that will push peers to continuously upload authentic files. Thirdly, we propose a contribution -level-based server selection policy which can balance request loads among peers and help peers being in the process of building contribution to build their contribution quickly.

The rest of the paper is organized as follows. Section 3 describes the system model. Section 4 presents the detailed contribution management scheme. Section 5 describes the service policy based on contribution and Section 6 presents the server selection policy. The rational behaviour of peers is introduced in Section 7. Section 8 presents the implantation of the contribution management for completely distributed P2P file sharing systems. In Section 9, we give the performance evaluations of the proposed scheme. We draw conclusion in Section 10.

III. System Model

In this paper, we consider the completely decentralized file sharing systems, in which the files are segmented into chunks of the same size and the size of a chunk is SizeF. We consider that our system progresses in periods of a fixed number of time units, called service period. At the beginning of each service period, peers in the system decide the number of file requests they will send to other peers in current period and where to send their file requests according to our contribution -level-based server selection policy. In each file request, peer reports the chunk it wants. For those peers having received file requests from the system, they decide how to serve received file requests according to our proposed contribution -based service policy. Considering the dynamic nature of peer behaviour, every new period, peers redirect their file requests to the same or other possible servers. In this paper, content discovery is out of our discussion, but we rather consider that

chunks can be found by any other than the requesters in the P2P file sharing systems.

In our P2P file sharing system, contribution data that are needed to describe the contribution behaviour of peers and to compute contribution value of peers are stored in a distributed manner. To keep the safety and accuracy of the contribution data, our scheme will store the contribution data of peer P_i at third party peer P_h determined by $h = hash(i)$ with hash being a function known to all peers. For example, we can use a distributed hash table mechanism such as Chord [16] to maintain the contribution data in a scalable manner. That is, the contribution data of peer P_i is stored at peer P_h. Details about the implementation of contribution management scheme will be given in Section 8. The contribution data of peer P_i include:

1) Satisfied downloads of peer P_i from other peers, denoted as $D_{i,*}^{+}$.
2) Unsatisfied downloads of peer P_i from other peers, denoted as $D_{i,*}^{-}$.
3) Satisfied uploads of peer P_i to other peers, denoted as $D_{*,i}^{+}$.
4) Unsatisfied uploads of peer P_i to other peers, denoted as $D_{*,i}^{-}$.
5) Feedback credibility of peer P_i until service period q, denoted as FB_i^q.
6) Service credibility of peer P_i until service period q, denoted as AB_i^q.
7) Feedback of P_i about the quality of chunk C uploaded by peer P_j denoted as $A_{i,j}^{C}$.

After downloading chunk C from peer P_j, peer P_i will evaluate the quality of chunk C received from peer P_j. If the quality of the chunk is satisfied, then $A_{i,j}^{C} = 1$, otherwise $A_{i,j}^{C} = -1$ which represents that the received chunk does not correspond to the requested one or the quality of received chunk is not acceptable.

IV. Peer Behavior Description

In a peer-to-peer file sharing system, peers are expected to practice good peer-to-peer behaviour. Peers are implicitly trusted that they will share good quality files, that they will upload requested files, and that they will send honest feedback. Unfortunately, real life peer-to-peer systems have proved that a mechanism is needed to identity the behaviour of peers in order to guarantee the performance of the file sharing systems.

According to [17], the behaviour of peers can be described by the following three dimensions:

1) *Authentic Behaviour* which describes the reliability of a peer in providing accurate and good quality files. It helps to differentiate good peers and malicious peers.
2) *Feedback Behaviour* which represents the sincerity of a peer in providing honest feedback about the quality of received files.
3) *Contribution Behaviour* which describes how much a peer contributes to other peers compared to how much it obtains from the P2P system. Just as we mentioned above, we'll consider contribution value as a guideline for service differentiation. In order to identify both malicious peers and free riders, we differentiate satisfied uploads and unsatisfied uploads when computing the contribution value of peers.

A. Feedback Behavior

In this section, we introduce the method of computing feedback credibility of peers. Suppose the current service period is q. At the beginning of service period q, peer P_i sends TM_i^q file requests to other peers according to our server selection policy. At the end of service period q, P_i receives AC_i^q chunks according to our proposed service policy, and the lth chunk peer P_i received is uploaded by peer $P_{server(l)}$. After receiving a chunk from $P_{server(l)}$, peer P_i reports the received chunk and provides feedback $A_{i,server(l)}^{l}$ to peer $P_{hash(i)}$.

To detect peers that provide dishonest feedbacks, we adopt a method similar to [16] by introducing the concept of suspicious transaction. A suspicious transaction in [16] is defined as a transaction in which the feedback provided by downloader is different from the opinion of the third party about the authentic behaviour of the uploader. To illustrate this concept, we consider a transaction that delivers chunk C from peer P_i to peer P_j occurred in service period q. This means that, if $A_{i,j}^{c} = 1$ and $AB_j^{q-1} < 1$, or if $A_{i,j}^{c} = -1$ and $AB_j^{q-1} \geq 1$, then we consider this transaction is suspicious and the feedback that peer P_i provides is also dishonest. The feedback credibility of a peer is defined as the ratio of number of honest feedbacks to total number of feedbacks provided by the peer. Then after receiving all feedbacks, $P_{hash(i)}$ updates the value of $D_{i,*}^{+}$, $D_{i,*}^{-}$ and FB_i^q as follows:

$N_i^* = 0$
$for\ k = 1\ to\ AC_i^q$
  $if\ (A_{i,server(k)}^{k} * (AB_{server(k)}^{q-1} - 1) < 0)$
    $N_i^* = N_i^* + 1$
  $endif$
  $if (A_{i,server(k)}^{k} == 1)$
    $D_{i,*}^{+} = D_{i,*}^{+} + SizeF$
  $else$
    $D_{i,*}^{-} = D_{i,*}^{-} + SizeF$
  $endif$
$endfor$
$FB_i = 1 - \frac{N_i^*}{AC_i^q}$

Then we can obtain the feedback credibility of peer P_i until service period q:

$$FB_i^q = \begin{cases} FB_i\ , & if\ \ q = 1 \\ (1-\alpha) * FB_i^{q-1} + \alpha * FB_i\ , otherwise \end{cases} \quad (1)$$

To motivate peers to display good behaviour, we think that peers' previous actions have an impact on their future interactions. So we consider the feedback behaviour both in current service period and in previous period when computing the feedback credibility of peer P_i. $\alpha \in (0,1]$ is introduced to reflect the dynamic nature of peer behaviour: the larger α is, the more attention we pay on feedback behaviour in current period.

After calculating the feedback credibility of peer P_i, for each uploader of peer P_i in current period, $P_{server(l)}$ for example, $P_{hash(i)}$ sends $A^l_{i,server(l)}$ an FB^q_i to $P_{hash\ (server(l))}$ to help it update service credibility of peer $P_{server(l)}$.

B. Authentic Behaviour

In each service period, every peer decides how to serve its received requests according to our proposed contribution -based service policy. We assume the current service period is q, and peer P_i has served NSR^q_i file requests in current service period. The downloader of the dth chunk is denoted as $requester(d)$. At the end of the service period, $P_{hash(i)}$ will receive $A^d_{requester(d),i}$ and $FB^q_{requester(d)}$ from the peers $P_{hash(requester(d))}, d \in \{1, 2, \dots, NSR^q_i\}$. After receiving all feedbacks, $P_{hash(i)}$ updates the values of $D^+_{*,i}$, $D^-_{*,i}$ and AB^q_i for peer P_i as follows:

$TD^{q+}_{*,i} = TD^{q-}_{*,i} = 0$
$for \quad d = 1\ to\ NSR^q_i$
  $if\left(A^d_{requester(d),i} == 1\right)$
    $TD^{q+}_{*,i} = TD^{q+}_{*,i} + FB^q_{requester(d)} * SizeF$
  $else$
    $TD^{q-}_{*,i} = TD^{q-}_{*,i} + FB^q_{requester(d)} * SizeF$
  $endif$
$end\ for$
$D^+_{*,i} = D^+_{*,i} + TD^{q+}_{*,i}$
$D^-_{*,i} = D^-_{*,i} + TD^{q-}_{*,i}$

The credibility of peer P_i in providing authentic files in service period q is denoted as AB_i:

$$AB_i = \begin{cases} min\left(\frac{TD^{q+}_{*,i}}{TD^{q-}_{*,i}}, MaxQ\right), if\ TD^{q-}_{*,i} \neq 0 \\ MaxQ \qquad\qquad , otherwise \end{cases} \quad (2)$$

In the current period q, when the files uploaded by peer P_i are all authentic or the size of the satisfied uploads is larger than $MaxQ$ times of the size of unsatisfied uploads, i.e., $TD^{q+}_{*,i}/TD^{q-}_{*,i} \geq MaxQ$ or $TD^{q-}_{*,i} = 0$, we think peer P_i is fully trusted in providing satisfied files during current period, and service credibility of peer P_i in current period q is set to $MaxQ$. We can easily obtain that AB_i is between 0 and $MaxQ$.

Then, the service credibility of peer P_i until the current service period q is computed as:

$$AB^q_i = \begin{cases} AB_i\ , & if\ \ q = 1 \\ (1-\beta) * AB^{q-1}_i + \beta * AB_i\ , & otherwise \end{cases} \quad (3)$$

In which, $\beta \in (0,1]$ is introduced to reflect the dynamic nature of peer behaviour and the initial service credibility AB^0_i of peer P_i is set to 0. When computing the value of service credibility, we consider the authentic behaviour of peer P_i in both current period and previous periods. That is to say, a peer's service credibility is based on its past interaction with other peers till period q. Therefore the service credibility value AB^q_i can subjectively indicate how reliable the peer P_i is in proving authentic files until period q. And we use the value of AB^q_i to predict the authentic behaviour of peer P_i in the next service period q+1: if $AB^q_i \geq 1$, that is to say, the size of satisfied files uploaded by peer P_i is larger than the size of unsatisfied files, we consider peer P_i as a trusted peer in providing authentic files, and we predict the peer will upload satisfied files in the next service period q+1, otherwise, we think peer P_i will provide unsatisfied files in the next period. In addition, $MaxQ$ can take any value greater than 1, for $AB^q_i = 1$ is the threshold value for a peer to be trusted in providing satisfied files, however $AB^q_i = MaxQ$ means peer P_i is fully trusted.

In the following, we give an explanation why we adopt ratio between size of satisfied uploads and size of unsatisfied uploads as the service credibility value of peers. Now we check how the feedback from downloader P_{n_r} with feedback credibility v affects the service credibility of server P_{n_s}. Suppose at the end of certain service period, $P_{hash(n_s)}$ received feedbacks about the quality of chunks provided by P_{n_s} from downloaders of peer P_{n_s}, including peer P_{n_r}. Without considering the feedback of peer P_{n_r}, we assume the service credibility of peer P_{n_s} is $\frac{M}{N}$. After including feedback of P_{n_r}, we can get the change in the service credibility value of server P_{n_s}, denoted as ΔAB:

$$\Delta AB = \begin{cases} \frac{v * SizeF}{N}\ , & if\ \ P_{n_r} = 1 \\ \frac{M}{N}\frac{v * SizeF}{[N + v * SizeF]}\ , & otherwise \end{cases} \quad (4)$$

We can see that the value of ΔAB increases with $v * SizeF$. As the value of $SizeF$ is constant, this means that value of ΔAB increases with v, i.e., feedback credibility of the downloader P_{n_r}. Thus it's easy to see that feedbacks from these downloaders with higher feedback credibility will have a greater effect on the service credibility of the uploader than feedbacks from those downloaders with lower feedback credibility. On one hand, since lying peers always have a lower feedback credibility value, their impacts on the service credibility of the uploader will be minimized. On the other hand, those good peers who always provide honest feedbacks will keep having a greater impact. Therefore we can effectively handle with the dishonest feedback and colluding peers, and therefore the service credibility can more accurately reflect peers' behaviour of proving satisfied files.

C. Contribution Behavior

At the end of service period q, the third party peer $P_{hash(i)}$ computes the contribution value of peer P_i, denoted as CTB^q_i. Before computing CTB^q_i, we introduce $Involvement^q_i$ to describe the contribution behaviour of peer P_i which is defined as:

$$Involvement^q_i = \begin{cases} \frac{D^+_{*,i} - D^-_{*,i}}{D^+_{i,*} + D^-_{i,*}}, & if\ D^+_{i,*} + D^-_{i,*} \neq 0 \\ D^+_{*,i} - D^-_{*,i}, & otherwise \end{cases} \quad (5)$$

$Involvement^q_i$ describes how many satisfied files peer P_i uploaded to the system compared to the size of files it obtained from the system until the service period q. Then CTB^q_i, i.e., the contribution value of peer P_i, is computed as:

$$CTB^q_i = min\left(max\left(Involvement^q_i, 0\right), 1\right) \quad (6)$$

In case that $Involvement^q_i \geq 1$, CTB^q_i is set to 1 which means that peer P_i is contributing to the system more than

what it obtains. In other words, it's a contributor. In other case that $Involvement_i^q \leq 0$, CTB_i^q is set to 0 which means the unsatisfied uploads of peer P_i are more than the satisfied ones. The term $D_{*,i}^+ - D_{*,i}^-$ means that the contribution value is sensitive to the peer's maliciousness. This term affects both free riders and malicious peers, and thus the value will be smaller for both free riders and malicious peers. For free riders who only download from the system without uploading to others, if they want to increase their contribution values, they have to upload positively to the system. For malicious peers, if they want to increase their contribution values, they have to change their behaviour to upload authentic files to the system.

In addition, every peer in the P2P systems possesses two private parameters which are hidden to other peers and only known to themselves. Take peer P_i for example, the one is willingness-to-serve probability $ProbS_i^q$ which is the probability at which peer P_i will serve the arriving requests in the service period q. The willingness-to-serve probability is a dynamic parameter which reflects the rational behaviour of peers in P2P file sharing systems in this paper. Note that a peer using a high value of willingness-to-serve probability means that the peer is more willing to provide service. The other is request successful rate σ_i^q which is the percentage of file requests that are successful served by other peers for peer P_i in the service period q.

V. Service Policy Based On Contribution

Suppose peer P_I receives H file requests in a given service period q, we use an identifier vector $I = \{I_1, I_2, \ldots, I_H\}, P_i \notin I$, each element of which represents a specific requester of peer P_i. Similarly, the contribution vector, $\mathrm{CTB} = \{\mathrm{CTB}_{I_1}^{q-1}, \mathrm{CTB}_{I_2}^{q-1}, \ldots, \mathrm{CTB}_{I_H}^{q-1}\}$, is introduced to describe the contribution values of requesters, in which $\mathrm{CTB}_{I_t}^{q-1}$ represents the contribution value of requester $I_t, t \in \{1,2,\ldots,H\}$, until service period $q-1$. Peer P_i with willingness-to-serve probability $ProbS_i^q$ will serve its received file requests according to contribution values of these requesters, the file request of requester $I_m, I_m \in I$, will be served at the probability $ProbRA_m$, which is defined as follows:

$$ProbRA_m^c = \begin{cases} \dfrac{CTB_{I_m}^{q-1}}{\sum_{s=1}^{H} CTB_{I_s}^{q-1}}, & if \sum_{s=1}^{H} CTB_{I_s}^{q-1} \neq 0 \\ 0, & otherwise \end{cases} \quad (7)$$

$$ProbRA_m = ProbS_i^q * ProbRA_m^c, \quad (8)$$

From Eq. (7), firstly, we can know that the larger the contribution value, the greater the probability at which the file requests sent by the peer will be served. As we mentioned before, the contribution value is sensitive to malicious uploads, and therefore the contribution values of free riders and malicious peers are both low, so the file requests of free riders and malicious requests will be served at a lower probability as a punishment. If they want to obtain files from the P2P system, they have to change their behaviour to actively upload authentic files to the system. Secondly, we easily obtain the fact that peers in the same P2P file sharing system create a competitive environment that will push peers to continuously upload to the system. That's to say, take peer P_i for example, when other peers in the system increase their contribution values by positively uploading to P2P system and peer P_i keeps its contribution value constant, then the probability at which the requests of peer P_i are served will decline. So for peer P_i, if it wants to maintain a certain success rate of requests, it must increase its contribution value correspondingly by increasing its positive uploads.

VI. Contribution-Level-Based Server Selection Policy

Papaionanous et al. introduce two different dimensions of reputation-based policies, provider selection (server selection policy) and contention resolution (service policy), and point out that a service policy combined with appropriate server selection policy can effectively enhance the performance of P2P systems [9]. So in this paper we propose a contribution -level-based server selection policy to help contribution-based service policy enhance the performance of our system.

First we introduce the concept of contribution level. Suppose the total contribution level in the system is $TClevel$, the contribution level of the peer with contribution value $convalue$ is defined as follows:

$$Clevel(convalue) = \begin{cases} 1, & if\ convalue = 0 \\ l, & if\ \dfrac{l-1}{TClevel} < convalue \leq \dfrac{l}{TClevel} \\ & (l = 1, \ldots, TClevel) \end{cases}, \quad (9)$$

In which $Clevel(\cdot)$ is a function which maps the contribution value into contribution level.

Having the concept of contribution level, we could introduce the following server selection policy: according to the server selection policy, the file chunks of peer P_i can only be accessed by peers with the same contribution level. It means that the peer with contribution level $l \in \{1, \ldots, TClevel\}$ can only send its file requests to those peers with the contribution level being l. Take peer P_i for example, suppose P_i wants to obtain files from the system at the beginning of service period q, it will adopt random server selection in the peer set $ST = \{P_j | Clevel(CTB_j^q) = Clevel(CTB_i^q)\}$ to select servers to send its file requests to.

The features of the server selection policy include: Firstly, the server selection policy restricts that peers with low contribution level can only access the files of these peers with low contribution level, and similarly files of peers with high contribution level can only be accessed by peers with high contribution value. Take Gnutella for example, just as we mentioned above, 85% of users in Gnutella are free-riders, and the contribution values of these free riders will be much smaller than that of contributors. As a result, free riders can only access files of other free riders of the same system according to contribution-level-based server selection policy. As the willingness-to-serve probability of free riders is extremely low, even almost zero, file requests of free riders will be served at a very probability according to the proposed contribution-based service policy. On the other hand, in the same way, requests of those peers with high contribution value will be served at a relatively higher probability. It

means our scheme can effectively provide service differentiation. Secondly, the selection policy can balance requests load among peers, therefore avoid more service requests directing to the peers with high contribution value. In addition, it can help peers being in the process of building their contribution to build their contribution values for there are always file requests from peers with the same contribution value directed to them.

For newcomers, the initial contribution values are set to 0. We assume peer P_i enters the P2P system for the first time in the service period q. On one hand, in order to help peer P_i obtain files from P2P system to serve other peers, for a short period time which we call acquaintance duration, new peers direct their requests to other peers in the same P2P system with equal probability. For the given peer P_j who receives file requests from peer P_i in the period q during acquaintance time, peer P_j will serve the requests at probability $ProbS_i^q$. On the other hand, to help peer P_i build its contribution value to continuously obtain files after acquaintance duration, we should guarantee there are enough file requests directed to it, and at the same time, for those pre-existed peers with lowest contribution level, we should restrict their downloads from the newcomer when the newcomer is highly cooperative. Therefore, during the acquaintance time, peer P_j for example, adopts random server selection policy in the peer set:

$$\{ P_k \mid Clevel(CTB_k^q) = Clevel(CTB_i^q) \} \cup P_i$$

to select server. After time period equal to acquaintance time, we adopt the contribution-based service policy to serve P_i, and similarly P_i chooses server according to contribution -level-based server selection.

VII. Rational Behavior Of Peers

We assume that P2P system consists of rational peers who aim at maximizing their own benefits defined as request successful rate in this paper. We use the dynamic strategy similar to [17] to describe rationality of peers in the system. More details can refer to [17]. The way in which peer P_i changes its willingness-to-serve probability is described as follows:

$$\begin{aligned}
&if(q == 1)\\
&\quad if\ \ randomprobability < 0.5\\
&\quad\quad NAction = 1\\
&\quad else\\
&\quad\quad NAction = -1\\
&\quad endif\\
&else\\
&\quad\quad \sigma_i^q = \frac{SuccessRequest_i^q}{Request_i^q}\\
&\quad\quad NAction = LAction\\
&\quad\quad if\left(\sigma_i^q < \sigma_i^{q-1}\right)\\
&\quad\quad\quad NAction = -LAction\\
&\quad\quad else\ \ if\left(\sigma_i^q == \sigma_i^{q-1}\ and\ \ \sigma_i^q \le 0.1\right)\\
&\quad\quad\quad NAction = 1\\
&\quad\quad endif\\
&endif\\
&ProbS_i^{q+1} = Operate\left(NAction, ProbS_i^q\right)
\end{aligned}$$

$$\begin{aligned}
&LAction = NAction\\
&Operate\left(NAction, ProbS_i^q\right)\\
&= \begin{cases} min\left(ProbS_i^q + Increment, 1\right), if\ \ NAction = 1 \\ max\left(ProbS_i^q - Increment, 0\right), \quad otherwise \end{cases}
\end{aligned}$$

In which, $SuccessRequest_i^q$ and $Request_i^q$ represent the number of requests successfully performed by other peers for peer P_i and the total number of requests sent by peer P_i in service period q respectively; $LAction$, $NAction \in \{1, -1\}$, denotes action performed on willingness-to-serve probability during the previous period and current period respectively. The value 1 denotes peer P_i increases its willingness-to-serve probability, and -1 means peer P_i lowers the probability. The concrete value of Increment is set to 0.05 in the simulation section.

According to the strategy, peer P_i for example, periodically measures its request successful rate at the end of each service period, and changes its willingness-to-serve probability depending on the effect of the previous action on its benefit. That is to say, if the new strategy $ProbS_i^q$ brings out the benefit σ_i^q higher than benefit σ_i^{q-1}, then the same action as $LAction$ will be performed on willingness-to-serve probability; otherwise the opposite of $LAction$ will be executed. In addition, at the end of the first service period, peers choose their actions with equal probability.

VIII. Distributed Contribution Management

In this section, we describe the distributed contribution management scheme to provide protection against possible cheating and to properly adjust the contribution data for peers. To illustrate this idea, we consider a transaction that sends chunk C from peer P_i to peer P_j occurred in service period q.

Contribution data that are needed to describe the contribution behaviour of peers and to compute the contribution values of peers are stored across the network in a distributed manner. To enhance the trustiness of contribution data, thus subjectively reflecting contribution behaviour of peers, we store the contribution data of peer P_i at the third party peer P_h determined by $h = hash(i)$ with being a function known to all peers. For example, we can use a distributed hash table mechanism such as Chord [16] to maintain the contribution data in a scalable manner. It means the contribution date of peer P_i is stored at peer P_h. In general, we can expect peer P_i has no management authority on altering the contribution value stored at the third party peer P_h. To ensure that the third party peer P_h provide honest contribution value for peer P_i, we can store the contribution data of peer P_i at several different third party peers whose coordinates are defined by applying a set of hash functions $h_1, h_2, \ldots, h_M$, to a peer's unique identifier, which also provides a feasible method to deal with the case in which some of the third party peers P_{h_i} are offline. Furthermore, as peers join/leave the system from time to time, considering the situation that all the third party peers of peer P_i are offline simultaneously in a certain service period, then we will raise the values of some relevant coefficients (such as α, β) to a relatively high level when calculating the data of peer P_i, without regard to the previous values as well, at the end of this service period. Then we store the data at other new online

third party peers acquired by above $hash$ functions. To simplify description, we assume that contribution date of peer P_i is only stored at peer P_h.

To make sure that the third party peer P_h and peer $P_{hash(j)}$ will accurately adjust the contribution data of peer P_i and P_j, that is to say, peer P_h should know peer P_i has provided chunk C to peer P_j, and similarly $P_{hash(j)}$ should also know that peer P_j has received chunk C from peer P_i, peer P_i will send chunk C encrypted in session key k to peer P_j, then sent the key k plus the identifier of the requester P_j to its third party P_h. After receiving the key k and the identifier P_j, the third party peer P_h send the key k to peer $P_{hash(j)}$. Peer P_j can only decrypt the file by requesting the key k from its third party peer $P_{hash(j)}$ and then receiving it. After decrypting the chunk, peer P_j will send feedback $A_{i,j}^C$ and identifier of uploader P_i to $P_{hash(j)}$, and then $P_{hash(j)}$ sends $A_{i,j}^C$ and FB_j^q to P_h. At last peer P_h and $P_{hash(j)}$ update the contribution data of P_i and P_j respectively.

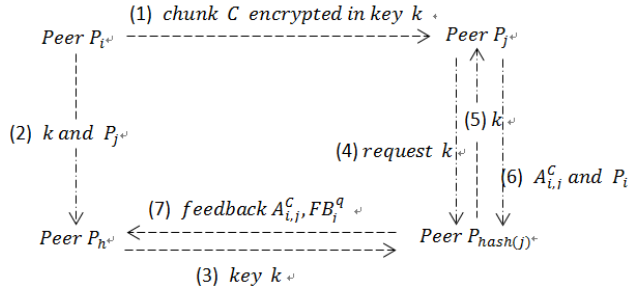

Fig.1 Distributed Contribution Data Management Using Third Party Peer

The third party peer P_h of peer P_i is responsible for the following tasks:

In case peer P_i serves as uploader, peer P_h receives session key k from peer P_i, and then sends key k to the third party of the peer requesting the file encrypted in session key k.

In other case peer P_i serves as a downloader, peer P_h receives feedbacks from peer P_i, and then sends feedbacks and the feedback credibility of peer P_i to the third party peer of the peer uploading chunks to peer P_i.

Adjust contribution data of peer P_i dynamically according to behaviour of peer P_i at the end of each service period.

IX. Performance Evaluation

A. Simulation

In order to evaluate the performance of our system, we simulate a P2P file sharing system consisting of 100 peers. Peers are divided into two categories: Contributors and Free riders. Free riders constitute 70% of the peers, for each category, 30% of the peers are malicious peers that upload inauthentic files. Peers' behaviour and distribution are summarized in Table I. We consider a homogeneous P2P file sharing system of peers who have the same request generation rate $g = 2$, i.e., number of requests sent by a peer in a service period. Acquaintance duration is set to 100 service periods. The total simulation time is set to 1000 service periods, the total contribution level is set to 3, and the service credibility of peers of being fully trusted in providing authentic files $MaxQ$ is set to 2.

TABLE I. Peer's Behaviour and Distribution

Category	Percentage	Probability to send inauthentic files	
		Malicious 30%	Not Malicious70%
Contributors	30%	0.9	0.01
Free Riders	70%	0.9	0.01

In Table I, peers with indices from 1 to 70 belong to the category of free riders (FR), peers with indices from 71 to 100 belongs to the category of contributor peers (CP). Accordingly, peers with indices from 1 to 49 are good free riders (GFR) and peers with indices from 50 to 70 are malicious in addition to free riders (MFR). Peers with indices from 71 to 79 are malicious contributor peers (MCP) that provide inauthentic files but still participating in uploading files to other peers. Peers with indices from 80 to 100 are good contributor peers (GCP). We consider a situation where we have a high percentage of free riders to show the effectiveness of our proposed scheme in identifying and handling both free riders and malicious.

B. Feedback Behavior

In this section, we study the feedback behaviours of peers in providing honest feedbacks. We assume peers with four behaviour types mentioned above provide honest feedback at different probability. In detail, GFR, MFR, MCP, GCP provide honest feedbacks with probability 0.5, 0.1, 0.1, 0.9 respectively and α is set to 0.9. To testify the effectiveness of our method in detecting peers providing dishonest feedback, we adopt the random policy, which is to say that, peers use random server selection policy with no service differentiation scheme. Fig.1 depicts the feedback behaviour of peers when using the method of computing feedback credibility value introduced in Section 4.1.

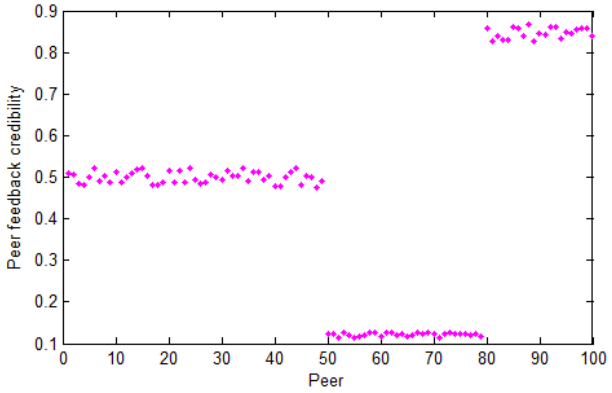

Fig.2 Peer Feedback Behaviour

Fig.2 shows that feedback credibility value is a good indicator of the liar behaviour of peers. Indeed, good contributors (with indices from 80 to 100) have a high value of feedback credibility, while the liar peers (from 1 to 79) have a lower feedback credibility value. This indicator is also able to differentiate peers with different lying degree. Peers

with lower probability of providing dishonest feedback (indices from 1 to 49) have higher feedback credibility than those with higher probability of lying (from 50 to 79).

C. Static Behavior

As we have confirmed the ability of our proposed policy to detect peers providing dishonest feedbacks in section 9.2, and the method of computing service credibility value can minimize the effect of the dishonest feedbacks on the service credibility value of uploaders as we mentioned in section 4.2. So in this section, we don't consider peers that lie in their feedbacks. In the first set of simulations, we consider static peer behaviour. This means that peers don't change their willingness-to-serve probability during the whole simulation time. We will compare the following schemes:

1) Service-credibility-based service policy with random server selection policy (SCNO). Since the service credibility value is between 0 and $MaxQ$. In this scheme the probability $ProbRA_l^C$ in service period q is computed as follow: $ProbRA_l^C = \frac{AB_l^q}{MaxQ}$.
2) Contribution-based service policy with most contributable server selection policy. The most contributable peers refer to those peers whose contribution value is on the top 20% (CBMC).
3) Contribution-based service policy with contribution -level-based server selection policy (CBCB).

Free riders and contributor peers share files with probability 0.05, 1 respectively. In order to help peers identify behaviour of other peers in the same file sharing systems, in the first 100 service periods, we adopt random server selection policy. In this simulation, we will focus on the following performance parameters:

1) The number of successful requests: computed as the total number of requests that have been performed successfully by other peers during the simulation period. As we consider a homogeneous P2P file sharing system in which peers have the same request generation rate, thus the number of successful requests can accurately reflect the request successful rate.
2) Peer contribution value: shows the contribution behaviour of the peer which is computed by using the Eq. (6) mentioned above.
3) Peer load share: this is computed as the sum of the service requests directing to the peer.

1) *Service Differentiation Based on the Service Credibility Value:*

In this case that service credibility value of peer is considered as the guideline for service differentiation. Good free riders can obtain high service credibility value almost the same as good contributors by only uploading few authentic files. As a result, good free riders and good contributors receive the same level of service. It is unfair for good contributors who serve most of the service requests of the system without receiving any special rewards.

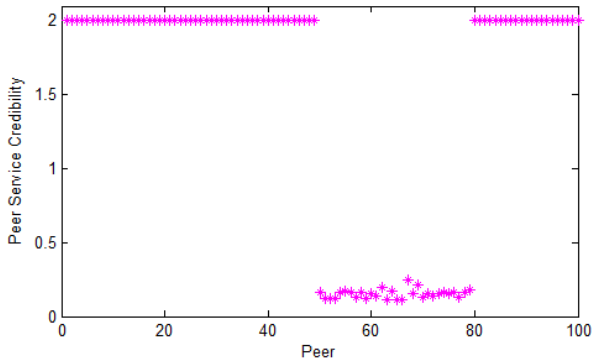

Fig.3 Peer Authentic Behavior in SCNO

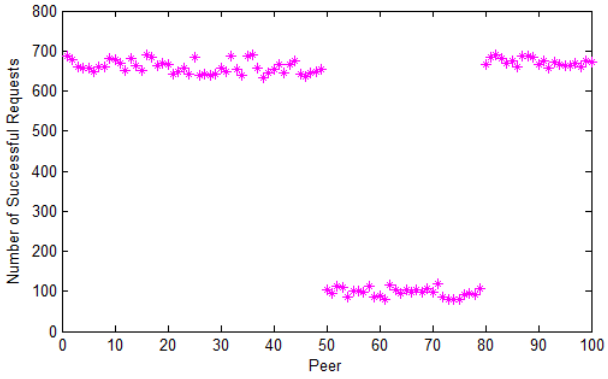

Fig.4 Number of successful requests for SCNO

Fig.3 and Fig.4 depict the service credibility value and the number of successful requests of peers with different behaviour. From the two figures, we can see that the policy SCNO can't identify between good free riders and good contributors. As we adopt random server selection, there are always requests directing to good free riders. Even though the probability of providing service is small, once they serve a file request directing to them, they will gain a high service credibility value. As a result, service credibility of peers belonging to the two types of both good free rider and good contributors are almost similar, and therefore they receive similar level of service. In a word, the policy SCNO cannot differentiate good free rider and good contributors.

2) *Contribution-based Service Differentiation:*

Fig.6 depicts the contribution values of peers when adopting contribution-based service policy and the contribution-level-based server selection policy (CBCB scheme). By comparing this figure with Fig.5, we can notice that contribution behaviour value is a good indicator of peer's participation in the file sharing systems. In other words, a peer with high contribution value is serving more files than peers with a low contribution level. Note that contribution values of malicious peers (49-79) are near to zero and contribution values of good free riders are much smaller than good contributors. This can be explained by the fact that malicious peers upload much more malicious files than satisfied files and good free riders obtain more files compared to its satisfied uploads. From Fig.6, we can easily get the fact that the contribution value can effectively identify both free riders and malicious peers.

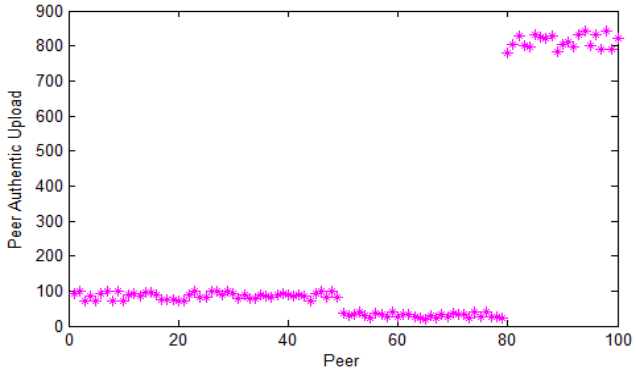

Fig.5 Authentic Upload in CBCB

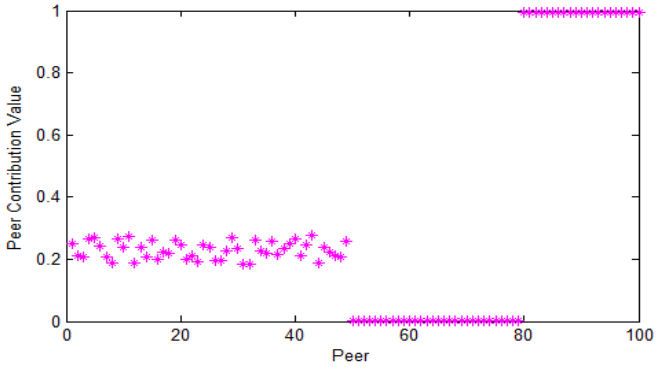

Fig.6 Contribution Behavior in CBCB

Fig.7 and Fig.8 show the number of successful requests of peers. First, from the two figures, we can notice that successful requests of good contributors are much more than both free riders and malicious peers. It means that considering contribution value as a guideline for service differentiation will effectively reward good contributors and punish both free riders and malicious peers. Second, comparing Fig.7 with Fig.8, we can notice that the total successful requests for scheme CBCB is more than total successful requests of scheme CBMC, and, in particular, the increase in number of successful requests of contributors is larger compared to other types of peers. It means the proposed contribution-level-based server selection policy can help contribution-based service policy use resource more efficiently. Thirdly, Fig.9 and Fig.10 show the number of requests directing to peers with different behaviour, when using CBMC, most of the requests direct to the peers with large contribution value, however when using CBCB, the service requests direct to peers of the same P2P file sharing systems on average, which can reduce the load of contributors.

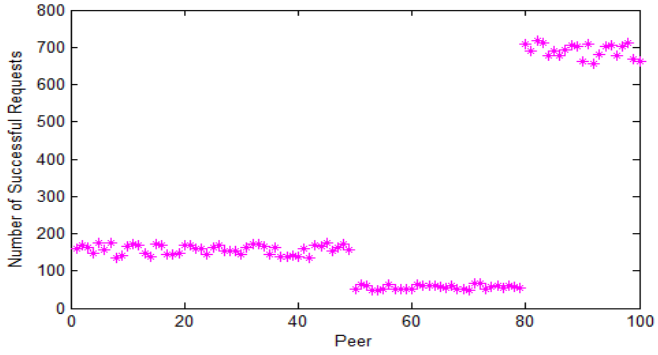

Fig.7 Number of successful requests in CBMC

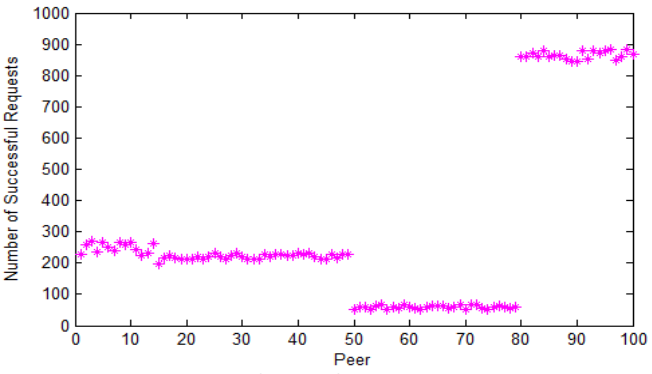

Fig.8 Number of successful requests in CBCB

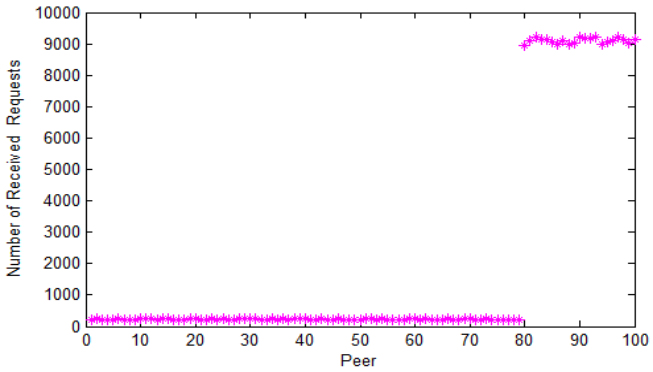

Fig. 9 Number of Received Requests in CBMC

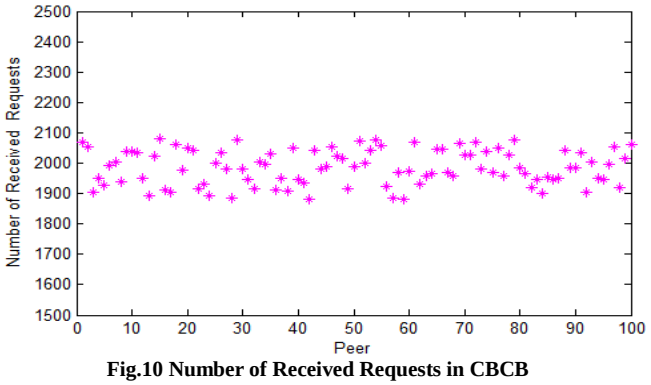

Fig.10 Number of Received Requests in CBCB

In the above simulations, we assumed a static peer behaviour to testify the ability of our CBCB scheme in detecting both free riders and malicious peers. In real life systems, rational peers will tend to change their behaviour. Free riding peers with a rational behaviour will change from free riders to contributors.

D. Rational Behavior

In this section, we assume that peers use rational behaviour as presented in section 7. The goal is to show that using rational behaviour, free riders will change their behaviour from free riding to sharing and uploading files. Regarding malicious peers who upload malicious files, they have to change their probability in providing authentic files in order to increase their contribution values, which is similar to the rational behaviour of good free riders, i.e., change their willingness-to-serve probability. So we just need to study the rational behaviour of good free riders. In the simulation we consider the system consisting of good free riders and good contributors in order to observe the rational behaviour of free riders. Peers with indices from 1 to 70 are good free riders,

with indices from 71 to 100 are good contributors. Initially, free riders share files with probability 0.05, and contributor peers with a probability equal to 1. The probability of sharing is increased or decreased by Increment set to 0.05.

Fig.11 shows the average peer willingness-to-serve probability for different categories of peers. At the beginning of the simulation, the willingness-to-serve probability of free riders is very low, for they serve file requests with a very low probability. In order to obtain files from the file sharing systems, they increase their probability of sharing files. As we mentioned above, peers in the same file sharing systems create a competitive environment, as a results, free riders continuously increase their probability of sharing until they reach a similar value, close to 1, as good contributors. As we adopt the contribution-level-based server selection policy, therefore, for contributors, their probabilities of sharing files slightly decrease.

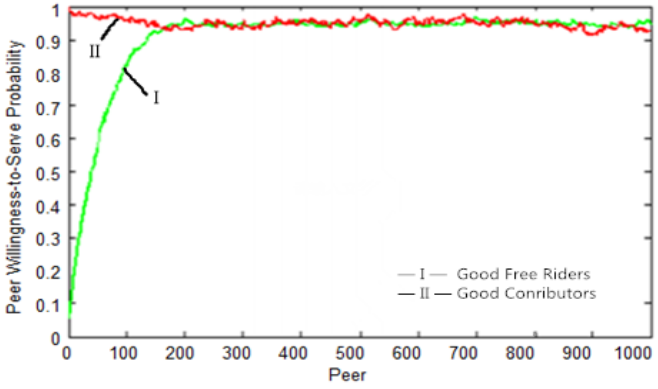

Fig. 11 Peers' rational behavior

X. Conclusion

In this paper, we adopt a new scheme to assess the contribution values of peers and consider the contribution value as a guideline for service differentiation. The contribution values of peers treat the authentic uploads and malicious uploads differently in order to identify both free riders and malicious peers. In order to motivate both free riders and malicious to positively cooperation with other peers, we propose the service policy based on contribution and contribution-level-based server selection policy. The contribution-based service policy combined with contribution -level-based server selection policy can provide right incentives for free riders and malicious peers to change their behaviour to provide positive uploads to the system. Performance evaluations confirm the ability of our proposed policy to reward good contributors and punish both free riders and malicious peers. Moreover the proposed scheme can create a competitive environment and balance request loads among peers of the same file sharing systems.

Acknowledgement

This work is supported in part by the National Natural Science Foundation of China under grant Nos. 61174124, 61233003, in part by Research Fund for the Doctoral Program of Higher Education of China under grant No. 20123402110029 and in part by Natural Science Research Program of the Anhui High Education Bureau of China under grant No. KJ2012A286.

References

[1] Hughes D, Coulson G, Walkerdine J. Free Riding on Gnutella Revisited: The Bell Tolls? IEEE Distributed Systems Online, 2005, 6(6): 1-15.
[2] Hardin G. The Tragedy of the Commons, Science, 1968, 1243-1248.
[3] Kumar C, Kemal A, De P. A mechanism for pricing and resource allocation in peer-to-peer networks. Electronic Commerce Research and Applications, 2011, 10: 26-37.
[4] Satsiou A, Tassiulas L. Trust-based exchange of services to motivate cooperation in P2P networks. Peer-to-Peer Networking and Applications, 2011, 4(2): 122-145.
[5] SelÃ§uk A A, Uzun E, Pariente M R. A Reputation-based Trust Management System for P2P Networks. International Journal of Network Security, 2008, 6(2): 235-245.
[6] Papaionanou T G, Stamoulis G D. Reputation-based Policies that Provide the Right Incentives in Peer-to-Peer Environments. Computer Networks, special Issues on management in peer-to-peer systems: trust, reputation and security, 2006, 50(4): 563-578.
[7] Hawa M, As-Sayid-Ahmad L, Khalaf L D. On enhancing reputation management using Peer-to-Peer interaction history. Peer-to-Peer Networking and Applications, 2013, 6 (1): 101-113.
[8] Wang Y F, Nakao A, Vasilakos A V, Ma J H. P2P soft security: On evolutionary dynamics of P2P incentive mechanism. Computer Communications, 2010, 34(3): 241-249.
[9] Sasabe M, Wakamiya N, Murata M. User selfishness vs. file availability in P2P file-sharing systems: Evolutionary game theoretic approach. Peer-to-Peer Networking and Applications, 2010, 3(1): 17-26.
[10] Park J, Van Der Schaar M. A game theoretic analysis of incentives in content production and sharing over peer- to-peer networks. IEEE Journal of Selected Topics in Signal Processing, 2010, 4(4): 704–717.
[11] Prasad R, Vegi S, Kumari V V, Raju K. An Effective Calculation of Reputation in P2P Networks. Journal of Networks, 2009, 4(5): 332-342.
[12] Fedotova N, Veltri L. Reputation management algorithms for DHT-based peer-to-peer environment. Computer Communications, 2009, 32(12): 1400-1409.
[13] Chu X W, Chen X W, Zhao K Y, Liu J C. Reputation and trust management in heterogeneous peer-to-peer networks. Telecommunication Systems, 2010, 44(3): 191-203.
[14] Pogkas I, Kriakov V, Chen Z Q, Delis A. Adaptive neighborhood selection in peer-to-peer networks based on content similarity and reputation. Peer-to-Peer Networking and Applications, 2009, 2(1): 37-59.
[15] Zhang Y, Fang Y. A Fine-garined Reputation System for Reliable Service Selection in Peer-to-Peer Networks, IEEE, Tans. Parallel and Distributed Systems, 2007, 18(8): 1134- 1145.
[16] Stocia I, Morris R, Karger D, Kaashoek M F, Balakrishnany H. Chord: A scalable Peer-to-Peer Lookup Service for Internet Applications. IEEE/ACM Trans. Netw, 2003, 11(1): 17-32.
[17] Mekouar L, Iraqi Y, Boutaba R. A Contribution-based Service Differentiation Scheme for Peer-to-Peer Systems, Peer-to-Peer Netw Appl, 2009, 2(2): 146-163.

Jinfa Yao is currently a postgraduate in Control Engineering at School of Information Science and Technology, University of Science and Technology of China, Hefei, China. His research interests include network modeling and performance optimization.

Teng Wang received his M.S. degree from University of Science and Technology of China, Hefei, China. Her research interests include network modeling and performance optimization.

Baoqun Yin received his B.S. degree in fundamental mathematics from Sichuan University, Chengdu, China, in June 1985, his M.S. degree in applied mathematics from University of Science and Technology of China, Hefei, China, in May 1993, and his Ph.D. degree in pattern recognition and intelligent system from University of Science and Technology of China, Hefei, China, in Dec. 1998.He is currently a Professor of Control Science and Engineering at Department of Automation, University of Science and Technology of China, Hefei, China. His current research interests include discrete event dynamic systems, Markov decision processes, queuing systems, and network resource management and optimization.