Free Riders under Control through Service Differentiation in Peer-to-Peer Systems

Loubna Mekouar
University of Waterloo
Waterloo, Canada
Email: lmekouar@bbcr.uwaterloo.ca

Youssef Iraqi
Dhofar University
Salalah, Oman
Email: y_iraqi@du.edu.om

Raouf Boutaba
University of Waterloo
Waterloo, Canada
Email: rboutaba@bbcr.uwaterloo.ca

Abstract— **Trust is required in a file sharing peer-to-peer system to achieve better cooperation among peers. In reputation-based peer-to-peer systems, reputation is used to build trust among peers. In these systems, highly reputable peers will usually be selected to upload requested files, decreasing significantly malicious uploads in the system. However, these peers need to be motivated to upload files by increasing the benefits that they receive from the system. In addition, it is necessary to motivate free riders to contribute to the system by sharing files. Malicious peers must be forced to contribute positively by uploading authentic files instead of malicious ones. In this paper, the *Contribution Behavior* of the peer is used as a guideline for service differentiation. The new concept of *Availability* is introduced for partially-decentralized peer-to-peer systems. Both *Availability* and *Involvement* of the peer are used to assess its *Contribution Behavior*. Simulation results confirm the ability of the proposed scheme to effectively identify both free riders and malicious peers and reduce the level of service provided to them. Simulation results also confirm that based on *Rational Behavior*, peers are motivated to increase their contribution to receive services. Moreover, using our scheme, peers must continuously participate, reducing significantly the so-called *milking* phenomenon.**

I. Introduction

In recent years, peer-to-peer (P2P) file sharing systems have gained tremendous popularity. In these systems, peers communicate directly with each other to exchange information and share files.

In an open P2P system, peers often have to interact with strangers, cooperate with each other, and manage the risks involved with the interactions. The most important problem of P2P systems is that the open nature of these systems opens the door to misuse by malicious peers and abuse by free riders. Dealing with untrustworthy peers increases peers' frustration and disappointment. Trust is needed to achieve better cooperation among peers and maximize peers' satisfaction.

Diego Gambetta defines trust as follows [1]: "Trust (or, symmetrically, distrust) is a particular level of the subjective probability with which an agent assesses that another agent or group of agents will perform a particular action, both before he can monitor such action (or independently of his capacity ever to be able to monitor it) and in a context in which it affects his own action". Building trust is difficult especially with strangers in virtual communities. Marsh [2] is one of the first authors to give a formal model of trust that can be used in computer science. This model is based on properties of trust taken from sociology. Several reputation-based systems [3], [4], [5], [6], [7] were introduced to build trust by using peer reputation values as selection criteria to distinguish between malicious and non-malicious peers. The reputation value of a peer is based on its past interactions with other peers, and over time, peers learn about each other's real behavior.

A. Motivation and Contribution

Peers are motivated to display good behavior because it will have an impact on their future interactions. Political scientist Robert Axelrod refers to this phenomenon as *the shadow of the future* [8]. For example, in the case of the eBay reputation system, members need to build good reputation and maintain good transaction histories. Members with high reputation values are more selected and buyers trust to deal with them. In a peer-to-peer system, if all peers receive the same service regardless of their behavior, peers will not be motivated to strive for high reputation values since they will be always asked to upload files without receiving any special benefit or reward.

Most of reputation-based P2P systems [4], [9], [10] use the number of satisfied and/or unsatisfied transactions as a basis for computing the reputation of a peer. If service differentiation is enforced according to the peer's reputation, free riders can have high reputation values by uploading few authentic files. Then, they will start taking advantage of the system thanks to their high reputation values. This behavior is called the *milking* phenomenon.

For these reasons, the authors in [11] proposed a contribution management scheme that can be combined with a reputation management scheme. Peers will have to contribute to the system to get services. The *shadow of the future* is maintained because peers are forced to contribute to be served. The higher the contribution value, the greater the services available to the peer. In [11], the contribution of peers rather than the reputation of peers is used as a guideline for service differentiation.

The use of the contribution of peers allows to detect and punish free riders in addition to malicious peers by providing lower services to them compared to the services provided to good contributor peers. Furthermore, the use of the contribution of peers will reduce the milking phenomenon since peers

1-4244-0030-9/05/$20.00 ©2005 IEEE

have to be continuously participating to have their requests handled by the system.

However, having the contribution based on peer's uploads and downloads as a measure for service differentiation may lead to the peer's starvation. If the system does not allow some peers to download files, these peers may have no or not enough files to upload to other peers.

In this paper, we argue that the contribution of a peer should not be based only on its uploads compared to its downloads, but also on the availability of the peer. Recognizing peers that are available to upload requested files will help significantly peers that are in the process of building their reputation. These peers will receive services, and increase their contribution and reputation gradually.

The proposed scheme will allow to achieve the following objectives:

- Stopping the egoistical behavior of free riders that want only to take advantage of the system. Through service differentiation, peers receive services according to their contribution to the system.
- Providing the right incentives for free riders to change their behavior from free riding to positively contributing to the system.
- Creating a competitive environment that will push peers to continuously being available to provide files.
- Allowing new comers or formerly free riders to build their reputation and increase their contribution.

In this paper, we propose a new mechanism for measuring the contribution of a peer and we show in the performance evaluation section that we are able to achieve the goals mentioned above. Using our scheme, peers with a dynamic behavior, as in real life, are motivated to increase their contribution to obtain the maximum benefit from the system. Peers will have to participate on a regular basis which will decrease the *milking* phenomenon significantly.

The paper is organized as follows. Section II presents the general framework for trust management considered in this paper. Section III describes how contribution is computed and Section IV describes the service differentiation strategy proposed in this paper. Section V presents the rational behavior executed by peers and Section VI presents the performance evaluation of the proposed scheme. Finally, section VIII concludes the paper.

II. Trust Management

In this paper, we consider partially decentralized P2P systems. In these systems, peers connect to their supernodes that index shared files and proxy search requests on behalf of these peers. Queries are therefore sent to supernodes, not to other peers.

A. Notations and Assumptions

In the remaining of the paper, the following notations are used:

- Let P_i denotes peer i
- Let $D_{i,j}$ denotes the size of downloads performed by peer P_i from peer P_j
- Let $D_{i,*}$ denotes the size of downloads performed by peer P_i
- Let $D_{*,j}$ denotes the size of uploads by peer P_j
- Let $A^F_{i,j}$ be the appreciation of peer P_i after downloading file F from P_j
- Let $Sup(i)$ denotes the supernode of peer P_i

B. Peer Behavior

In a peer-to-peer file sharing system, peers are expected to practice a good peer-to-peer behavior. Peers are implicitly trusted that they will share good quality files, that they will upload requested files, and that they will send right feedback. Unfortunately, real life peer-to-peer systems have proved that a mechanism is needed to measure explicitly trust in order to deal only with trustworthy peers, and achieve fundamental goals of file-sharing systems.

In [11], the authors proposed to address trust according to the following dimensions:

1) *Authentic Behavior (AB)* is the reliability of a peer in providing accurate and good quality files. This value represents the reputation of a peer. It allows to differentiate between good and malicious peers.
2) *Credibility Behavior (CB)* represents the sincerity of a peer in providing a honest feedback. The *Credibility Behavior* allows to identify liar peers and represents peers' reputation in terms of credibility and sincerity.
3) *Contribution Behavior (CTB)* allows to distinguish between peers that contribute positively to the system (i.e., altruistic) and free riders (i.e., egoistic). In [11], the *Contribution Behavior* considers only peer's uploads and downloads.

In this paper, we recognize peers that were available to upload files and we reward them. In a reputation-based system with millions of users, the competition to upload requested files is very high. Since peers with higher reputation values are always chosen, these peers will have higher contribution values and will receive better services. Peers that are still in the process of building their reputation will not be selected to perform effectively the upload. These peers will receive lower services and will not be able to increase their contribution values. If the *Contribution Behavior* of a peer is computed based only on its uploads and downloads, some peers may wrongfully receive lower services. With the recognition of peers' availability, peers with a null or a low contribution value will have a chance to receive services, and build their reputation. These peers will, slowly, but surely, have their requests handled by the system. These peers will be able to download files, have more chances to share with others, and increase their reputation and contribution values gradually.

We propose that the *Contribution Behavior* of peers should be based on:

- Peers' *Availability*: being available for uploading requested files.

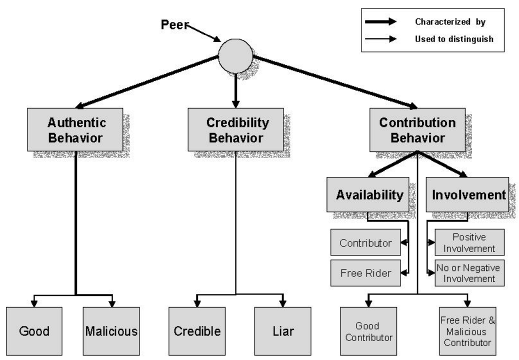

Fig. 1. Peer Trust and Behavior Analysis

- Peers' *Involvement*: non-malicious uploads performed versus downloads received by a peer.

The *Contribution Behavior* of a peer represents its reputation in terms of sharing files and positively contributing to the system.

It is important to clarify that the global reputation of a peer is based on its real behavior in terms of *Authentic Behavior* (sending authentic or inauthentic files), *Credibility Behavior* (lying or not in the feedback) and *Contribution Behavior* (availability and positive involvement). The trust given to peer P_i is characterized by the triplet $(AB_i, CB_i, CTB_i(Availability_i, Involvement_i))$. Peers with a good behavior are peers that send authentic files, right feedback (i.e. without lying), and are available to share files in addition of being effectively involved in uploading files. Figure 1 shows the three dimensions of trust along with different aspects of behavior that they characterize.

Figure 2 shows a typical file request-download procedure involving the sender and receiver peers and their supernodes. The figure also shows steps affected by the values of trust triplet. When peer P_i is requesting a search service Req_i^F from its supernode Sup_i, this latter will perform the request only after considering the *Contribution Behavior* of peer P_i. According to peer's *Availability* and *Involvement*, the request can be performed or rejected. When peer P_i is given a list of peers providing the requested file Res_i^F which represents the result of the search request, peer P_i will choose peer P_j according to the *Authentic Behavior* of P_j. Peer P_i is not interested to know other characteristics of peer P_j since the most important issue for peer P_i is to receive the exact requested file with a good quality. Peer P_i sends a request Req_{ij}^F to download file F from peer P_j. After downloading this file, peer P_i sends feedback $A_{i,j}^F$. The credibility of peer P_i will have a significant impact on feedback and the reputation of peer P_j.

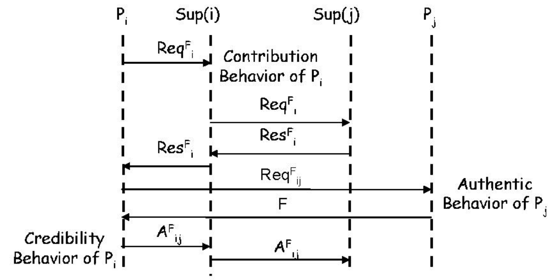

Fig. 2. A typical exchange between peers

C. Trust Mechanism

In this section, we describe briefly the trust mechanism considered in this paper. More details are provided in [12].

After downloading file F from peer P_j, peer P_i will evaluate this download. If the file received corresponds to the requested file, then P_i sets $A_{i,j}^F = 1$. If not, P_i sets $A_{i,j}^F = -1$. In the latter case, either the file has the same title as the requested file but different content, or that its quality is not acceptable.

Each peer P_i in the system has four values, called *reputation data* (REP_{P_i}), stored by its supernode:

1) $D_{i,*}^+$: Satisfied downloads of peer P_i from other peers,
2) $D_{i,*}^-$: Unsatisfied downloads of peer P_i from other peers,
3) $D_{*,i}^+$: Satisfied uploads from peer P_i to other peers,
4) $D_{*,i}^-$: Unsatisfied uploads from peer P_i to other peers

When peer P_i joins the system for the first time, all values of its *reputation data* REP_{P_i} are initialized to zero. When receiving the appreciation (i.e., $A_{i,j}^F$) of peer P_i, its supernode $Sup(i)$ will perform the following operation:

If $A_{i,j}^F = 1$ then $D_{i,*}^+ = D_{i,*}^+ + Size(F)$,
else $D_{i,*}^- = D_{i,*}^- + Size(F)$.

Then, the appreciation is sent to $Sup(j)$ that will perform the following operation:

If $A_{i,j}^F = 1$ then $D_{*,j}^+ = D_{*,j}^+ + Size(F)$,
else $D_{*,j}^- = D_{*,j}^- + Size(F)$.

Where $Size(F)$ denotes the size of the file F.

The *Authentic Behavior* of a peer P_j is computed as:

$$\begin{array}{ll} AB_j = \frac{D_{*,j}^+ - D_{*,j}^-}{D_{*,j}^+ + D_{*,j}^-} & \text{if } D_{*,j}^+ + D_{*,j}^- \neq 0 \\ AB_j = 0 & \text{otherwise} \end{array} \tag{1}$$

As in [7], the supernode of peer P_i sends (REP_{P_i}) periodically to the peer. The latter will keep a copy of (REP_{P_i}) to be used the next time it joins the system or if its supernode changes. To prevent tempering with (REP_{P_i}), the supernode digitally signs (REP_{P_i}).

III. Contribution Behavior

In this section, we present the two concepts of *Availability* and *Involvement* that make the contribution of a peer.

A. Peer Availability

When peer P_i requests a file and receives a list of peers providing this file, all these peers are available for an eventual upload. These peers can be considered as contributor peers (not free riders) since they are willing to upload the requested file. Since only after an upload is effectively performed that it can be assessed as good or malicious, all these peers have to be rewarded for being available irrespective of being good or malicious.

For each peer P_i in the system, $Available_i$ is added to its *reputation data* (REP_{P_i}), that is stored by its supernode Sup_i. The value of $Available_i$ represents the number of times, the peer P_i was available for an upload. This value is incremented by the supernode of P_i each time peer P_i is available to provide the requested file after a search request is received from a peer (that belongs to the same supernode) or from another supernode.

The availability of peer P_i $Availability_i$ can be computed as the ratio between $Available_i$ and the average of $Available_j$ for all peers P_j attached to the same supernode. The average of $Available_j$ can be computed easily by each supernode since $Available_j$ is stored at the supernode level for each peer P_j that is connected to this supernode. This mechanism works as follows:

$$Average = \frac{\sum_j Available_j}{NbrPeers}$$
if $Average > 0$
$$Availability_i = \frac{Available_i}{Average}$$
else $Availability_i = 0$

Where $NbrPeers$ is the number of peers attached to the supernode Sup_i. Note that $Availability_i$ of peer P_i is computed based on the average of availability for all peers that belong to the same supernode. The goal is to create a competitive environment that will push peers to continuously being available for providing files to receive benefits from the system.

In [13], it has been found that most of the shared content in Gnutella is provided by only 30% of peers which means that 70% of peers are free riders. For each supernode, we can expect to have almost the same distribution. Free riders will have to be available to cope with the high availability of contributor peers to receive services from this supernode.

A peer can achieve a high *Availability* value by accepting to share files with others, and being available for uploads during long periods of time. The greater the number of files this peer is sharing, the greater its *Availability* value will be.

B. Peer Involvement

We call the contribution value defined in [11] $Involvement_i$ of peer P_i and it is defined as:

$$\begin{array}{ll} Involvement_i = \frac{D^+_{*,i} - D^-_{*,i}}{D^+_{i,*} + D^-_{i,*}} & \text{if } D^+_{i,*} + D^-_{i,*} \neq 0 \\ Involvement_i = D^+_{*,i} - D^-_{*,i} & \text{otherwise} \end{array} \quad (2)$$

The $Involvement_i$ of peer P_i is the ratio between what the peer has positively uploaded to the system and what it has downloaded from it. The term $D^+_{*,i} - D^-_{*,i}$ means that the $Involvement_i$ value is sensitive to peer's maliciousness. This term allows to affect both free riders and malicious peers since it will be very low for free riders and maybe negative for malicious peers. Peers that download much more than they upload to other peers will get a low *Involvement* value. Thus, peers have to continuously upload files if they want to receive files from others.

In [11], we argued that peer's reputation is not a good indicator to use in service differentiation since free riders can get easily a high reputation value by uploading few files and after that these peers will stop uploading files. Peer's contribution is a good indicator of the real peer behavior in terms of taking advantage of the system and contributing to it.

C. Peer Contribution

The *Contribution Behavior* CTB_i of peer P_i is computed as follows:

$a = Min\{Availability_i, 1\}$
if $Involvement_i < 0$
  $b = -1$
else $b = Min\{Involvement_i, 1\}$
$CTB_i = Max\{a + \frac{b}{2}, 0\}$
$CTB_i = Min\{CTB_i, 1\}$

The value of CTB_i can also be computed based on a weighted sum of a and b: $CTB_i = \alpha a + \beta b$, (with $\alpha \geq 0$ and $\beta \geq 0$). α and β are application dependent and represent the weights given to *Availability* and *Involvement* of peer P_i. In this paper, $\alpha = 1$ and $\beta = 1/2$ since *Involvement* is more important than *Availability*.

Note that in case that $Availability_i \geq 1$, a is set to 1 which means that the peer is available more than the average availability of all peers that belong to its supernode Sup_i. In case that $Involvement_i \geq 1$, b is set to 1 which means that the peer is contributing to the system more than what it is downloading from it.

The greater CTB_i value is, the more likely peer P_i will benefit from the system. Even if a peer did not get a chance to upload a file, it can still have its requests handled by the system based on its *Availability*.

It is important also to notice that if peer P_i is contributing negatively by uploading malicious files, this peer will get a negative $Involvement_i$ value which will reduce its contribution value, and hence its probability to benefit from the system although this peer may have a high $Availability_i$ value. The proposed mechanism allows to reduce significantly services provided to malicious contributor peers that harm the system by providing corrupted content.

IV. Service Differentiation

When reputation is used as a guideline for service differentiation, a free rider can increase its reputation by uploading authentic files until it reaches a high reputation value. Then, this peer can just stop sharing and uploading files. This *milking* process will be useful for the peer for a long period. This peer will have no need to upload any more files. The use of reputation as a criterion for service differentiation is not adequate when reputation is computed based only on satisfied and/or unsatisfied uploads because peers can have the same reputation regarding their *Authentic Behavior* but without downloading at similar levels. In this case, the *shadow of the future* as discussed in Section I-A is not reflected in peers' reputation.

The *shadow of the future* can be enforced if the *Contribution Behavior* dimension is taken into account. Only peers that contribute to the system receive services. Thus, peers are forced to increase their contribution values to receive better services (e.g., higher priority/probability of performed requests).

In this paper, we enforce service differentiation policies at the supernode level[1]. When peer P_i sends a request to its supernode $Sup(i)$, this latter will associate to the request a probability $prob_i$ according to the contribution level of peer P_i. $Sup(i)$ will perform the requested service with this probability. The higher the contribution level, the more likely the supernode will execute the requests for this peer[2].

When supernode Sup_i receives a request for searching a file on behalf of peer P_i for example, Sup_i will compute the probability $prob_i$ for peer P_i, and will execute the request according to this probability. This probability $prob_i$ is computed as follows:

if $D_{i,*} \leq MinDownload$
  $prob_i = 1$
else $prob_i = CTB_i$

New comers to the system are entitled to download up to a minimum amount set to $MinDownload$. The probability $prob_i$, used by the supernode in this case, is equal to 1 to allow new comers (i.e., with no involvement) to download files. After exceeding this minimum amount of downloads, the probability used by the supernode will be computed according to *Contribution Behavior* CTB_i that is based on peer's $Availability_i$ and $Involvement_i$.

V. Rational Behavior

Rational Behavior for peers has been introduced and explained for completely decentralized P2P systems in [14][3]. The algorithm assumes a periodical update of peer behavior in terms of probability of sharing files $ProbShare_i$.

The following values are stored by each peer P_i:

1) $SuccessfulRequest_i$: Number of requests successfully performed by Sup_i for P_i during the current evaluation period,
2) $Request_i$: Number of requests sent to supernode Sup_i by peer P_i during current evaluation period,
3) $ProbShare_i$: The probability of peer P_i to share files with other peers during current evaluation period. Free riders will have lower values of $ProbShare_i$ than contributor peers,
4) $OldBenefit_i$: Benefit obtained during previous evaluation period,
5) $LastAction_i$: The action performed on $ProbShare_i$ during previous evaluation period. The value of $ProbShare_i$ will increase or decrease and the value of $LastAction_i$ will be 1 or -1 respectively.

The algorithm has been modified from its original version and it works as follows:

```
At the end of each evaluation period,
if Request_i ≥ 0
  NewBenefit = SuccessfulRequest_i/Request_i
  if NewBenefit ≥ OldBenefit_i
    if LastAction_i == 1
      ProbShare_i = ProbShare_i + increment
      ProbShare_i = min(ProbShare_i, 1)
    else
      ProbShare_i = ProbShare_i − increment
      ProbShare_i = max(ProbShare_i, 0)
  if OldBenefit_i ≥ NewBenefit
    if LastAction_i == −1
      ProbShare_i = ProbShare_i + increment
      ProbShare_i = min(ProbShare_i, 1)
      LastAction_i = 1
    else
      ProbShare_i = ProbShare_i − increment
      ProbShare_i = max(ProbShare_i, 0)
      LastAction_i = −1
  if (OldBenefit_i == NewBenefit)
    and (NewBenefit <= 0.1)
      ProbShare_i = ProbShare_i + increment
      ProbShare_i = min(ProbShare_i, 1)
      LastAction_i = 1
  OldBenefit_i = NewBenefit
  SuccessfulRequest_i = 0
  Request_i = 0
```

Rational behavior involves comparing benefits before and after the evaluation period. If the new strategy (the new value of $ProbShare_i$) is better than the old strategy, the same action as in $LastAction$ will be performed. Otherwise, the opposite of $LastAction$ will be executed.

In our algorithm, if the old benefit and the new one have low values (almost null), the peer will increase its probability of sharing files $ProbShare_i$. In the original version, this case was not treated and peers could not receive any benefits when their $ProbShare_i$ is equal to zero. The original version suffers

[1] Note that the *Contribution Behavior* can be used to enforce service differentiation at any level (i.e., supernode or peer)

[2] To prevent peers from repeatedly sending the same request to the supernode over and over until the request is handled, a time period can be associated with each request. This will force peers to contribute if they want their requests to be processed by the system.

[3] Although this algorithm was proposed for completely decentralized P2P systems, it can be used in a partially decentralized P2P system

Category	Percentage	Probability of sending inauthentic files Malicious (30%)	Non malicious (70%)
Contributors	30%	0.9	0.01
Free Riders	70%	0.9	0.01

TABLE I

PEERS' BEHAVIOR AND DISTRIBUTION

from a deadlock and peers could not change their behavior to receive better services. Moreover, in our algorithm, peers can evaluate their benefits from the system at different periods of time instead of making this evaluation in a synchronous way as noticed in the original version [14].

VI. PERFORMANCE EVALUATION

A. Simulation Parameters

We use the following simulation parameters:

- We simulate a system with 500 peers and 500 files.
- File sizes are uniformly distributed between 10MB and 150MB.
- At the beginning of the simulation, each peer has at most 15 randomly chosen files and each file has at least one owner.
- As observed by [15], KaZaA files' requests do not follow the Zipf's law distribution. In our simulations, file requests follow the real life distribution observed in [15]. Each peer can ask for a file with a Zipf distribution over all the files that the peer does not already have. The Zipf distribution parameter is chosen close to 1
- Peers are divided into two categories: Contributors and Free Riders. Free riders constitute 70% of peers. From each category, 30% of peers are malicious peers that send inauthentic content. Peers' behavior and distribution are summarized in table I.
- To assess the performance of the considered schemes in a highly dynamic environment, only 40% of all peers with the requested file are found in each search request since only partial search results are obtained in partially decentralized P2P systems.
- Initially, free riders share files with a null probability and contributor peers with a probability equal to 1.
- $MinDownload$ is set to the average file size.
- The probability of sharing ($ProbShare$) is increased or decreased by a parameter set to 0.2.
- We simulate 150000 requests.

In this paper, we do not consider peers that lie in the feedback. This issue was addressed in [12].

According to table I, peers with indices from 1 to 350 belong to the category of free riders (FR), peers with indices from 351 to 500 belong to the category of contributor peers (CP). Accordingly, peers with indices from 1 to 245 are good free riders (GFR) and peers with indices from 246 to 350 are malicious peers in addition of being free riders (MFR). Peers with indices from 351 to 395 are malicious contributor peers (MCP) that provide malicious content but still participate in uploading files to other peers. Peers with indices from 396 to 500 are good contributor peers (GCP). We have considered a situation where we have a high percentage of free riders as observed by [13] to show the effectiveness of our proposed scheme in identifying and handling free riders (good and malicious).

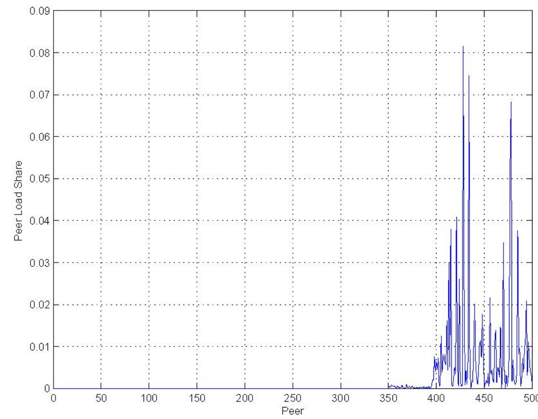

Fig. 3. Percentage of Performed Requests with Contribution Behavior based on Availability and Involvement using Static Behavior

B. Simulation Results

First, we assume that peers do not use rational behavior (i.e, static behavior). This behavior is considered to show that using service differentiation, services are received according to the contribution behavior of peers. In case that there is no service differentiation, all peers categories (i.e, GCP, MCP, GFR, and MFR) will receive the same level of service. Obviously, it is unfair that free riders (GFR and MFR) and malicious contributor peers (MGP) benefit from the system even if they are not supporting the same load as good contributor peers (GCP) do.

Figure 3 depicts the normalized load supported by different peers in a contribution-based service differentiation after 150000 requests sent to the system. Since the probability of sharing for GCP is equal to 1, these peers are supporting almost all the load. Free riders do not share any files since their probability of sharing is null. Malicious peers are detected and identified by the reputation scheme and are not selected to upload files, preventing the peers from receiving malicious content.

Figure 4 depicts the percentage of performed requests (i.e. accepted requests by the supernode) for different categories of peers after 150000 requests sent to the system. Since GCP are contributing positively, they are rewarded with a high level of service. Both free riders and malicious contributor peers receive a very low level of service since their contribution values are very low. Indeed, these peers are not involved in uploading files nor available to share files and hence, their Availability and Involvement values are very low.

In the following set of simulations, we assume that peers use rational behavior. The goal is to show that using the rational behavior, free riders will change their behavior from free riding

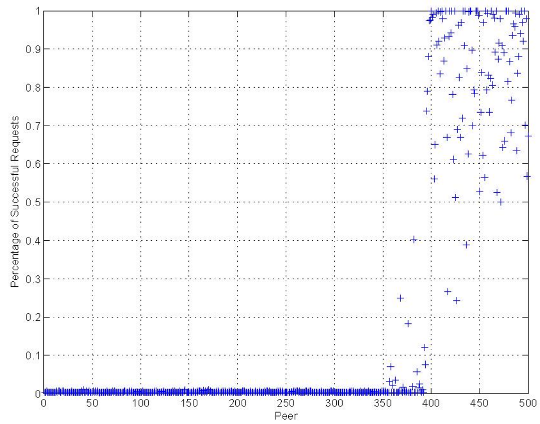

Fig. 4. Percentage of Performed Requests with Contribution Behavior based on Availability and Involvement using Static Behavior

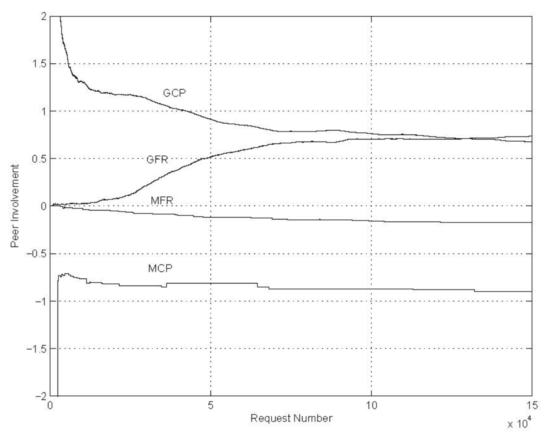

Fig. 5. Peer Involvement

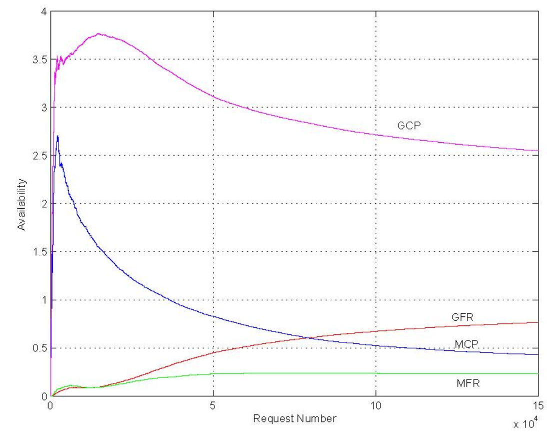

Fig. 6. Peer Availability

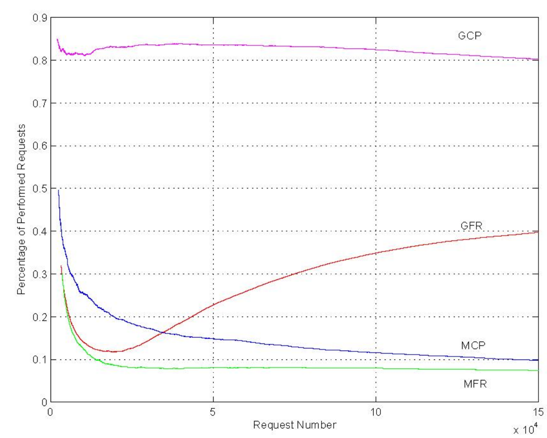

Fig. 7. Percentage of Performed Requests with Contribution Behavior based on Availability and Involvement

to sharing and uploading files. As in real life, peers will tend to change their behavior to maximize the benefit obtained from the system.

Figure 5 shows the average peer involvement for different categories of peers. The X axis represents the number of requests while the Y axis represents the average peer involvement. At the beginning of the simulation, the involvement of free riders is very low since they are not sharing any files. As their probability of sharing increases, good free riders (GFR) get more involved in the system by uploading files until they reach a similar value as good contributor peers (GCP). The average peer involvement for good contributor peers decreases gradually since they are uploading less than before due to the fact that good free riders are becoming more involved in the system. As a consequence, GCP are released from supporting a high load, reducing the amount of resources dedicated by those peers to the system. This is considered an additional benefit received by GCP in addition to receiving higher services as will be shown in figure 7. However, malicious peers (both MFR and MCP) have negative involvement values since they are uploading more malicious content.

Figure 6 shows the average peer availability for different categories of peers. The X axis represents the number of requests and the Y axis represents the average peer availability. At the beginning of the simulations, the availability of free riders is null since their probability of sharing files is null. As this probability of sharing of good free riders increases, the availability of these peers also increases. Hence, their contribution increases and also the amount of received services. During the beginning of the simulation, the availability of good contributor peers (GCP) increases as they are the only ones available to upload files. However, the availability of malicious contributor peers (MCP) decreases. Using the contribution behavior as a guideline for service differentiation, these peers get less services, hence they will not be able to download as many files as good contributor peers.

We want to investigate the impact of Availability on the percentage of performed requests (i.e. accepted requests by the supernode) for free riders and contributor peers. Figure 7 shows the results obtained in the case where the Contribution Behavior is based on both peers' Availability and Involvement. Figure 8 shows the results in the case where the Contribution

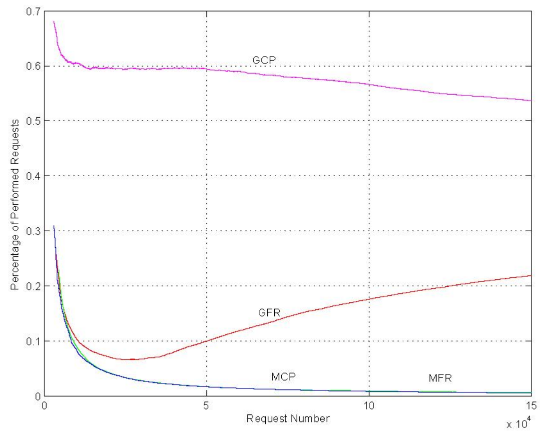

Fig. 8. Percentage of Performed Requests with Contribution Behavior based only on Involvement

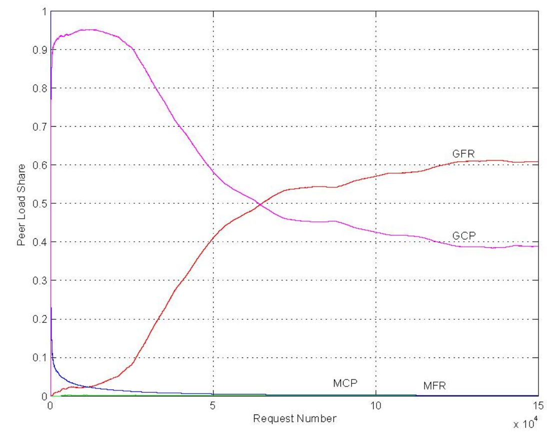

Fig. 9. Peer Load Share with Contribution Behavior based on Availability and Involvement

Behavior is computed based on peers' Involvement only.

Figure 7 shows that at the beginning of the simulation, only 30% of free riders' requests are performed by the system. This is thanks to the minimum amount of downloads $MinDownload$ they are authorized to have. This percentage will decrease gradually until free riders do not receive any significant benefit from the system due to their low contribution as explained earlier. This will push these peers to change their behavior and start sharing files with others. As the probability of sharing of good free riders (GFR) increases, so does the benefit they receive from the system. Malicious contributor peers (MCP) and malicious free riders (MFR) have a very low percentage since their involvement is negative. Good contributor peers (GCP) get a high percentage of accepted requests since they have a high contribution value due to their high availability and high positive involvement.

Figure 8 shows the results in the case where the Contribution Behavior is computed based on peers' Involvement only. This figure shows that good free riders (GFR) receive a lower level of service compared to the previous case (c.f. figure 7). Also, good contributor peers (GCP) receive a lower percentage of performed requests in figure 8 compared to the percentage received by these peers in figure 7. Using peers Availability and Involvement to compute the contribution behavior will reward better GCP and GFR. Note that in this case, both MCP and MFR also receive a slightly better service as shown in figure 7. Although, these peers do not deserve any benefit from the system, our new scheme provides them with an opportunity to receive services and change their behavior from acting maliciously to contributing positively to the system. Using the new scheme, these peers can slowly download good quality files and be able to upload them increasing their contribution and hence, their reputation.

We also want to investigate the impact of Availability on the normalized load supported by different categories of peers. Figure 9 shows the normalized load in the case where the Contribution Behavior is computed based on both peers Availability and Involvement. Figure 10 shows the load in the case where the Contribution Behavior is computed using only peers Involvement.

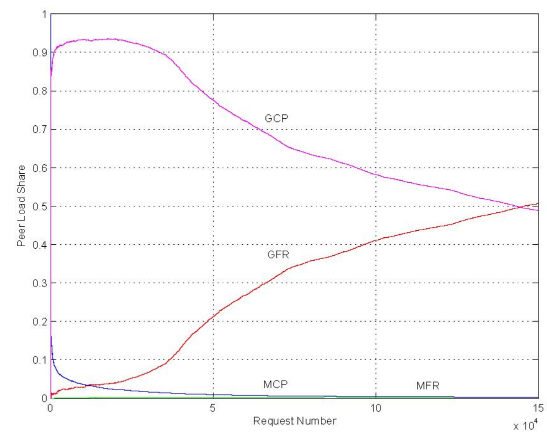

Fig. 10. Peer Load Share with Contribution Behavior based only on Involvement

Figure 9 shows that at the beginning of the simulation, since the probability of sharing for free riders is null, they were not participating in uploading files and all the load was exclusively supported by good contributor peers (GCP). Note that malicious contributor peers (MCP) are detected very quickly by the system and are isolated (i.e. not requested for uploads). Using our proposed scheme for service differentiation and with rational behavior, good free riders (GFR) are forced to share and upload files to get high level of service. Good contributor peers (GCP) are rewarded by the reduction of the supported load since good free riders are now uploading files. As to malicious peers (both MFR and MCP), they are not participating in uploading files since our proposed scheme is able to identify and isolate them.

As shown in Figure 10, using the Contribution Behavior based only on peers *Involvement* does not motivate free riders to share files in the same manner as shown in figure 9. In figure 10, GFR will need more time (150000 requests) to support equally the load with GCP. In figure 9, GFR will start supporting the load equally with GCP only after 60000 requests.

In summary, free riders change their behavior and start participating positively to the system. The new scheme provides the right incentives and opportunity to help free riders to start sharing. The new scheme successfully achieves the objectives described in section I-A.

VII. Related Work

The authors in [16] proposed a service differentiation protocol (SDP) for completely decentralized unstructured P2P networks. This protocol works as follows:

- During the *search* phase, a peer sends its *reputation score* along with the Query message. Each peer that receives this query extracts the reputation score and maps this value to a Level of Service (LoS). This peer will provide service to the requester peer according to this level.
- During the *content download* phase, the peer requesting the file sends its reputation score to the peer uploading the requested file. This latter will send the file with a rate of transfer according to the *reputation score* of the requesting peer.

This scheme is suitable for completely decentralized P2P systems, but not for partially decentralized systems. Furthermore, maliciousness of peers is not taken into consideration which restricts the use of this scheme.

In [17], the authors introduce a reputation-based mechanism that assigns better service to higher performing peers. The reputation is classified into two categories: provider selection and contention resolution. In provider selection, a peer among peers offering a service is chosen to provide the service. In contention resolution, a peer among peers requesting a service is selected by the provider peer. This scheme uses the reputation value as a guideline for service differentiation. In this paper, we have shown that the use of reputation for service differentiation is not adequate nor efficient. In addition, the proposed algorithm in [17] provides the requesting peer with a list of peers having similar reputation values using the concept of "Layered Communities". This approach will incur an important increase of malicious uploads. Indeed, if a peer receives a service from a lower reputable peer, it will most probably receive a bad service (e.g. malicious file) and hence does not help the peer in providing good service to others. In our scheme, we propose to provide only eligible peers with the requested service. Once the request is approved, peers will receive the service from the most reputable providers.

In [18], the authors analyze the effectiveness of different incentives mechanisms to motivate peers to share files. The paper presents a *reputation-based peer-approved* scheme. The scheme uses a reputation mechanism based on rating peers according to the number of files they are advertising. Peers are allowed to download files only from peers with lower or equal rating. The results show that the scheme can be used to counter the selfish behavior. However, rating peers according to the number of files they are advertising is not efficient. For example, malicious peers can advertise a high number of malicious files. According to the proposed scheme, these peers will still receive good services. Even non malicious peers may advertise a large number of useless files and still benefit from the system. In our scheme, once malicious peers are detected, these peers will not receive good services as good contributor peers.

KaZaA Media Desktop (KMD) a proprietary partially-decentralized P2P system, has introduced a *Participation Level* for rating peers. Priority is given to peers with high *participation level*, however the exact process of how this priority is given is not known. In KaZaA, malicious peers will still have a high value of *participation level* even if their participation is affecting badly other peers since there are uploading corrupted content. In our scheme, malicious peers will be detected, and punished by receiving lower services than good peers.

VIII. Conclusion

In this paper, we propose a novel contribution management scheme for partially decentralized peer-to-peer systems. The peer's *Contribution Behavior* is computed based on both *Availability* and *Involvement* of the peer. The *Contribution Behavior* is used as a guideline for service differentiation. This new scheme provides the right incentives for free riders to share files. Using rational behavior in the performance evaluation, it is clearly shown that good free riders tend to increase their contribution to benefit from better services, reducing the load supported by good contributor peers. Moreover, thanks to the generated competitive environment, peers will be forced to continuously participate to benefit from the system reducing significantly the milking phenomenon.

References

[1] D. Gambetta, "Can we trust trust?," in *Trust: Making and Breaking Cooperative Relations*, chapter 13, pp. 213–237. Published Online, 2000.

[2] S. Marsh, *Formalising Trust as a Computational Concept*, Ph.D. thesis, University of Stirling, 1994.

[3] K. Aberer and Z. Despotovic, "Managing Trust in a Peer-2-Peer Information System," in *The 9th International Conference on Information and Knowledge Management*, 2001.

[4] F. Cornelli, E. Damiani, S. De Capitani di Vimercati, S. Paraboschi, and P. Samarati, "Choosing Reputable Servents in a P2P Network," in *The 11th International World Wide Web Conference*, 2002.

[5] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina, "The EigenTrust Algorithm for Reputation Management in P2P Networks," in *The 12th International World Wide Web Conference*, 2003.

[6] M. Gupta, P. Judge, and M. Ammar, "A Reputation System for Peer-to-Peer Networks," in *ACM 13th International Workshop on Network and Operating Systems Support for Digital Audio and Video*, 2003.

[7] L. Mekouar, Y. Iraqi, and R. Boutaba, "Peer-to-Peer Most Wanted: Malicious Peers," *to appear in the Computer Networks Journal: Special Issue on Management in Peer-to-Peer Systems: Trust, Reputation and Security*, 2005.

[8] R. Axelrod, *The Evolution of Cooperation*, Basic Books, New York, 1984.

[9] L. Xiong and L. Liu, "Peertrust: Supporting reputation-based trust for peer-to-peer electronic communities," *IEEE Transactions on Knowledge and Data Engineering*, vol. 16, no. 7, pp. 843–857, 2004.

[10] S. Marti and H. Garcia-Molina, "Limited Reputation Sharing in P2P Systems," in *ACM Conference on Electronic Commerce*, New York, USA, May 2004, pp. 91–101.

[11] L. Mekouar, Y. Iraqi, and R. Boutaba, "Handling Free Riders in Peer-to-Peer Networks," in *to appear in Proceeding of the Fourth International Workshop on Agents and Peer-to-Peer Computing (AP2PC)*, 2005.

[12] L. Mekouar, Y. Iraqi, and R. Boutaba, "Detecting Malicious Peers in A Reputation-Based Peer-to-Peer System," in *The IEEE Consumer Communications and Networking Conference (CCNC)*, 2005.

[13] E. Adar and B. A. Huberman, " Free Riding on Gnutella," Tech. Rep., HP, 2000, http://www.hpl.hp.com/research/idl/papers/gnutella/.

[14] T. G. Papaioannou and G. D. Stamoulis, "Reputation-based policies that provide the right incentives in peer-to-peer environments," *to appear in the Computer Networks Journal: Special Issue on Management in Peer-to-Peer Systems: Trust, Reputation and Security*, 2005.

[15] K. Gummadi, R. J. Dunn, S. Saroiu, S. D. Gribble, H. M. Levy, and J. Zahorjan, "Measurement, Modeling, and analysis of a Peer-to-Peer File Sharing Workload," in *The 19th ACM Symposium on Operating Systems Principles*, 2003.

[16] M. Gupta and M. Ammar, "Service Differentiation in Peer-to-Peer Networks Utilizing Reputations," in *ACM Fifth International Workshop on Networked Group Communications*, 2003.

[17] T. G. Papaioannou and G. D. Stamoulis, "Effective use of reputation in peer-to-peer environments," in *Proceedings of IEEE/ACM CCGrid: International Symposium on Cluster Computing and the Grid*, 2004.

[18] K. Ranganathan, M. Ripeanu, A. Sarin, and I. Foster, "Incentive mechanisms for large collaborative resource sharing," in *Proceedings of IEEE/ACM CCGrid: International Symposium on Cluster Computing and the Grid*, 2004.