Peer-to-Peer Netw Appl (2009) 2:146–163
DOI 10.1007/s12083-008-0026-2

A contribution-based service differentiation scheme for peer-to-peer systems

Loubna Mekouar · Youssef Iraqi · Raouf Boutaba

Received: 10 June 2008 / Accepted: 30 December 2008 / Published online: 3 February 2009

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 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 should be also motivated to contribute positively by uploading authentic files instead of malicious ones. Service differentiation is required to motivate peers to get involved by sharing and uploading the requested files. To provide the right incentives for peers to contribute to the system, the new concept of *Contribution Behavior* is introduced for partially decentralized peer-to-peer systems. In this paper, the *Contribution Behavior* of the peer is used as a guideline for service differentiation instead of peer's reputation. Both *Availability* and *Involvement* of the peer are used to assess its Contribution Behavior. Performance evaluations 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. On the other hand, good peers receive better service. Simulation results also confirm that based on a *Rational Behavior*, peers are motivated to increase their contribution to receive services. Moreover, using our scheme, peers must continuously participate, reducing significantly the *milking* phenomenon.

Keywords Trust · Reputation · Contribution · Availability · Involvement · Peer-to-peer systems

1 Introduction

In Peer-to-peer (P2P) file sharing systems, peers communicate directly with each other to exchange information and share files. Peers often have to interact with strangers peers and need to manage the risk involved in these interactions. For example, if a user wants to download a file, the user is given a list of peers that can provide the requested file. The user has then to choose one peer from which the download will be performed. The open and anonymous nature of Peer-to-Peer systems open the door to misuses (by malicious peers) and abuses (by free riders[1]). Dealing with untrustworthy

Preliminary versions were published in the Fourth International Workshop on Agents and Peer-to-Peer Computing (AP2PC), 2005, and in the IEEE International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom), 2005.

L. Mekouar (✉) · R. Boutaba
David R. Cheriton School of Computer Science,
University of Waterloo, 200 University Avenue West,
Waterloo, Ontario, N2L 3G1, Canada
e-mail: lmekouar@bbcr.uwaterloo.ca

R. Boutaba
e-mail: rboutaba@bbcr.uwaterloo.ca

Y. Iraqi
Department of Computer Science, Dhofar University,
P.O. Box 2509, 211 Salalah, Oman
e-mail: y_iraqi@du.edu.om

[1] Free riders are peers that take advantage of the system without contributing to it or with a very small contribution.

peers increases peers' frustration and disappointment. Trust is needed to achieve better cooperation among peers and maximize peers' satisfaction.

Trust management is a mechanism that allows to establish mutual trust which will motivate peers to cooperate. Building trust is difficult especially when we are dealing with strangers in virtual communities where risk is involved. Marsh [14] 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.

Diego Gambetta defines trust as follows [7]: "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".

Reputation can be used to measure the trustworthiness of entities. Reputation has been widely used in different disciplines such as psychology, sociology, economics. From the Oxford dictionary, Reputation is *what is generally said or believed about a person's or thing's character or standing*. In service-oriented environments, [5] defines reputation as "an aggregation of the recommendations from all of the third-party recommendations agents and their first, second and third hand opinions as well as the trustworthiness of the recommendation agent in giving correct recommendations to the trusting agent about the quality of the trusted agent". The same definition holds for the quality of service and the quality of product. Several reputation-based systems [2, 5, 6, 10, 12, 17, 19, 25] were proposed to build trust by using peer reputation values as selection criteria to distinguish between malicious and non-malicious peers. A peer's reputation is based on its past interactions with other peers and it represents its reliability in sending authentic files. Over time, peers get a good estimation of each other's real behavior.

1.1 Motivation

Most reputation management schemes try to achieve the following goals:

1. Isolate malicious peers from the network by downloading files from the reputable peers, hence reducing malicious uploads
2. Increase the users' satisfaction
3. Use the network resources more efficiently

However, mechanisms for providing incentives and service differentiation are needed to achieve the following goals:

1. Motivate peers to share files and contribute to the system
2. Reward the reputable peers by providing better services to them and punish malicious peers

Reputation systems combined with service differentiation are needed to achieve better cooperation among peers in P2P systems.

Peers need to be 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* [4]. For example, in the case of the eBay reputation system, members have interest to get a high reputation value and maintain a good history of transactions as members with high reputation values are more trusted and selected for commercial transactions. For example, the higher is the reputation of a seller, the higher is the chance that buyers will trust to deal with him.

In a P2P file sharing system, the situation is different. What is the benefit that a peer can gain from having a high reputation value? This peer will be more and more requested for uploads which is not a gain for this peer, but more for the peers that download from it. In P2P systems, 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. This is why service differentiation is needed.

Some of the reputation-based P2P systems that use the number of satisfied and/or unsatisfied transactions as a basis for computing the reputation of a peer [6, 12, 15], considered peers' reputation as a guideline for service differentiation. This means that a peer with a high reputation, will receive better service than a peer with a lower reputation. This however does not address the problem of free riders. For example, a free rider may upload few authentic files and get a high reputation. Then, the free rider starts taking advantage of the system thanks to its high reputation. In the literature, this phenomenon is called "milking". If the reputation is used as a guideline for service differentiation, then free riders will also receive the same service as the good participating peers. Using reputation for service differentiation will provide better service to high reputable peers and lower service to low reputable peers, but, will not allow detecting free riders.

1.2 Contribution of the paper

In light of the above discussion, we argue that a good scheme for service differentiation should be able to detect free riders and malicious peers and lower the service provided to them. This will have a double effect. On one hand, this will encourage free riders and malicious peers to change their behavior. And, on the other hand, good peers will receive a better service and will be motivated to continue providing good service.

In this paper, we propose for partially decentralized P2P systems a contribution management scheme that can be combined with a reputation management scheme. Peers will have to contribute to the system to receive 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. The contribution of peers rather than the reputation of peers is used as a guideline for service differentiation.

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. This is achieved by providing the right incentives for free riders to change their behavior from free riding to positively contributing to the system and punishing them in case they don't.
- Creating a competitive environment that will push peers to continuously being available to upload files.
- Allowing new comers or formerly free riders to build their reputation and increase their contribution.

The proposed contribution scheme along with the reputation and the credibility schemes already proposed in [17] present a framework that addresses major issues related to peers behavior in partially decentralized P2P systems.

The paper is organized as follows. Section 2 presents the general framework for trust management considered in this paper. Section 3 describes how contribution is computed. Section 4 describes the correlation between trust components and the trust data, while Section 5 describes the service differentiation strategy proposed in this paper. Section 6 presents the rational behavior adopted by peers and Section 7 presents the performance evaluation of the proposed scheme. Section 8 lists related work and finally, Section 9 concludes the paper.

2 Trust management

In this paper, we consider partially decentralized P2P systems that are the most popular. In partially decentralized P2P file sharing 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.

2.1 Notations and assumptions

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

1. Let P_i denotes peer i
2. Let $D^+_{i,*}$ denotes the satisfied downloads of peer P_i from other peers,
3. Let $D^-_{i,*}$ denotes the unsatisfied downloads of peer P_i from other peers,
4. Let $D^+_{*,i}$ denotes the satisfied uploads from peer P_i to other peers,
5. Let $D^-_{*,i}$ denotes the unsatisfied uploads from peer P_i to other peers
6. Let $A^F_{i,j}$ be the appreciation of peer P_i after downloading file F from P_j
7. Let Sup(i) denotes the supernode of peer P_i

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 the appreciation $A^F_{i,j} = 1$. If not, P_i sets $A^F_{i,j} = -1$. 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.

2.2 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 honest feedbacks. 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.

We believe that trust in a peer-to-peer system should be addressed according to the following dimensions: (1) Authentic Behavior, (2) Credibility Behavior, and (3) Contribution Behavior

Authentic Behavior (AB): this is the reliability of a peer in providing accurate and good quality files. Good peers have usually a high *authentic behavior* value, while malicious peers usually get lower values since they are providing malicious content. This value represents the reputation of a peer. It allows to differentiate between good and malicious peers. In all other works, what researchers call *reputation* is in fact the Authentic Behavior of the peer which represents only one dimension of the trust associated with the peer.

Credibility Behavior (CB): this represents the sincerity of a peer in providing a honest feedback. The *credibility behavior* is an important indicator that allows to identify liar peers and reduce their effect on the reputation system. In [17], the concept of *Suspicious Transaction* was introduced to compute the *credibility behavior*. The credibility behavior allows to identify liar peers and represents peers' credibility and sincerity.

Contribution Behavior (CTB): we introduce in this paper, the new concept of *contribution behavior* that allows to distinguish between peers that contribute positively[2] to the system (i.e. altruistic) and the free riders (i.e. egoistic).

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 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. Therefore, we need to recognize peers that are available to upload files and reward them. 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 in this paper that the Contribution Behavior of peers should be based on:

- Peers' *Availability*: being available for uploading requested files.
- Peers' *Involvement*: non-malicious uploads performed versus downloads received by a peer.

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

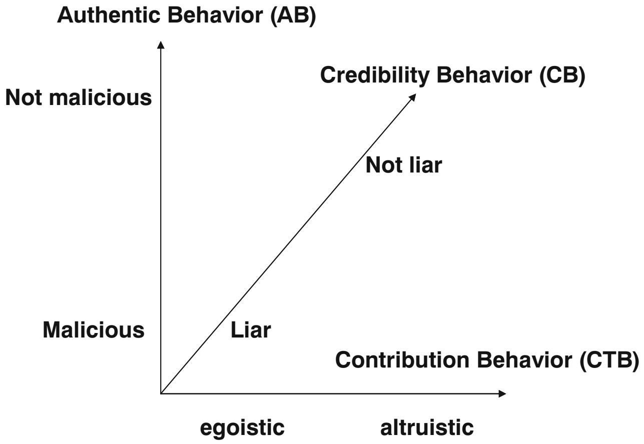

Fig. 1 Trust dimensions

An investigation about why we have to include Availability in addition to Involvement in computing the Contribution Behavior is presented in Section 7.3.1.

It is important to clarify that the trust given to 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 involvement) (cf. Fig. 1). 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, honest feedback (i.e. without lying), and are available to share files in addition of being effectively involved in uploading files. Good peers will have high values along the three defined dimensions.

Figure 2 shows the three dimensions of trust along with different aspects of behavior that they characterize.

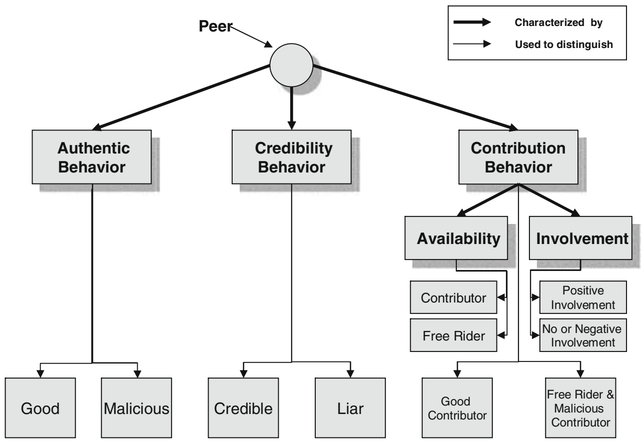

Fig. 2 Peer trust and behavior analysis

[2] We do not consider uploading malicious content as a contribution. Only authentic uploads are taken into consideration.

2.3 Credibility behavior

In this section, we describe briefly the Credibility Behavior considered in this paper. More details are provided in [17].

To detect peers that lie in their feedbacks, we use the concept of *suspicious transaction*. A *Suspicious transaction* is defined as a transaction in which the feedback is different from the one expected knowing the reputation of the peer uploading the file. This means, if $A^F_{i,j} = 1$ and $\mathrm{AB}_j < 0$ or if $A^F_{i,j} = -1$ and $\mathrm{AB}_j > 0$ then we consider this transaction as suspicious.

For peer P_i, the following information is required:

1. N_i: The total number of downloads performed by peer P_i
2. N^*_i: The number of downloads by peer P_i where the sign of the appreciation sent by peer P_i is different from the sign of the sender's reputation, i.e. $A^F_{i,j} \times \mathrm{AB}_j < 0$ (i.e. during a suspicious transaction)

When receiving the appreciation (i.e. $A^F_{i,j}$) of peer P_i, its supernode Sup(i) will update the values of N_i and N^*_i as follows:

$$\begin{aligned} &N_i = N_i + 1 \\ &\text{If } (A^F_{i,j} \times \mathrm{AB}_j) < 0 \text{ then } N^*_i = N^*_i + 1 \end{aligned} \tag{1}$$

Let α_i be the ratio of N^*_i and N_i:

$$\alpha_i = \frac{N^*_i}{N_i} \tag{2}$$

α_i is the ratio of the number of suspicious feedbacks. The Credibility Behavior of peer P_i be: $\mathrm{CB}_i = 1 - \alpha_i$.

2.4 Authentic behavior

In this section, we describe briefly the Authentic Behavior considered in this paper. More details are provided in [17].

When receiving the appreciation (i.e., $A^F_{i,j}$) of peer P_i, its supernode Sup(i) will perform the following operation:

$$\begin{aligned} &\text{If } A^F_{i,j} = 1 \text{ then } D^+_{i,*} = D^+_{i,*} + \mathrm{Size}(F), \\ &\qquad\qquad \text{else } D^-_{i,*} = D^-_{i,*} + \mathrm{Size}(F). \end{aligned}$$

Then, the appreciation and α_i are sent to Sup(j) that will perform the following operation:

$$\begin{aligned} &\text{If } A^F_{i,j} = 1 \text{ then } D^+_{*,j} = D^+_{*,j} + (1-\alpha_i) \times \mathrm{Size}(F), \\ &\qquad\qquad \text{else } D^-_{*,j} = D^-_{*,j} + (1-\alpha_i) \times \mathrm{Size}(F). \end{aligned}$$

$$\mathrm{TF}_j = \mathrm{TF}_j + \mathrm{Size}(F)$$

Where Size(F) denotes the size of the file F and TF_j is the total size of all the files uploaded by P_j.

The Authentic Behavior of a peer P_j is computed as:

$$\begin{aligned} &\mathrm{AB}_j = \frac{D^+_{*,j} - D^-_{*,j}}{\mathrm{TF}_j} && \text{if } \mathrm{TF}_j \neq 0 \\ &\mathrm{AB}_j = 0 && \text{otherwise} \end{aligned} \tag{3}$$

The computed value indicates how reliable the peer P_j is in providing authentic files (i.e. its reputation).

Since liar peers will have a high value of α_i, their effect on the reputation of the peer sending the file is minimized. On the other hand, good peers will have a lower value of α_i and hence will keep having an impact of the reputation of other peers.

Colluding peers are malicious peers that send inauthentic files in addition to lying in the feedbacks sent after transactions. To increase the reputation of malicious peers, positive feedbacks are sent and to decrease the reputation of good peers, negative feedbacks are sent. The Credibility Behavior of these peers will be low and their impact on the reputation of good peers is minimized.

3 Contribution behavior

To measure the contribution of a peer based only on its uploads and downloads may lead to peer's starvation. If the system does not allow new peers or peers that are in the process of building their reputation to download files, these peers may have no or not enough files to upload to other peers. It will be difficult for these peers to increase their contribution and reputation. Recognizing peers that are available to upload requested files will help significantly these peers by allowing them to receive service, and increase their contribution and reputation gradually.

Several other features may be taken into consideration to assess peers' contribution in addition to the Availability and Involvement. These features could be peer's upload rate, peer's uptime, the diversity of files that a peer is providing, or sharing rare and hard to find files, etc. However, in our opinion the most important concepts that prove the peer's contribution to the system are its Availability in terms of being available and ready to upload the requested files and its Involvement in terms of what the peer has positively uploaded to the system compared to what it has downloaded from it.

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

3.1 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.

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.

The availability of peer P_i Availability_i is 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:

$$\text{Average} = \frac{\sum_j \text{Available}_j}{\text{NbrPeers}}$$

if $\text{Average} > 0$

$$\text{Availability}_i = \text{Min}\left(\frac{\text{Available}_i}{\text{Average}}, 1\right)$$

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. In case that $\text{Availability}_i \geq 1$, Availability_i is set to 1 which means that the peer is available more than the average availability of all peers that belong to its supernode. The Availability_i value for peer P_i will be high if P_i is a contributor peer, otherwise, this value will be low. This is because the Average value is continuously increased by contributor peers. The goal is to create a competitive environment that will push peers to continuously being available for providing files. The value of Average could also be the average value among several supernodes. This value can be easily exchanged between supernodes. If a peer P_i is available yet never solicited, this peer will not be deprived from benefiting from the system since its Availability_i value will not be null. This peer will get a chance to receive service. It is important to note that the supernode Sup_i updates the value Availability_j for all its peers periodically. The frequency of this update should not be high (e.g. after each search request) to avoid extra overhead and not too low to preserve accuracy.

In [3], 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. Assuming that peers are uniformly distributed among supernodes. We can expect to have almost the same distribution for each supernode. This means that 70% of peers connected to a supernode are free riders and only 30% are contributor peers. Because of the high availability of contributor peers, free riders will have to be available in order 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 offering, the greater its Availability value will be.

3.2 Peer involvement

The Involvement_i of peer P_i is defined as:

$$\begin{aligned}
&\text{Involvement}_i = \frac{D^+_{*,i} - D^-_{*,i}}{D^+_{i,*} + D^-_{i,*}} && \text{if } D^+_{i,*} + D^-_{i,*} \neq 0 \\
&\text{Involvement}_i = D^+_{*,i} - D^-_{*,i} && \text{otherwise} \\
&\text{Involvement}_i = \text{Min}(\text{Involvement}_i, 1)
\end{aligned} \tag{4}$$

The intuition behind Eq. 4 is as follows. While the reputation value is based only on the uploads of a peer to reflect its Authentic Behavior (cf. Eq. 3), the involvement should be based on both the uploads and the downloads of the peer to express how much the peer gave to the system compared to how much the peer took from the system. 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 affects 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 case that $\text{Involvement}_i \geq 1$, Involvement_i is set to 1 which means that the peer is contributing to the system more than what it is downloading from it.

Ideally, a peer should be charged only for its authentic downloads since it is not responsible for the malicious content that it received from other peers. However, some malicious peers may rate all their downloads as inauthentic so that these downloads will not be counted in the Involvement value. To avoid this situation, the total downloads is used for computing the Involvement value. This will also motivate peers to deal only with high reputable peers.

3.3 Peer contribution

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

$a = \text{Availability}_i$

if $\text{Involvement}_i < 0$

$\quad b = -1$

else $b = \text{Involvement}_i$

$c = \text{Max}\left(\frac{a}{2} + b\,, 0\right)$

$\text{CTB}_i = \text{Min}(c, 1)$

The value of CTB_i is based on the maximum between $\frac{a}{2} + b$ and 0 and the minimum between the obtained result and 1. This guarantees that CTB_i value will be between 0 and 1. The value of CTB_i can also be computed based on a weighted sum of a and b: $\text{CTB}_i = \alpha a + \beta b$, (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. An in depth analysis can be realized for parameter settings to achieve a better performance for the system. In this paper, we choose $\alpha = 1/2$ and $\beta = 1$ relying on the fact that Involvement is more important than Availability.

The use of Credibility Behavior in computing peers' involvement will reduce significantly the impact of colluding peers that report fake transactions among themselves to increase their Contribution Behavior value.

The main target of this paper is free riders, that represent according to some studies up to 70% of peers, and not specifically malicious peers. Our previous paper [17] handled detecting malicious peers. Based on peers' Authentic Behavior and Credibility Behavior, the Inauthentic Detector Algorithm (IDA) and the Malicious Detector Algorithm (MDA) are able to identify malicious peers and preventing them from uploading malicious content to other peers. Using the IDA, the proposed contribution management scheme has an effective capability to identify both free riders and malicious peers and reduce the level of service provided to them. This will definitely affect free riders. In addition, the resources of uploading peers will be protected from being used by both free riders and malicious peers.

4 Trust components

4.1 Relationship between trust components

Figure 3 shows a typical file request-download procedure involving the sender and receiver peers and their supernodes. The figure 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 processed 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 the feedback and the reputation of peer P_j. Indeed, if peer P_i has a low credibility (i.e. is a liar), this peer will send a wrong feedback and hence, affects the reputation of peer P_j. For example, a peer may decide not to upload a file to a peer with a low credibility value (along the Credibility Behavior dimension), since the latter peer may wrongfully send negative feedback and affect badly the reputation of the peer performing the upload. A peer may also decide not to upload a file to a peer with a low contribution value (along the Contribution Behavior dimension), since the peer requesting the upload may be a free rider.

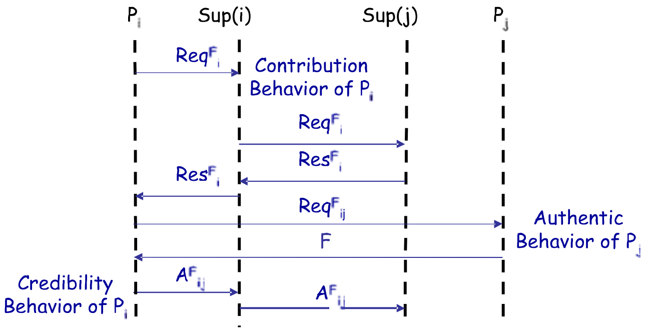
P_i
Sup(i)
Sup(j)
P_j
Req^F_i
Contribution Behavior of P_i
Req^F_i
Res^F_i
Res^F_i
Req^F_ij
Authentic Behavior of P_j
F
Credibility Behavior of P_i
A^F_ij
A^F_ij

Fig. 3 A typical exchange between peers

4.2 Trust data

Each peer P_i in the system has *Trust data* (TRUST_{P_i}), stored by its supernode $\mathrm{Sup}(i)$:

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
5. N_i: The total number of downloads performed by peer P_i
6. N^*_i: The number of suspicious transactions
7. TF_i: The total size of all the files uploaded by P_i
8. $\mathrm{Available}_i$: The number of times P_i was available to share files
9. $\mathrm{Involvement}_i$: The peer P_i uploads compared to its downloads

When peer P_i joins the system for the first time, all values of its Trust data TRUST_{P_i} are initialized to zero. Protecting the integrity of peers' Trust data is imperative to prevent malicious peers from increasing their Authentic and Contribution Behavior values and take advantage from the system. Since supernodes handle efficiently the search requests on behalf of peers, we assume that supernodes can be trusted and that they share a secret key [21]. Cryptography is used to provide authentication and integrity. To keep Trust data protected, guaranteed to be authentic and since peers are not trustful, the supernode digitally signs (TRUST_{P_i}). The supernode of peer P_i sends (TRUST_{P_i}) periodically to the peer. This period of time is not too long to preserve accuracy and not too short to avoid extra overhead. The peer will keep a copy of (TRUST_{P_i}) to be used the next time it joins the system or if its supernode changes. Hence, peers cannot tamper with Trust data.

5 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 files any more. 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 1.1 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 service (e.g., higher priority/probability of performed requests).

We divide service differentiation into two categories: *implicit* and *explicit*.

Implicit service differentiation, is the service differentiation that results from the normal evolution of the system. For example, when a peer has a low reputation (e.g. its Authentic Behavior value), this peer will have a low probability of being selected for uploads, which will not allow it to increase significantly its contribution value.

Explicit service differentiation, is the one that results from the explicit decision of system entities. For example, a supernode may decide to enforce service differentiation policies on the peers it manages. *Explicit* service differentiation can also be enforced at the level of peers. For example, a peer may decide not to upload a file to a peer with a low credibility value (along the Credibility Behavior dimension), since the latter peer may wrongfully send negative feedback and affect badly the reputation of the peer performing the upload. A peer may also decide not to upload a file to a peer with a low contribution value (along the Contribution Behavior dimension), since the peer requesting the upload may be a free rider.

In this paper we focus on enforcing service differentiation policies at the supernode level. When a peer P_i sends a request to its supernode $\mathrm{Sup}(i)$, this latter will associate to the request a probability prob_i according to the contribution level of peer P_i. This is the probability of performing the requested service by $\mathrm{Sup}(i)$. The higher the contribution value is, the more chances the supernode will execute the requests for this peer.[3]

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

[3] To prevent peers from repeatedly sending the same request to the supernode over and over until the request is handled, a minimum time period can be enforced between consecutive requests. This will motivate peers to contribute if they want their requests to be processed by the system.

execute the request according to this probability. This probability prob_i is computed as follows:

```
if $(D^+_{i,*} + D^-_{i,*}) \leq \text{MinDownload}$
  $\text{prob}_i = 1$
else $\text{prob}_i = \text{CTB}_i$
```

The greater CTB_i value is, the more likely peer P_i will receive service 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_i. 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 peers that harm the system.

New comers to the system are entitled to download up to a maximum 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 maximum amount of downloads, the probability used by the supernode will be computed according to the Contribution Behavior CTB_i.

White-washing is when a peer changes its identity and rejoins the network with a new one. This may be beneficial for malicious peers. Indeed, once these peers are identified as malicious, they will not be able to upload inauthentic files. To start over again, malicious peers can rejoin the system with a new identity and their past history will simply be forgotten. However, a new comer has a null Authentic Behavior value (AB); hence, it has a neutral trust. To be able (to be chosen by others) to upload, this peer has to increase its AB. This peer will need to contribute positively by uploading authentic files before being able to upload inauthentic ones. This will not be very efficient to malicious peers whom their main goal is to harm the system as quickly as possible. White-washing may also be beneficial for free riders. These peers could take advantage from the MinDownload value to download files without uploading to others. The value of MinDownload should be carefully chosen not to encourage peers to change identities and benefit from free downloads. A small MinDownload value will not allow newcomers to quickly benefit from the system. On the other hand, a high MinDownload value may encourage white-washing.

6 Rational behavior

Rational Behavior for peers has been introduced for completely decentralized P2P systems in [20]. The algorithm in [20] assumes a periodical update of peer behavior in terms of probability of sharing files *Prob Share*. In this paper, we adapted and modified this algorithm to fit partially decentralized P2P systems.

The following values are stored by each peer P_i:

1. *SuccessfulRequest*: Number of requests successfully performed by Sup_i for P_i during the current evaluation period,
2. *Request*: Number of requests sent to supernode Sup_i by peer P_i during current evaluation period,
3. *Prob Share*: The probability of peer P_i to share files with other peers during current evaluation period. Typically, free riders will have lower values of *Prob Share* than contributor peers,
4. *increment*: Represents the unit of increasing or decreasing the probability *Prob Share*,
5. *OldBenefit*: Benefit obtained during previous evaluation period,
6. *LastAction*: The action performed on *Prob Share* during previous evaluation period. The value of *Prob Share* will increase or decrease and the value of *LastAction* will be 1 or −1 respectively.

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

```
  At the end of each evaluation period, do
if $Request \geq 0$
  $NewBenefit = SuccessfulRequest/Request$
  if $NewBenefit > OldBenefit$
    if $LastAction == 1$
      $ProbShare = ProbShare + increment$
      $ProbShare = min(ProbShare, 1)$
    else
      $ProbShare = ProbShare - increment$
      $ProbShare = max(ProbShare, 0)$
  if $OldBenefit > NewBenefit$
    if $LastAction == -1$
      $ProbShare = ProbShare + increment$
      $ProbShare = min(ProbShare, 1)$
      $LastAction = 1$
    else
      $ProbShare = ProbShare - increment$
      $ProbShare = max(ProbShare, 0)$
      $LastAction = -1$
  if $(OldBenefit == NewBenefit)$
    and $(NewBenefit <= 0.1)$
      $ProbShare = ProbShare + increment$
    $Prob\,Share = min(Prob\,Share, 1)$
    $Last\,Action = 1$
  $Old\,Benefit = New\,Benefit$
  $Successful\,Request = 0$
  $Request = 0$
```

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 addressed and peers cannot receive any benefits when their ProbShare_i is equal to zero. The original version leads to a deadlock and peers cannot 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 it is the case in [20].

The majority of peers in the network are selfish (e.g. free riders) and they want to maximize their own utility. According to the proposed algorithm, this utility is the number of requests sent by a peer and handled successfully by its supernode. For free riders, the only way to achieve this goal is by increasing the probability of sharing files. As a result, the availability of these peers will increase and also their involvement. Thus, their contribution value will also increase. For malicious peers, they will have to upload authentic files. As a result, their involvement will become positive increasing their contribution.

7 Performance evaluation

As explained in Section 2.2, our proposed trust management is addressed according to three dimensions: Authentic Behavior, Credibility Behavior and Contribution Behavior. Our previous paper [17] focused on the Authentic Behavior and the Credibility Behavior. We have provided a thorough comparison between the Inauthentic Detector Algorithm (IDA) based on the Authentic Behavior and *KaZaA*. KaZaA is a proprietary partially decentralized P2P system that has proposed a Participation Level for the peers. This Participation Level is assigned to each user in the network based on the files that are uploaded by the user and the files the user downloads from others. The Participation Level is computed as: (Size of Uploads/Size of Downloads)*100. In the performance evaluation Section in [17], we considered the scheme where each peer uses the Participation Level as selection criterion. The performance parameters that were considered are: peers' satisfaction, percentage of malicious uploads and the distribution of load among peers. According to these simulations, our proposed reputation management scheme based on the Authentic Behavior outperforms the KaZaA-based scheme by achieving a higher peers' satisfaction, a lower percentage of malicious uploads and a better distribution of the load. KaZaA does not make any distinction between malicious and good peers. Hence, malicious peers are neither identified nor isolated from the system and can still upload inauthentic files. Consequently, peers' satisfaction is reduced significantly and network resources are wasted. The performance evaluation Section in [17] also shows the performance of the Malicious Detector Algorithm (MDA) compared to the Inauthentic Detector Algorithm (IDA) in terms of peers' satisfaction and the percentage of malicious uploads. Based on the Credibility Behavior, MDA is able to detect and identify liar peers and reduce their negative impact on the system. This allows the system to take more clear-sighted decisions which will, of course, results in using the network bandwidth more efficiently.

In this paper, we focus on the third dimension of the trust management framework which is the Contribution Behavior along with service differentiation. The greater the Contribution Behavior value for a peer, the more likely this peer will receive better services from the system. The goals from the proposed contribution management scheme are to motivate peers to share files and contribute positively to the system and also to reward good contributor peers by providing better services to them. The conducted simulations in this paper are related to the importance of Contribution Behavior as a service differentiation criterion and the impact of both Availability and Involvement.

In this section, we will simulate a system under three scenarios: no service differentiation, service differentiation with static peer behavior and service differentiation with rational peer behavior. The goal from these simulations is to show that:

- Service differentiation is important
- Service differentiation based on Contribution Behavior instead of peers' reputation identifies better free riders and reduces the services provided to these peers

- Service differentiation based on Contribution Behavior will motivate free riders to change their behavior from free riding to contributing to the system.

7.1 Simulation parameters

We use the following simulation parameters:

- We simulate a system with one supernode that supports 500 peers. We have chosen 500 peers since a supernode typically supports between 300 to 500 peers, depending on availability of resources (Gnutella2, http://www.gnutella2.com/).
- Peers share 500 files and file sizes are uniformly distributed between 10 and 150 MB.
- 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 [8], KaZaA files' requests do not follow the Zipf's law distribution. In our simulations, file requests follow the real life distribution observed in [8]. 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 the peers. From each category, 30% of peers are malicious peers that send inauthentic content. Peers' behavior and distribution are summarized in Table 1.
- MinDownload is set to the average file size.
- We simulate 150, 000 requests.

In this paper, we do not consider peers that lie in the feedback. This issue was addressed in [17].

Following Table 1, 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 [3] to show the effectiveness of our proposed scheme in identifying and handling free riders both good and malicious.

7.2 Static behavior

In this first set of simulations, we consider static peer behavior. This means that peers do not change their behavior over time. We will compare the following schemes:

1. The reputation management scheme with no service differentiation (NOSD). This is to show the importance of service differentiation among the peers.
2. The reputation management scheme with the reputation value as a guideline for service differentiation. We will call this scheme the Reputation-Based Service Differentiation (RBSD). Since the reputation values (i.e. AB_i) are between -1 and 1, in this scheme, the probability $prob_i$ is computed as follows: $prob_i = (1 + AB_i)/2$, where AB_i is computed as in Eq. 3.
3. The reputation management scheme with the Contribution Behavior as a guideline for service differentiation. We will call this scheme the Contribution-Based Service Differentiation (CBSD).

Free riders share files with a probability of 5%. In addition, 100 of the non malicious free rider peers will accept uploading the first file to get a high reputation.

In these simulations, we will focus on the following performance parameters:

- Percentage of successful requests: computed as the total number of requests that have been performed for the peer during the simulation over the total number of all submitted requests by this peer.
- Peer contribution level: shows the contribution behavior of each peer which is computed using Eq. 4.

Table 1 Peers' behavior and distribution

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

- Peer load share: this parameter is computed as the normalized load supported by the peer. This is computed as the sum of the uploads performed by the peer over the total uploads in the system.

7.2.1 No service differentiation case

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 4 depicts the normalized load supported by different peers after 150,000 requests sent to the system in the case of the $NOSD$ scheme. The X axis represents peers' id while the Y axis represents the normalized peer load share. From the figure, it is clear that the proposed reputation management scheme (Authentic Behavior) is able to detect, identify and isolate malicious peers (i.e. peer id 246 to 395), as they are not requested to upload files, preventing the peers from receiving malicious content. Since the probability of sharing for GCP is equal to 1, all the load is almost supported by non malicious contributor peers (i.e. peer id 396 to 500). Free riders do not contribute significantly to the system since they do not share any files as their probability of sharing is only 0.05. Good free riders that are using the milking strategy (i.e. peer id 1 to 100) have been participating in uploading some files to get a high reputation value.

Since there is no service differentiation, all the requests sent to the supernode will be performed regardless of the contribution of the peers. This is obviously unfair to the peers that contribute significantly to the system.

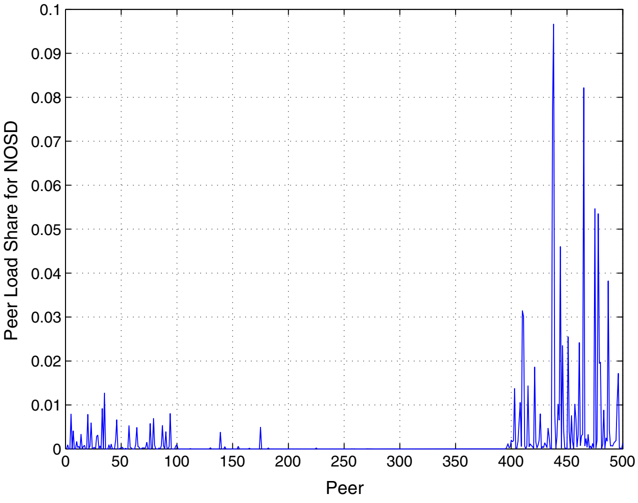

Fig. 4 Peer load share for NOSD

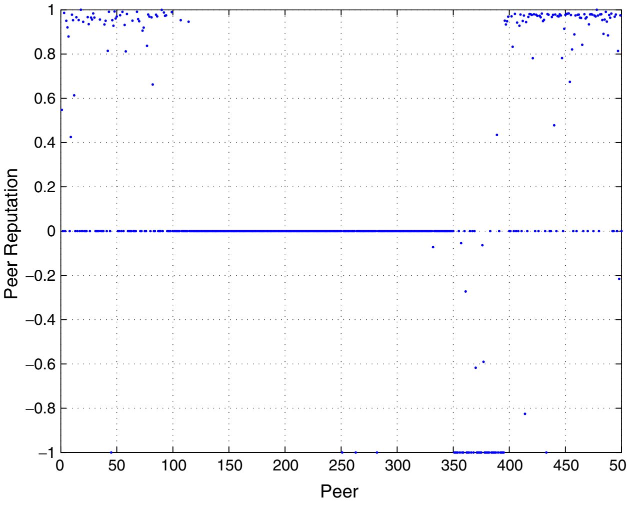

Fig. 5 Peers' reputation in RBSD

7.2.2 Service differentiation case

Figure 5 depicts the reputation values of the peers (i.e. the Authentic Behavior) in the case of the Reputation Based Service Differentiation (RBSD) scheme. It is clear that the scheme is able to identify malicious peers. However, the scheme is not able to differentiate between free riders and contributor peers. Reputation is not a good indicator of the contribution of the peer as we can see from comparing Figs. 4 and 5.

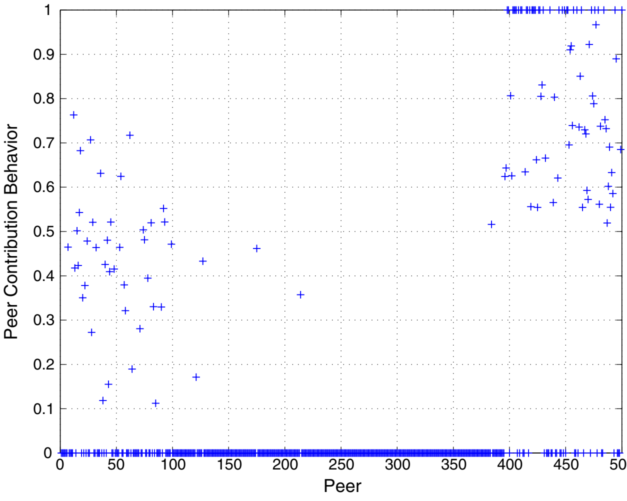

Fig. 6 Peers' contribution behavior in CBSD

Figure 6 depicts the Contribution Behavior value in the case of the Contribution Based Service Differentiation (CBSD) scheme. By comparing this figure with Fig. 4, we can notice that the Contribution Behavior value is a good indicator of the peer load share. In other words, a peer with a high contribution level is supporting more load than a peer with a low contribution level. Note that the Contribution Behavior values of malicious peers (i.e. peer id 246 to 395) are null. This is because malicious peers are harming the system by uploading malicious files. This means that the Contribution Behavior value can be used for service differentiation which will effectively reward good peers and punish both free riders and malicious peers.

Figures 7 and 8 show the percentage of successful requests for RBSD and CBSD respectively (i.e. accepted requests by the supernode). From Fig. 7, we can notice that free riders have about 50% chance to have their request processed by the supernode. Free riders with high reputation values (i.e. peer id 1 to 100) have almost the same percentage of successful requests as non malicious contributor peers. However, free riders did not contribute at the same level. In Fig. 8, free riders with id from 1 to 100, have a lower percentage of successful requests since they uploaded only few files compared to non malicious contributor peers GCP. The latter peers are rewarded with a high level of service since they have supported almost all the load. They contributed significantly and positively to the system. The supernode processed their requests with a high probability. Some of the malicious peers uploaded more malicious content than good one, hence their percentage of successful requests is very low. This is because their contribution is null as shown in Fig. 6. Also, free riders with id from 101 to 350 receive a very low level of service since their contribution values are very low (almost null). Indeed, these peers are not involved in uploading files nor available to share files and hence, their Availability and Involvement values are very low.

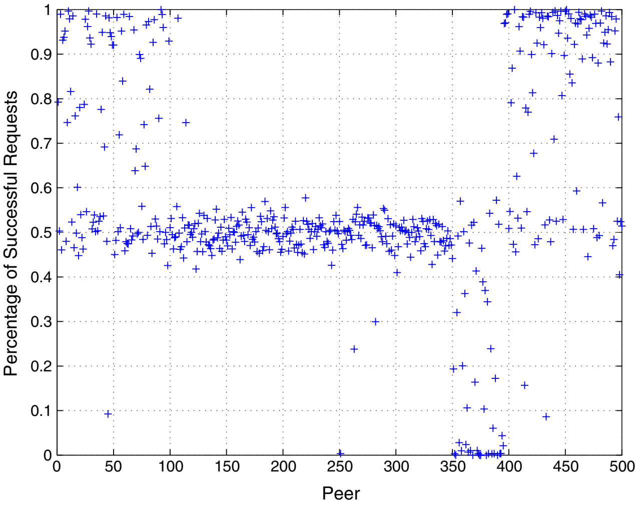

Fig. 7 Percentage of successful requests for RBSD

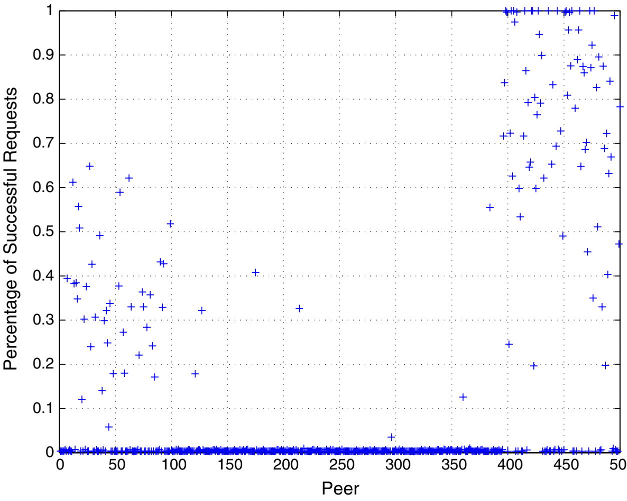

Fig. 8 Percentage of successful requests for CBSD

Note that in these simulations, we assumed a static peer behavior. This is to assess the capability of the proposed scheme in detecting malicious and free rider peers and preventing them from obtaining good service. In a real life system, however, peers will tend to change their behavior. Free rider peers with a rational behavior will change from free riding to contributing to the system.

7.3 Rational behavior

In the following set of simulations, we assume that peers use rational behavior as presented in Section 6. The goal is to show that using the rational behavior, free riders will change their behavior from free riding to sharing and uploading files. As in real life, peers will tend to change their behavior to maximize the benefit obtained from the system.

Initially, free riders share files with a null probability and contributor peers with a probability equal to 1. The probability of sharing (ProbShare) is increased or decreased by a parameter set to 0.2.

Figure 9 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

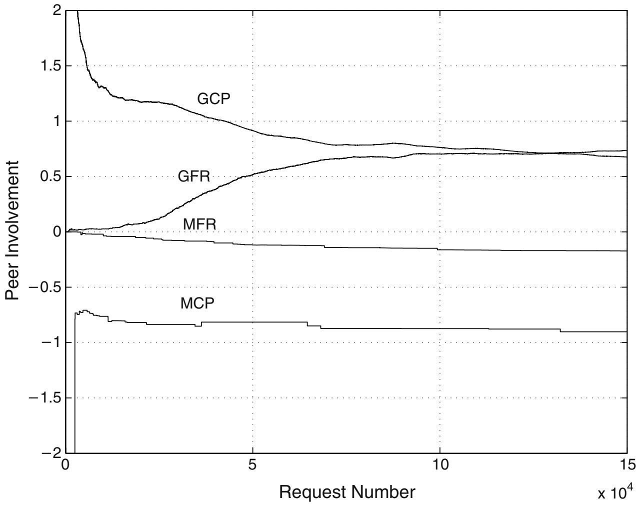

Fig. 9 Peer involvement

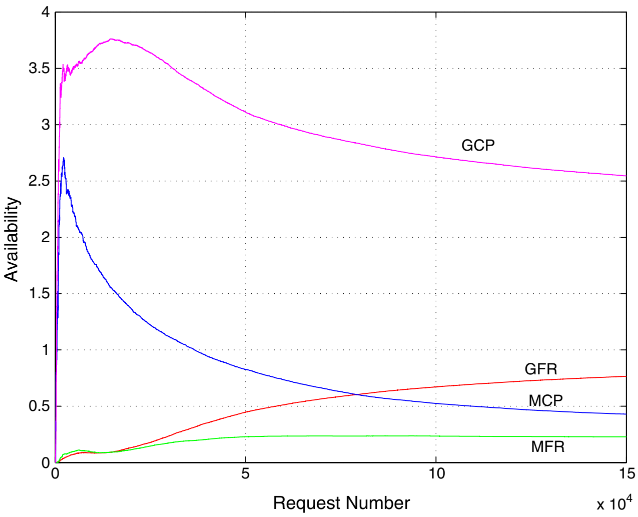

Fig. 10 Peer availability

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 Fig. 11. However, malicious peers (both MFR and MCP) have negative involvement values since they are uploading more malicious content.

Figure 10 shows the average peer availability for different categories of peers. 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.

7.3.1 Impact of availability

In this subsection, 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 11 shows the results obtained in the case where the Contribution Behavior is based on both peers' Availability and Involvement. Figure 12 shows the results in the case where the Contribution Behavior is computed based on peers' Involvement only.

Figure 11 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

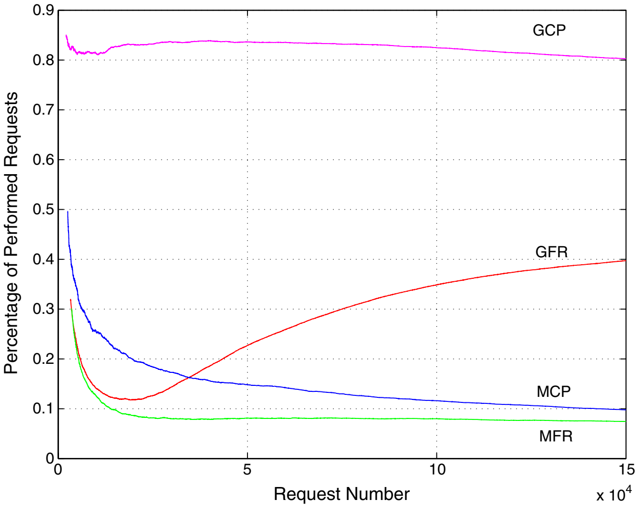

Fig. 11 Percentage of performed requests with contribution behavior based on availability and involvement

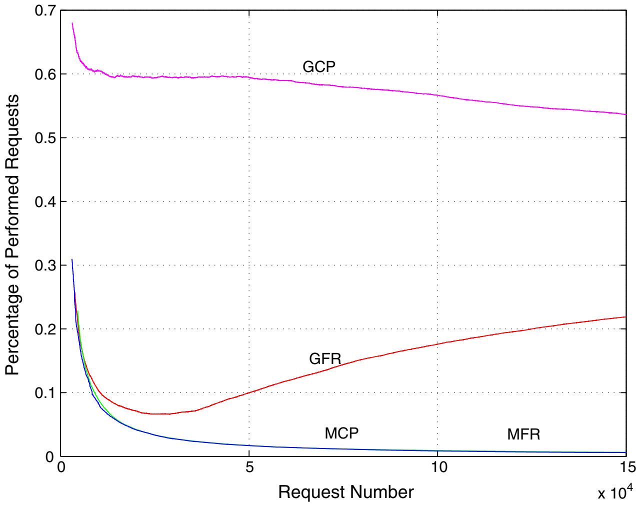

Fig. 12 Percentage of performed requests with contribution behavior based only on involvement

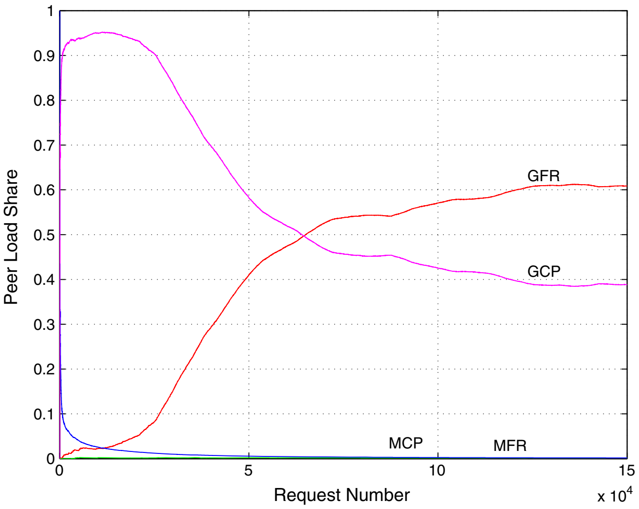

Fig. 13 Peer load share with contribution behavior based on availability and involvement

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 12 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. Fig. 11). Also, good contributor peers (GCP) receive a lower percentage of performed requests in Fig. 12 compared to the percentage received by these peers in Fig. 11. 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 Fig. 11. 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. 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 13 shows the normalized load in the case where the Contribution Behavior is computed based on both peers Availability and Involvement. Figure 14 shows the load in the case where the Contribution Behavior is computed using only peers Involvement.

Figure 13 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

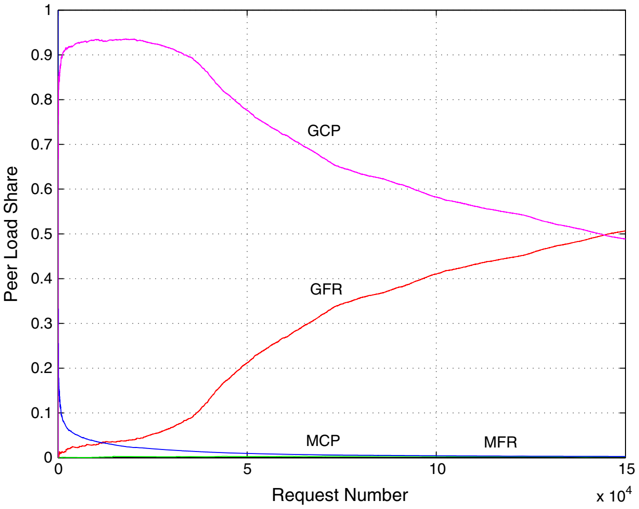

Fig. 14 Peer load share with contribution behavior based only on involvement

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 Fig. 14, using the Contribution Behavior based only on peers Involvement does not motivate free riders to share files in the same manner as shown in Fig. 13. In Fig. 14, GFR will need more time (150,000 requests) to support equally the load with GCP. In Fig. 13, GFR will start supporting the load equally with GCP only after 60,000 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 motivate free riders to start sharing. The new scheme successfully achieves the objectives described in Section 1.2.

8 Related work

The authors in [9] 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.

In [20], 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. However, this scheme uses the reputation value as a guideline for service differentiation. In addition, the proposed algorithm in [20] 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. Receiving malicious content will just pollute the P2P file sharing system and waste network’s resources.

In [22], 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, this scheme will allow malicious peers to advertise a high number of corrupted files. According to this scheme, these peers will still receive good service. Even non malicious peers may advertise a large number of non popular or useless files and still benefit from the system. In our scheme, once malicious peers are detected, these peers will not receive the same service 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 they are uploading corrupted content. As shown in [16], KaZaA is not able to detect malicious peers. In our scheme, malicious peers will be detected, and punished by receiving less service.

BitTorrent, a widely used and scalable second generation P2P protocol adopts the *tit-for-tat* strategy. Using this strategy, peers are able to optimize their download and upload rates. Typically, peers upload to the k peers that recently provided them with the best downloading rate. Recent studies [1, 11, 23] have shown that the tit-for-tat strategy does not effectively reward good peers and punish rogue peers. In addition, this incentive mechanism is rate-based and selfish peers can get more bandwidth while honest peers can unfairly receive low download rates [18].

In [24], the authors propose a reputation scheme that combines trust and incentive mechanisms. The proposed scheme uses explicit and implicit evaluations such as files’ vote and retention time, download volume and users’rank to construct direct trust relationships.

Based on the reputations, service differentiation is used to motivate users to share, vote on files, rank users and remove fake files. However, performance evaluation is needed to assess the performance of the proposed scheme.

In [13], the authors propose a service differentiation based on the amounts of services each node has provided to a P2P community. A resource distribution mechanism is proposed to increase the utility of the whole network and provides incentive for nodes to share information. A generalized mechanism that provides incentives for nodes having heterogenous utility functions is also described. However, all the proposed incentives mechanisms have been focusing on completely decentralized systems and almost no attention was given to partially decentralized systems. The proposed schemes for completely decentralized systems are poorly suited for partially decentralized systems.

9 Conclusion

In this paper, we introduced a novel scheme to assess the contribution behavior of peers and use it for service differentiation in partially decentralized peer-to-peer systems. The peer's Contribution Behavior is computed based on both its Availability and Involvement. The Contribution Behavior is used as a guideline for service differentiation rather than the peer's reputation. While, the peer's reputation reflects the authenticity of the files this peer is uploading, the peer's contribution reflects its availability to sharing files taking into account its uploads compared to its downloads. The proposed scheme provides the right incentives for free riders to share files. Performance evaluations 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. On the other hand, good peers receive better service. Assuming a rational behavior, free riders tend to increase their contribution to get better service and indirectly reducing the load supported by good contributor peers. Moreover, the proposed scheme generates a competitive environment where peers are forced to continuously participate to benefit from the system, this way, reducing significantly the milking phenomenon.

References

1. Bharambe A, Herley C, Padmanabhan V (2006) Analyzing and improving a BitTorrent network's performance mechanisms. In: IEEE 25th INFOCOM, pp 1–12
2. Aberer K, Despotovic Z (2001) Managing trust in a peer-2-peer information system. In: The 9th international conference on information and knowledge management, pp 310–317
3. Adar E, Huberman BA (2000) Free riding on Gnutella. Tech. rep., HP
4. Axelrod R (1984) The evolution of cooperation. Basic Books, New York
5. Chang E, Dillon T, Hussain FK (2006) Trust and reputation for service-oriented environments. Wiley, New York (2006)
6. Cornelli F, Damiani E , di Vimercati SDC, Paraboschi S, Samarati P (2002) Choosing reputable servents in a P2P network. In: The 11th international world wide web conference, pp 376–386
7. Gambetta D (2000) Can we trust trust? In: Trust: making and breaking cooperative relations, chap. 13, pp 213–237. Published Online
8. Gummadi K, Dunn RJ, Saroiu S, Gribble SD, Levy HM, Zahorjan J (2003) Measurement, modeling, and analysis of a peer-to-peer file sharing workload. In: 19th ACM symposium on operating systems principles, pp 314–329
9. Gupta M, Ammar M (2003) Service differentiation in peer-to-peer networks utilizing reputations. In: ACM Fifth international workshop on networked group communications
10. Gupta M, Judge P, Ammar M (2003) A reputation system for peer-to-peer networks. In: ACM 13th international workshop on network and operating systems support for digital audio and video, pp 144–152
11. Jun S, Ahamad M (2005) Incentives in BitTorrent induce free riding. In: Workshop on economics of peer-to-peer systems, pp 116–121
12. Kamvar SD, Schlosser MT, Garcia-Molina H (2003) The EigenTrust algorithm for reputation management in P2P networks. In: The 12th international world wide web conference, pp 640–651
13. Ma RTB, Lee SCM, Lui JCS, Yau DKY ((2006)) Incentive and service differentiation in P2P networks: a game theoretic approach. IEEE/ACM Trans Netw 14(5):978–991
14. Marsh S (1994) Formalising trust as a computational concept. Ph.D. thesis, University of Stirling
15. Marti S, Garcia-Molina H (2004) Limited reputation sharing in P2P systems. In: ACM conference on electronic commerce, pp 91–101
16. Mekouar L, Iraqi, Y, Boutaba R (2005) Detecting malicious peers in a reputation-based peer-to-peer system. In: The IEEE consumer communications and networking conference (CCNC), pp 37–42
17. Mekouar L, Iraqi Y, Boutaba R (2006) Peer-to-peer most wanted: malicious peers. In: International computer networks journal. Special issue on management in peer-to-peer systems: trust, reputation and security, vol 50(4). pp 545–562
18. Shah P, Pâris JF (2007) Incorporating Trust in the BitTorrent Protocol. In: International symposium on performance evaluation of computer and telecommunication systems
19. Papaioannou TG, Stamoulis GD (2004) Effective use of reputation in peer-to-peer environments. In: The proceedings of IEEE/ACM CCGrid: international symposium on cluster computing and the grid, pp 259–268
20. Papaioannou TG, Stamoulis GD (2006) Reputation-based policies that provide the right incentives in peer-to-peer environments. In: The computer networks journal: special issue on management in peer-to-peer systems: trust, reputation and security
21. Rafaeli S, Hutchison D (2003) A survey of key management for secure group communication. ACM Comput Surv 35(3):309–329

22. Ranganathan K, Ripeanu M, Sarin A, Foster I (2004) Incentive mechanisms for large collaborative resource sharing. In: The proceedings of IEEE/ACM CCGrid: International symposium on cluster computing and the grid, pp 1–8
23. Thommes R, Coates MJ (2005) BitTorrent faireness: analysis and improvements. In: Workshop of the internet telecommunications and signal processing
24. Yang M, Feng Q, Dai Y, Zhang Z (2007) A multi-dimensional reputation system combined with trust and incentive mechanisms in P2P file sharing systems. In: Proceedings of the 27th international conference on distributed computing systems workshops
25. Zhang Y, Fang Y (2007) A fine-grained reputation system for reliable service selection in peer-to-peer networks. Trans. IEEE 18(8):1134–1145

Loubna Mekouar received her M.Sc. degree in Computer Science from the University of Montreal in 1999. She is currently a Ph.D. student at the School of Computer Science at the University of Waterloo. Her research interests include trust and reputation in peer-to-peer systems, Quality of Service in multimedia applications, and network and distributed systems management.

Youssef Iraqi received his B.Sc. in Computer Engineering, with high honors, from Mohammed V University, Morocco, in 1995. He received his M.Sc. and Ph.D. degrees in Computer Science from the University of Montreal in 2000 and 2003 respectively. From 1996 to 1998, he was a research assistant at the Computer Science Research Institute of Montreal, Canada. From 2003 to 2005, he was a research assistant professor at the David R. Cheriton School of Computer Science at the University of Waterloo. He is currently an assistant professor at Dhofar University, Salalah, Oman. His research interests include network and distributed systems management, resource management in multimedia wired and wireless networks, and peer-to-peer networking.

Raouf Boutaba received the M.Sc. and Ph.D. Degrees in Computer Science from the University Pierre & Marie Curie, Paris, in 1990 and 1994 respectively. He is currently a Professor of Computer Science at the University of Waterloo. His research interests include network, resource and service management in wired and wireless networks. Dr. Boutaba is the founder and Editor-in-Chief of the IEEE Transactions on Network and Service Management and on the editorial boards of several other journals. He is currently a distinguished lecturer of the IEEE Communications Society, the chairman of the IEEE Technical Committee on Information Infrastructure. He has received several best paper awards and other recognitions such as the premier's research excellence award.