Peer-to-Peer Netw Appl
DOI 10.1007/s12083-010-0069-z

Trust-based exchange of services to motivate cooperation in P2P networks

Anna Satsiou · Leandros Tassiulas

Received: 17 April 2009 / Accepted: 18 March 2010

Abstract In this paper we propose a trust-based exchange framework to motivate cooperation among peers of different consumption, contribution and service evaluation profiles. Our framework consists of distributed resource allocation and server selection policies based on local reputation vectors. We present how proposed policies outperform previous work and lead to the autonomic formation of coalitions between peers who mutually profit by exchanging their services. In this way the utilities of all peers progressively improve without pre-existing knowledge of one another's service evaluation and capability profiles. Peers' coalitions are dynamically reformatted, adapting to network changes, e.g., when new peers enter the system or peers vary their profiles. Only misbehaving (non contributive) peers cannot benefit by our framework, which efficiently blocks misbehavior.

Keywords Cooperation enforcement · Exchange of multiple services · P2P networks · Reputation · Trust-based resource allocation policies

A preliminary version of this paper appeared in IEEE P2P Computing 2007 [1].

This work was supported through the 03ED918 research project within the framework of PENED, co-financed by E.U.-European Social Fund (75%) and the Greek Ministry-GSRT (25%), as well as the EU IST FP7 STREP NANODATACENTERS 223850.

A. Satsiou (✉) · L. Tassiulas
Department of Computer Engineering
and Telecommunications, University of Thessaly,
38221 Volos, Greece
e-mail: satsioua@uth.gr

L. Tassiulas
e-mail: leandros@uth.gr

A. Satsiou
Center for Research and Technology Hellas (CERTH),
Informatics and Telematics Institute,
6th Km Charilaou-Thermi, Thessaloniki, Greece
e-mail: satsiou@iti.gr

1 Introduction

Peer-to-peer (p2p) overlay systems' rapid and promising evolution is based on their numerous attractive features, providing the efficient means for resource exchange in the Internet. Peers can simultaneously act as clients and servers taking part in a collaborative work process. However, although cooperation is of utmost importance to p2p systems, peers seem to be naturally incited only to consume but not contribute to a community. This is a social phenomenon reported as "tragedy of the commons" [2] or "free riding" that most of the users are reluctant to cooperate, and only a small number of them are willing to share their resources. In Gnutella, for example, a report [3] in 2005 indicated that 85% of Gnutella users are free riders. Free riding leads to the degradation of the system performance, as not all peers' needs can be satisfied. The few peers that volunteer to provide their services are not necessarily those who have the desirable ones, and eventually act like centralized servers. Thus, peers become susceptible to denial of service attacks.

In order to alleviate free riding, many methods were proposed to enforce cooperation among users, like pricing-based schemes [4], game theoretical methods [5–7], and trust and reputation based methods [8–10, 28]. On one hand, the need of trusted virtual banks to

Published online: 21 April 2010

regulate the transactions in credit-based schemes and on the other hand, the fact that game theoretical methods rely on unrealistic assumptions that all peers have the exact information of the entire game, necessitated the use of simpler and more practical methods to foster users to cooperate.

Trust and Reputation systems have been extensively investigated in p2p systems to distinguish and avoid malicious peers by applying suitable provider selection mechanisms [11–13]. These mechanisms, though, do not provide incentives for cooperation, as malicious peers have no repercussions and cooperative peers are not motivated to contribute more resources than they currently do. Acknowledging such concerns, recent work in reputation systems [8–10, 28] provides simple incentive mechanisms to enforce collaboration between peers by controlling not only the provider selection policy but also the client selection policy. Trusted, thus contributive, peers are rewarded by receiving preferential treatment, while misbehaving peers are punished by not being served.

Aforementioned work is focused in the case that one single service is exchanged in the network (processing power in [10], bandwidth in [7], etc). However, many peers may be weak in providing a certain service and although they are cooperative enough they cannot increase their reputation, and thus their revenue, because of their low capability in providing this particular service. In a multiple services network, users can benefit by exchanging different kind of services according to their needs and capabilities.

The necessity for a common exchange framework for multiple services like memory, CPU, storage capacity, bandwidth, etc, arises from the fact that peers' capabilities and needs for various services are different. Some peers may have storage capabilities and CPU needs and some other peers may be processing powerful and have storage needs; these groups of peers could cooperate to improve their performance. However, this common framework, under which different services are exchanged, introduces a complex economy, as peers may value services differently, according to their needs and subjective criteria. It has been proposed [14] that one single unit could be used for all computing resources, the so called "computon" the price of which would follow supply and demand. However, till now no successful effort has been made to define "computon". On the other hand, several market-based resource allocation systems [15, 16] have been proposed for distributed computing infrastructures, like Planetlab (http://www.planet-lab.org/) or computational grids [17] to regulate the transactions. Market models, though, require a trusted authentication infrastructure to authenticate bids, report resource capabilities and preferences and verify account balances. These necessities complicate the system and make it impractical.

In this paper we propose a simple unified fully-distributed trust-based exchange framework for peers with different capability, consumption and service evaluation profiles. The goal is to foster cooperation and help peers recognize and transact with other peers with whom they can mutually benefit. We use a common system design that can be applied to both single service (e.g., file sharing systems) and multiple services' systems (e.g., p2p grids). Proposed framework consists of an autonomous reputation scheme, trust-based resource allocation and server selection policies which are robust under the presence of misbehavior, strategic peers, greedy peers and newcomers. We quantify a peer's capability in providing a good quality service by a novel reputation metric and an update mechanism is proposed to dynamically track the contribution level of the peer as the system evolves. Our trust-based allocation policies determine the quality of service given to all competing for resources peers at a given time according to their trust/reputation, demands, and profiles. On the other hand, our trust-based server selection policy is used to choose suitable providers of the requested service and avoid misbehaving (non contributive) ones. We note that the terms trust, reputation and contribution level are used interchangeably in this paper. Proposed policies have low complexity and the only information that is needed to pass from one peer to another is the requested amount of service. As far as we know, no previous work has attempted to encompass all aforementioned important aspects of services' exchange in p2p systems under a common design framework.

The remainder of this paper is organized as follows. Section 2 describes some related work for both the single and multiple services cases. Section 3 presents the proposed trust-based exchange framework for one and multiple services, its robustness over various attacks and potential application scenarios. In Section 4 we discuss the performance evaluation details and in Sections 5 and 6 we exhibit the results for the single and the multiple services case respectively, providing comparison results with related work. Finally, Section 7 concludes the paper.

2 Related work

For the single service case many trust and reputation based policies have been proposed. However, most of them [8–10, 24, 28] do not consider quality of service issues. Nodes are simply distinguished to altruistic (or cooperative) and egotistic (or misbehaving/malicious), according to whether they provide services or not. The different quality of service offered and received or the different capabilities and needs of the peers are not considered in the allocation decisions.

From our knowledge, there is no previous work in determining the quality of service offered to competing peers based on a reputation metric. There are several papers, though, that propose bandwidth allocation policies in p2p systems based on an implicit contribution level of the users. In [18] a token stealing algorithm based on a token-bucket model is proposed for p2p streaming systems, according to which each peer reserves a portion of his capacity to repay his neighbors for the packets they have uploaded; however it does not apply for the individual needs of the competing peers.

In [5] and [7] a peer's tendency to contribute is represented by the "contribution value", while in [19] "ranking" is a metric of the behavior of a peer. Allocation decisions are based on those; however these are used as abstract ideas and it is not explained how they are measured and kept, rather they are given random values so that proposed policies can be evaluated. In our approach and evaluation, on the contrary, reputation is dynamically adapted and affected by the allocation decisions taken by the peers as the system evolves, corresponding to a more realistic system. What's more, aforementioned policies do not account for peers' request generation profiles.

Finally, there is a lot of work that has studied BitTorrent-like networks. In order to confront free-riding effect, BitTorrent [20] uses a tit-for-tat (TFT) strategy to determine which neighbors to upload data to. However, the unfairness issues of BitTorrent reported in [21, 22] leaded to proposed improvements [22] over it and new reciprocation-based mechanisms. According to these mechanisms, the offering bandwidth rate to a given peer at a given time is controlled to be equal to the receiving bandwidth at that time by the same peer [21]. One drawback of these schemes is that their memoryless feature does not motivate peers who already downloaded their files (seeds) to stay in the system and keep on contributing resources to others. In Section 5.6 we show how our policies can prompt such peers to remain and contribute in the system with the profit of improving their performance in their future transactions.

All aforementioned work does not explore the possible autonomic formation of coalitions among peers of similar capabilities and needs. The autonomic creation of such coalitions could be of particular importance to self-organized network communities who seek to reach certain common goals. In [22] it is reported that the clustering of homogeneous (similar capacity) peers in a BitTorrent-like system ensures fairness in the system. The clustering in this work, though, presupposes that a tracker is aware of all peers' capabilities and clusters them accordingly, while in our work this kind of clustering (coalition formation) is the result of the implementation of the proposed algorithms and peers are unaware of one another's capabilities.

For the multiple services case, apart from the market-based models that induce several complexities, the only studies, as far as we know, that deal with the exchange of multiple services are [23] and [24]. In [23] a reciprocation-based economy is studied for multiple services in p2p grids. We describe this scheme in detail in Section 4.2.2, as we perform comparison results with it. Work in [24] examines a reciprocity-based mechanism to promote cooperation between peers who have different expertise in providing several types of resources. A reciprocative peer is going to serve a requester if his expected utility from the interaction with him is positive. The expected utility is judged by the previous interactions with the same peer. Both [23] and [24] face the same limitations, i.e., the individual peers' demands and capabilities are not considered in the allocation decisions, no server selection policy is used and the work does not investigate the possible formation of peers' coalitions.

3 Trust-based policies

3.1 Model description

We consider a p2p network of N peers who provide and consume a single or multiple services. Services being exchanged could be processing power, storage, bandwidth, etc. Each peer in the system has different capabilities in providing each service of the system, expressed by his capability profile. For M services in the system, the capability profile of a peer i is $\mathbf{C}_i = \{C_{i,1}, ..., C_{i,M}\}$, where $C_{i,m}$ represents the capability of peer i in providing service m. For example, in case of bandwidth, this capability is represented by the total upload capacity of the peer. In case of only one service being exchanged in the system the capability profile of a peer i is reduced to the value C_i indicating his capability in providing the particular service in the network.

Each peer is further characterized by his service value profile $\mathbf{V}_i = \{V_{i,1}, ..., V_{i,M}\}$, where $V_{i,m}$ represents the value put from peer i on a unit of service m, according to peer i's particular needs or other subjective criteria, like the importance of one service compared to the importance of another one. Hence, we consider that each time a peer i donates a unit of service m, he incurs a cost of $V_{i,m}$, whilst each time he receives a unit of resource m, he has a profit of $V_{i,m}$. Several possibilities are considered for the distribution of the values of peers for the various services; they are either (a) all equal to 1, or (b) randomly chosen in [0,1], or (c) correlated with the capabilities of peers in providing the corresponding services. Last case is considered under the concept that if a peer is weak in providing a certain service, he incurs a high cost in donating it and a high profit in receiving it. It is like peers value the various services of the system based on their own supply and demand. This is reasonable, given that peers mostly need services that they cannot produce by themselves, as in p2p grids. However, there may be systems where peers are weak in providing services which they do not need; thus these services' values are low to the corresponding peers. Hence, the most general case is (b), where each peer is requested to value the services he provides and requests with a value between 0 and 1, according to his own criteria. Service value and capability profiles do not need to be global information, but are somehow revealed by the reputation and allocation mechanisms, as we will see in the sequel.

Finally, each peer has his own consumption profile $\mathbf{S}_i = \{\{D_{i,1}, G_{i,1}\}, \ldots, \{D_{i,M}, G_{i,M}\}\}$, where $D_{i,m}$ represents the demands of peer i for service m and $G_{i,m}$ represents the request generation rate of peer i for service m. $D_{i,m}$ is Zipf distributed between [min $D_{i,m}$, max $D_{i,m}$] with skew $sk_{i,m}$ according to the particular needs of each peer i for service m.

We consider that our system progresses in periods. In each period, each peer i generates Q_i requests. Each one of the Q_i requests is for a service m with probability $p_{s_{i,m}}$, such that $Q_i \times p_{s_{i,m}} = G_{i,m}$. In each one of his requests, peer i reports his demands $D_{i,m}$ for the requested service m. $D_{i,m}$ can be certain bandwidth, storage, CPU cycles, etc, for Bag-Of-Tasks [30] or other applications. Peers act both as servers and clients simultaneously. A peer as a client selects the server to which he will direct his request either randomly or based on some attributes that indicate the quality of service he may expect from the requested service. A peer as a server collects at every period the requests that have been directed to him and allocates his available resources according to one of the proposed allocation policies. The service is not granted for more than a period, in order to give the chance for other peers to access it. Every new period, peers redirect their requests to the same or possibly other servers aiming at improving their received quality of service. In case of long-running applications peers should engage in continuous periods of resource competition.

3.2 Reputation

In this section we describe the reputation system that we propose. Each peer i keeps a reputation vector consisting of the local reputations of all other peers for all the services provided in the system, i.e., $\mathbf{R}_i^p = \{R_{i,z,m}^p, \forall z, m\}$, where $R_{i,z,m}^p$ represents the local reputation of peer z to peer i for service m at a given period p, and is calculated as follows: If in a given transaction t, a peer i demands from peer z a certain amount $D(t)$ of service m, and finally receives $x(t) \leq D(t)$, then the ratio $x(t)/D(t)$ represents how well server z satisfied the needs of peer i for service m for the given transaction. Peer i keeps a log with the satisfaction ratios $x(t)/D(t)$ from all transactions with peer z for service m. Then, he can calculate the local reputation of peer z in providing service m at a given period p following the formula:

$$R_{i,z,m}^p = \frac{1}{\mathrm{Req}_{i,z,m}^p} \sum_t \frac{x(t)}{D(t)}, \tag{1}$$

where $Req_{i,z,m}^p$ is the total number of i's past requests from peer z for service m till period p and t refers to the past transactions of peer i with peer z for service m till period p. In the same way, the reputation of any peer is calculated. It is obvious that reputations can take any value between 0 and 1. Please note that the reputation of a given peer for a given service reflects the average satisfaction, in terms of quality of service, he offers to his requesters for the corresponding service, as opposed to previous work [8–10, 28] where the reputation is calculated as a function of binary ratings (success or failure of a service request). This is an important difference, since a reputed user in our scheme is one that not only provides his help (e.g., upload a file) but also provides a high QoS in order to satisfy his requesters' demands (e.g., upload a file with the requested speed). In this way we can distinguish peers of different capabilities in satisfying the various service requests.

Each peer i further keeps a vector with the overall local reputations of all other peers $\widehat{\mathbf{R}}_i^p = \{\widehat{R}_{i,z}^p, \forall z\}$. The overall reputation of peer z to peer i at a given period p, $\widehat{R}_{i,z}^p$, is given by averaging the satisfaction ratios of peer i from all his service requests from server z till period p. Overall reputation reflects the general tendency of a peer to satisfy his requesters, irrespective of the requested services; whenever it falls down a threshold, called misbehavior threshold, peer is considered to misbehave.

New peers in the system are awarded an initial small reputation for each service, higher than the misbehavior threshold, in order to have the chance to receive some resources. If they prove to be collaborative enough, their reputation will increase, permitting them to improve their performance. We have seen from our simulations that a history of 10 past and most recent transactions for a given service is enough to evaluate one's local reputation for the particular service. More specifically, we performed simulation experiments with different peers' populations (10^2–10^6 peers) and profiles and saw that a 10-log most recent history was the smallest log achieving the same performance with the whole history log, irrespective of the system parameters used, as soon as peers don't vary their behavior.

Therefore, each time a peer i completes 10 transactions with peer z for service m, he starts replacing the oldest rating transactions/satisfaction ratios with newer ones, always keeping a history/log of, at the most, 10 most recent transactions with peer z and service m, keeping the overhead low. If the log has fewer inputs (e.g., due to limited interactions with peer z for service m), reputations are calculated based only on those. In order to clarify these issues, we describe the following example. Suppose that there are two services provided in the system, $s1$ and $s2$. If peer i has the following recent log for peer z in providing service $s1$, (0.1,0.3,0,0.6,0.4,0,0,1,0,0), and the following log for service $s2$ (0,0.1,0.2,0.1), he calculates the local reputation of peer z for $s1$ as $(0.1 + 0.3 + 0 + 0.6 + 0.4 + 0 + 0 + 1 + 0 + 0)/10 = 2.4/10 = 0.24$ and the local reputation of peer z for $s2$ as $(0 + 0.1 + 0.2 + 0.1)/4 = 0.4/4 = 0.1$. Peer i can also calculate the local overall reputation of peer z as $(2.4 + 0.4)/14 = 0.2$. We note that by using only most recent transactions to extract reputations, the system quickly adapts to network changes, e.g., when peers vary their contributions through time, as we will see in the evaluation section.

In our work, we use an autonomous reputation scheme, as in [10, 23], where peers use local reputations. A local reputation scheme is simple to implement and there is no need for a cryptographic infrastructure or a centralized server to keep the global reputations of the peers. Moreover, the load from communication exchanges due to recommendations and other possible problems like misreporting, reputation pollution and collusion, appearing in reputation rating and voting systems like the ones used in [8, 9, 11, 12] are avoided. When using local reputations, a peer gives preferential treatment to the ones that have been generous and helpful to him in the past and not to all the generally cooperative peers. In certain systems this could help him to recognize and favor the appropriate traders for him. In a file sharing system, for example, high capacity peers are better off by trading with other high capacity peers. Low capacity peers can be equally satisfied when trading either with other low capacity or high capacity peers. Their small capacity may not be enough to sustain the upload rate of a higher capacity peer, constraining them to download content in a lower rate [22]. Therefore, a low capacity peer who is considered as "low reputed" by high capacity peers, may be a good trader for some other low capacity peers. Similarly, in a multiple services system peers can recognize and prioritize their partners, based on their own evaluation and need of the various provided services. Such a reputation scheme along with our allocation and server selection policies lead to the dynamic formation of coalitions among peers who have similar capabilities or interests and needs.

On the other hand, when peers use recommendations/votes to determine one's reputation, they can speed up their perception of the p2p system. However, there should be mechanisms to guarantee the trustworthiness/sincerity of the peers who vote [25], which usually induce complexity to the system.

3.3 Trust-based allocation policy (TAP)

In this section we consider that a single service is being exchanged in the network (e.g., bandwidth) and we investigate the properties of our proposed allocation policy for the single service case. Suppose that a peer z receives at a given period p, n requests for bandwidth and that his upload capacity is C_z. We use a vector $\mathbf{L} = \{L_1, ... L_n\}$, each element of which represents a specific requester of peer z with demands expressed by $D_{L_l}, l \in \{1, .., n\}$. Then, we consider that peer z will allocate C_z in order to solve the maximization problem:

$$max \sum_{l=1}^{n} \frac{R^p_{z,L_l}}{D_{L_l}} x_{L_l} \text{ s.t. } \sum_{l=1}^{n} x_{L_l} \leq C_z \text{ and } x_{L_l} \leq D_{L_l} \forall l, \quad (2)$$

where R^p_{z,L_l} represents the local reputation of peer L_l to peer z at period p, and x_{L_l} represents the allocated resource of peer z to peer L_l. We will refer to this policy as TAP (Trust-based Allocation Policy).

In case that only one peer is competing for the available resources of a server, the server will completely satisfy the peer's demands, unless peer's reputation is below a certain threshold, set by the system to mark misbehavior. In this case, competing peer does not receive any resource, as a punishment for his zero contributions.

We thought of this allocation policy because its benefits are twofold. Firstly, it shares the available resources in a way to give more resources to more reputed peers, and secondly, it tries to maximize the satisfactions of the competing peers. The notion of satisfaction is very important, since the same amount of resources may be of different importance to different peers and has different significance in different times of the day, e.g., during congested periods a medium QoS can be considered more satisfying than in less congested ones. Peers have different criteria in judging a service; however, by using their reported demands, we can account for their different degrees of satisfaction.

Both the objective function and the constraints in the maximization problem (2) are linear, so the solution in a period p can be easily found by sorting peers in decreasing order according to their ratio $R^p_{z,L_l}/D_{L_l}$. Then, the server will satisfy the needs of the competing peers, starting from the first one in the order, satisfying all his demands, as soon as they do not exceed server's maximum upload capacity, and continuing with the rest of the peers, till all resources are exhausted or all peers are completely satisfied.

Applying TAP, peers with the highest ratio $R^p_{z,L_l}/D_{L_l}$ will gain the largest feasible bandwidth. So, it becomes obvious that peers do not have any incentive to request more than their real needs, aiming at receiving better service. On the other hand, peers do not have an incentive to ask for less than they need, because they will not be able to get more than what they asked for. Nevertheless, if they are reputed enough, even their higher demands will be satisfied, depending on the server's capabilities and availability (e.g., if no other equal or more reputed peers request for service at that moment).

Finally, it is evident from the solution of Eq. 2 that for any two competing peers i and j with $\left(R^p_{z,i}/D_i\right) < \left(R^p_{z,j}/D_j\right)$, their satisfactions for the given request will satisfy $\left(x_i/D_i\right) \leq \left(x_j/D_j\right)$. This means that peers with the highest contribution level per unit resource request will receive the highest satisfaction. This final point clearly shows that the proposed allocation policy provides incentives for cooperation and contributions. Actually, the higher demands a peer has, the more contributive/reputed he has to be, in order to satisfy them. As an example, consider three peers A, B and C who request service from peer D with $C_D = 6Mb/s$. Their demands are $D_A = 3Mb/s$, $D_B = 5Mb/S$, $D_C = 5Mb/s$ and their local reputations are $R_{DA} = 0.3$, $R_{DB} = 0.5$, $R_{DC} = 1$. Most reputed peer C will be fully satisfied, receiving 5Mb/s, while peers A and B will each receive 0.5Mb/s.

The benefits of the TAP policy are also exhibited through the simulation results of Section 5. The main results exhibit the correlation between the contribution levels, reputations and satisfactions of the peers. Peers can only receive resources in proportion to their contributions; thus misbehaving (non contributive) peers cannot receive any resources (Section 5.1) and strategic peers fail to maximize their obtained resources when they do not offer a proportional amount of resources to the network (Section 5.4). Our simulation results further exhibit the performance stability of the system in the presence of newly arriving peers (Section 5.2) and the formation of coalitions between peers of similar capabilities and needs (Section 5.3).

3.4 Trust-based allocation policy for multiple services (TAP-MS)

In this section we consider that peers may exchange more than one services. We extend the TAP policy for multiple services p2p systems, regarding the peers' service evaluation profiles. Suppose that a peer z receives in a given period p, n requests for a given service m and his capacity for service m is $C_{z,m}$. As in previous section, we use the vector $\mathbf{L} = \{L_1, \ldots L_n\}$ for representing the different requesters of peer z with demands expressed by $D_{L_l,m}$, $l \in \{1, .., n\}$. Then, we propose that peer z will allocate $C_{z,m}$ in order to solve the maximization problem:

$$max \sum_{l=1}^{n} \sum_{s=1}^{M} \left(V_{z,s} \times R^p_{z,L_l,s}\right) \frac{x_{L_l,m}}{D_{L_l,m}}$$

$$\text{s.t.} \sum_{l=1}^{n} x_{L_l,m} \leq C_{z,m} \text{ and } x_{L_l,m} \leq D_{L_l,m} \forall l, \tag{3}$$

where $x_{L_l,m}$ represents the allocated resource of peer z to peer L_l for service m.

In practice, each peer values a service with different criteria. According to this service valuation each peer should be able to recognize and favor those who better provide him with the most valuable services. Towards this goal, under TAP-MS each server multiplies his requesters' reputations for a given service with the value of the service, as estimated by himself. The allocated resources to each competing peer depend on the weighted sum of the peer's reputations in providing each service of the system, where the weights are the corresponding values of the services, as estimated by the server. By weighting reputations in the allocation decision, the server z provides more resources to those peers that have offered him more satisfaction (captured by their reputations) in the services he values more, in order to repay them and foster their bond.

Indeed, the simulation results of Section 6 show that the utilities of peers progressively improve throughout their lifetime in the system, indicating the formation of bonds (coalitions) between peers who mutually benefit by their transactions, when (a) all peers of the system have homogeneous service value profiles (Section 6.1), (b) peers have random service value profiles (Section 6.2), (c) their service value profiles are correlated with their capability profiles (Section 6.3) and finally (d) their service value profiles change through time (Section 6.5).

As in TAP, in case that only one peer is competing for the available resources of a server for a given service, the server will completely satisfy his demands, unless peer's overall reputation is below the misbehavior threshold. The solution of Eq. 3 can be found following the same procedure as in TAP. Similarly, it can be proved that peers with the highest weighted reputation per unit resource request will receive the highest satisfaction.

3.5 Enhanced trust-based allocation policy (ETAP)

The proposed allocation policy for the single service case, described in Section 3.3, is fairly suitable when all peers generate requests with the same average rate. However, when some peers produce many more requests than others, they should be properly handled, in order to not absorb most of the resources in the system in cost of other collaborative peers. The following enhanced allocation policy (ETAP) is proposed to cope with this case. Peer z will share his available resources to n competing peers by solving:

$$max \sum_{l=1}^{n} \left(\frac{R^{p}_{z,L_l}}{D_{L_l}} \times \left(\frac{Req^{p}_{z,L_l}}{Req^{p}_{L_l,z}} \right)^{a} \right) x_{L_l},$$
$$\text{s.t} \sum_{l=1}^{n} x_{L_l} \leq C_z \text{ and } x_{L_l} \leq D_{L_l} \forall l, \quad (4)$$

where Req^{p}_{z,L_l} is the total number of requests that peer z has sent to each peer L_l till period p. For any new peer, these variables are set to 1 and change through time. Variable a is a constant of the system and can be adjusted by the software provider according to his perspectives. The bigger a is, the more restrictive the policy is for the high rate peers.

By using ETAP, we favor peers to whom we have sent more requests compared to what they have sent to us, in order to balance this difference. Note that this policy is of the interest of both high rate and low rate peers. On one hand, high rate peers will have the chance to compensate peers who have satisfied a great amount of their requests, and thus hope for more future collaboration with these peers. On the other hand, low rate peers will be able to restrict the excess requests of high rate peers, satisfying even other collaborative ones whom they need.

The solution of Eq. 4 can be found with a similar procedure as in previously described policies. We can see that high rate peers need to be more contributive than low rate ones, in order to satisfy all of their requests. This is also exhibited through the simulation results of Section 5.5.

3.6 Enhanced trust-based allocation policy for multiple services (ETAP-MS)

ETAP policy can also be extended for the multiple services case by applying Eq. 5. Peer z will share his available resources to L_l competing peers for service m by solving:

$$max \sum_{l=1}^{n} \sum_{s=1}^{M} \left[\left(\frac{Req^{p}_{z,L_l,s}}{Req^{p}_{L_l,z,s}} \right)^{a} \times V_{z,s} \times R^{p}_{z,L_l,s} \right] \frac{x_{L_l,m}}{D_{L_l,m}},$$
$$\text{s.t} \sum_{l=1}^{n} x_{L_l,m} \leq C_{z,m} \text{ and } x_{L_l,m} \leq D_{L_l,m} \forall l, \quad (5)$$

where $Req^{p}_{z,L_l,s}$ is the total number of requests that peer z has sent to each peer L_l till period p for service s.

Compared to TAP-MS, under ETAP-MS each requester's reputation is further weighted by the ratio of the server's load (number of requests) served by the requester over the requester's load served by the server. The simulation results of Section 6.7 demonstrate the benefits of ETAP-MS compared to TAP-MS in systems of peers with heterogeneous request generation profiles.

3.7 Trust-based server selection policy

Along with the proposed allocation policies, a proper trust-based server selection policy is needed in order to block misbehavior and help peers direct their requests to those that can best provide the requested service. When a peer enters the system for the first time, he does not have information about the behaviors and the capabilities of the other peers. However, we assume that he is aware of his potential servers through an underlying service discovery mechanism. In our study, we consider that all peers are potential servers for each other; we will elaborate more on this later. So, for a short duration of time (acquaintance duration) a newcomer will direct his requests to all potential servers with equal probability (random server selection policy), till he obtains a global view of the system. Similarly, preexisted peers in the system will use the random selection policy among their top reputed and new peers during time duration equal to the acquaintance duration in order to test the behavior of the newcomers. After this time duration, the probability with which a peer i directs his request to a peer z for a service m at a given period p is directly proportional to peer z's local reputation to i, for service m, as $p^p_{i,z,m} = R^p_{i,z,m} \big/ \sum_{j \in J^p_i} R^p_{i,j,m}$, where J^p_i is the set of all peer i's transacting peers till period p.

We would like to clarify that peers treat anyone with whom they did not have previous transactions as a newcomer. Therefore, peers get to know newcomers, as soon as the latter sends requests to them. At that time, newcomers are considered as possible servers for future peers' requests. Service discovery issues are out of the scope of this paper but we rather consider that an underlying resource discovery mechanism, like NodeWiz [29] would be responsible to inform newcomers about their potential servers. A NodeWiz network consists of peers responsible to store adverts from service providers and answer to client queries. What service requesters and providers need to know is solely the address of one peer of the NodeWiz network, to which they will submit queries and advertise their own services. A query answer will contain a subset of those peers who match the request specification. Service requesters can then use our policies to recognize among suggested servers, the appropriate partners with whom they can improve their utility from the system, according to their capability, consumption and service evaluation profiles. The only information that they need to obtain from the system is the IP addresses of the peers who claim to provide the requested services, neither their reputations nor their profiles, which may be falsely advertised in the service discovery mechanism.

3.8 Robustness of proposed scheme over various attacks

Misbehaving (non-cooperative) peers Misbehaving peers are peers who do not contribute any resources to the system, seeking, however, to consume as much resources as possible. Our scheme incorporates the appropriate mechanisms to disclosure such peers from the system. As stated in Section 3.2 all newcomers are given an initial small reputation, InitRep, in order to bootstrap the system and let them prove their cooperation. However, if such peers are misbehaving and do not contribute any resources, their initial reputation will quickly decrease, as can be seen from the reputation update mechanism presented in the same section. Actually, after a single transaction of a peer X with a misbehaving one, the latter's local reputation to X will be half his initial one (i.e., (InitRep+0)/2) and will even more decrease at each new transaction (InitRep+0+0)/3, etc. Therefore, the chances of misbehaving peers with such low reputations to receive service compared to contributive or even newly arriving peers are small, as shown in Section 3.3. Moreover, according to our policies, as soon as peers' overall reputation falls below the misbehavior threshold, which is set smaller than InitRep, they will be denied service, even when they are the sole competitors for it. On the other hand, according to our trust-based server selection scheme, peers direct their requests to the most reputed ones; thus, after the acquaintance duration period, misbehaving peers will no longer be selected as servers (or selected with a minor probability). In this way, misbehaving peers are blocked both from the allocation and server selection decisions. The efficiency of our policies over misbehaving peers can also be seen from evaluation results of Sections 5.1 and 6.6.

Whitewashing and sybil attacks Under both Sybil and whitewashing attacks, peers create many pseudonymous entities in order to exploit the network. Sybil attack is known in reputation systems where the global reputation of a peer is based on other peers' votes. By creating many pseudonymous entities, a peer can provide a large number of positive votes for himself cheating the others. In this paper, peers calculate local reputations relying only on their own experience, and thus Sybil attack cannot affect them. Under whitewashing attack, misbehaving peers switch among different identities in order to improve their performance. As explained previously, as soon as misbehaving peers are revealed, they are blocked from the system. To start over again, misbehaving peers may rejoin the system with a new identity. However, in our system the newcomers' reputation is very small to be competed with the reputation of already existing contributive peers. Thus, misbehaving peers are not able to benefit by rejoining with new identities. Newcomers really have to prove themselves as contributive peers in order to improve their initial reputation and savor the acknowledgment (contributions) of the rest of the peers. The performance of whitewashers can be seen in Section 5.2.

Strategic peers Strategic peers are peers who seek to maximize their satisfaction from the network with the least possible contributions. They are usually peers who start misbehaving as soon as they reach a high reputation in the system. Since proposed reputations are calculated based on the log of only the 10 most recent transactions with a given peer, the high established reputation of strategic peers will soon reach very small values if they stop cooperating. Actually, even one unsatisfied request from a strategic peer will affect his local reputation with weight 1/10 and significantly decrease it. If more unsatisfied transactions occur, the reputation will further decrease, reaching zero after 10 concurrent unsatisfied transactions. With our scheme in place, peers have to constantly cooperate in order to maintain high reputations. Our proposed local reputation calculation mechanism, based only on recent transactions, enable reputations to adapt very fast to possible changes in the network. The performance of various strategic peers is exhibited in Section 5.4.

3.9 Application scenarios for our scheme

Our proposed trust-based allocation scheme can be easily applied to p2p-like systems to motivate cooperation, since it does not require any centralized components, trusted third parties, communication overhead due to voting about one's reputation, e-banking systems or computational complex algorithms. It is implemented in a distributed manner at each peer independently of the others and the only information that is passed from one peer to another is the requested amount of resources. Peers are totally unaware of any system information like the capability, service evaluation or consumption profiles of all other peers, which is actually the case in real systems. The only thing that peers need to know is the IP addresses of the peers who claim to provide the requested service, which is supported by popular p2p systems. For example, Gnutella [3] uses Gnutella Web Caches among other features towards this goal, while BitTorrent uses trackers keeping track of all peers associated with a requested file. Our allocation framework could fit in file sharing systems like Gnutella and BitTorrent to control the amount of bandwidth offered to competing peers, in order to ensure cooperation among them and shield the system from free riders. In Gnutella, for example, peers could direct their file requests to potential servers following our trust based server selection policy, and would express their demands in terms of desired download bandwidth; then servers would decide the amount of bandwidth offered to them, following our trust-based allocation policies. BitTorrent defers than other file sharing systems in that peers, who download the same files at a given time, "group together" and exchange blocks of files. In Section 5.6 we explain how our scheme can be incorporated in BitTorrent, to overcome well-known deficiencies of its current reciprocation mechanism.

On the other hand, our extended framework for multiple services systems can be used in p2p grids like OurGrid (http://www.ourgrid.org/) which support Bag-of-Tasks (BoT) applications. BoT [30] are parallel applications whose tasks do not communicate among themselves during execution; thus, the failure of one task does not affect another. Our policies can be applied to control the allocation of multiple services for the execution of tasks, such as processing power, disk space and data transfers in order to improve the cooperation and performance of peers. Service discovery issues (i.e., the IP addresses of suggested servers for a given service request) in grids can be supported by NodeWiz [29]. In Section 6, we provide comparison results of our policies with a reciprocity scheme proposed for multiple services exchange in OurGrid.

Furthermore, there are several recent research and commercial efforts towards enabling peer-to-peer communities of users sharing Internet access through their wireless access points for mobile use (http://en.fon.com/) [26]. The cooperation between the existed wireless access points in a close proximity neighborhood could lead to the formation of wireless neighborhood communities [27]. These communities can provide even other kind of services to its users, like additional network capacity (e.g., for content distribution or games), the sharing of other resources such as storage (e.g., for backup services) and content (file sharing or caching). Our scheme could fit in such kind of communities to ensure that the underlying network is formed among trusted and contributive users and control the exchange of different kind of services (from the network to the application layer); however, the investigation of these issues are part of our future work.

4 Performance evaluation

4.1 Simulation models

We have developed an event-driven simulator in C++ to evaluate the performance of our policies. In the first part of our analysis, we simulate a p2p network of 100 peers who share bandwidth for file sharing and have different upload capacities and varying demands for bandwidth. Peers generate bandwidth requests according to their consumption profiles and direct them to other peers either randomly or following our proposed server selection policy. Peers, acting as servers, allocate their available upload capacity to their requesters, following our proposed allocation policies. Over this model we inquire the efficacy of our policies for the case of one single service (bandwidth) being exchanged in the system. In the second part of our analysis we examine a system with two different services being exchanged. We simulate a p2p network of 100 peers with different capability, service value and consumption profiles, as detailed described in each subsection. We have also tested our policies in much bigger networks and report our results.

In both parts of our analysis all peers start with a small initial reputation for each service of 0.07 in order to bootstrap the system. As we saw from our simulation results, the system reaches the same steady state as with any initial reputation values for each peer; however a small initial reputation value for newcomers, as 0.07, speeds up the disclosure of misbehaving peers as we will see in the following sections. Overall reputation threshold (called misbehavior threshold), below which a peer is denied service, even when he is the only one competing for resources is set to 0.01. Acquaintance duration time is set to 100 periods, while the total simulation time is 1000 periods, unless otherwise stated. The remaining simulation parameters vary and are determined in each of the following sections.

4.2 Comparison with other schemes

In order to exhibit the benefits of our proposed allocation framework for the single and multiple services case, we make comparison results with other related schemes, briefly described in the following subsections.

4.2.1 Scheme in [8]

Scheme in [8] is a representative reputation-based work that calculates reputations as a function of binary service ratings (success-failure). More specifically, each peer's reputation equals the fraction of the weighted number of his successful service provisions over the weighted total number of his service provisions. The weight of each service provision is a negative exponential function of the elapsed time. In order to map this binary description to our model we explain that if the allocated resources to a given requester are greater than zero, the service provision to this requester is considered successful (rating $= 1$), otherwise, unsuccessful (rating $= 0$). We note that particular scheme uses global reputations which are calculated based on the ratings of all peers of the p2p system, which are assumed to be truthful. The allocation decision uses the following rule: among the peers j that request the same service from a particular server z, the probability for a specific requester i to be selected equals $R_i / \sum_j R_j$, where R_i is the global reputation of peer i. This selection is translated to our model as follows; the selected requester will be offered resources equal to his expressed demands as soon as they do not exceed the server's capacity, otherwise, he will receive resources equal to the server's capacity. It is necessary to make these mappings from model in [8] to our model, because model [8] does not account for peers of different capabilities and demands. Described scheme will be denoted as BRR in the figures of this paper standing for **B**inary-**R**ating **R**eputation-based scheme. Although aforementioned scheme is a good candidate to exhibit the benefits of TAP over general BRR schemes, it is not appropriate for comparisons with TAP-MS. It totally disregards the service evaluation profiles of the peers, which is an important factor in multiple services exchange systems. In order to make comparisons with TAP-MS policy, we examine a more appropriate scheme designed for multiple-services environments, described below.

4.2.2 Scheme in [23]

The scheme in [23], similarly with TAP-MS, takes into consideration the service evaluation profiles of the peers represented by their service cost and utility functions. Two allocation policies are proposed: *PosInt* that relies on knowledge of all peer's service evaluation profiles and, thus, is not applicable in real systems, and *ExtNetFav* with which we compare our scheme, as in both of them (our scheme and *ExtNetFav*) peers are totally unaware of other peers' service evaluation profiles. In [23] instead of using reputations, they use a metric called *cost balance*. Each peer keeps a record of this number for each other peer with which it interacts, which is calculated as follows: Before two peers have ever interacted, the cost balance is set to zero. If peer A offers requested service to peer B, A decreases his cost balance with interactions with B by the cost of donating particular service (if cost balance falls below zero, it is set to zero) and B increases his cost balance with his own cost of donating the particular service. As far as the allocation policy is concerned, a provider selects to serve at a given period the candidate peer with the highest cost balance. As far as the server selection policy is concerned, peers at a given period select all the servers in a random order, seeking someone to serve them. In our model, we consider that every unsuccessful trial leads to an unsatisfied request. In order to have a direct comparison with our scheme, we consider that peers using ExtNetFav generate the same number of service requests per period, as with TAP-MS, and direct them to specific randomly chosen servers. Described scheme will be denoted as ENF in the figures of this paper, standing for ExtNetFav.

It is important to note that our model under which ENF and TAP-MS are compared in this paper, is a more demanding model than the one considered in [23]. First, [23] evaluates ENF in a system of peers with the same capability profiles, assuming that the capabilities of a given server are enough to satisfy any requesters' demands. However, the peers' satisfactions in our model depend on their heterogeneous capability and consumption profiles. ENF scheme is oblivious to peers' demands and therefore the peers' satisfactions under this scheme are expected to be lower than TAP-MS. Second, in [23] it is considered that at a given period peers select all servers in a random order. Therefore, the chance of an unsatisfied request during a period in this model is much less than in our model, where peers generate on average a specific number of service requests per period directed to specific peers. Under our policies, these specific peers are expected to be the appropriate servers for the given requests, following our trust-based server selection policy. However, this is not the case for ENF, where servers are selected randomly. Third, [23] assumes that peers can be either consumers or providers with a certain probability at a given period, while in our model peers can be consumers and producers at the same time (e.g., providing service type A and consuming service type B). Finally, [23] considers that the utility of a peer receiving a unit of a given service is greater than the cost of donating it to someone else, while our model considers a more demanding scenario according to which the cost of donating a unit of a given service is equal to the profit of obtaining it and equal to its value for the corresponding peer.

4.2.3 Tit-for-tat strategy of BitTorrent

In BitTorrent [20], for each torrent (i.e., a small file containing metadata about one or more files) there is a centralized entity, called the tracker, which keeps track of all the peers who download files specified in the torrent. Each peer that wants to download a file, finds the torrent of interest and connects to the associated tracker, who is responsible to return a random set of peers (called its neighbors) currently transferring pieces of the file(s) specified in the torrent. The downloader then establishes a connection to his neighbors and requests pieces which it does not have. In order to confront free-riding effect, BitTorrent uses a tit-for-tat (TFT) strategy to determine which neighbors to upload data to.We describe the basic version of BitTorrent [20], which is also adopted by the most popular BT client, Azureus.

Each BT peer performs the so called *choked algorithm* every 10 s, according to which all his neighbors are ranked based on their upload rate, and only the four top peers are unchoked (i.e., allowed to upload). Each peer further uses an *optimistic unchoke* policy every 30 s, wherein each peer unchokes one randomly chosen peer, regardless of his rank, in order to discover neighbors that might offer higher download rates than the peers it is currently downloading from. In this way he gives the chance to new peers to download their first block and prove their collaboration by reciprocating. The upload bandwidth of a peer is allocated equally among all unchoked peers. In Section 5.6 we explain how we apply our allocation framework over BitTorrent and provide comparison results with its tit-for-tat reciprocation strategy, which we denote by BT in the figures of this paper.

Finally, we describe some alternative allocation policies which share similarities with TAP and evaluate their performance, when our proposed reputation metric and update mechanism (whenever are needed) are used. They are described below for the case of n requests directed to peer z and their performance is exhibited in Section 5.1 in comparison with the performance of TAP.

4.2.4 Maximize the quality of service:

Under this policy a server allocates his upload capacity by using a progressive filling algorithm. He increases all competing peers' bandwidth at the same rate of $1/n$ until one or several competing peers hit their limits (demands). Then, the algorithm continues to increase the bandwidth of the remaining peers at the same rate as soon as all peers hit their limits or the upload capacity of the server is fully utilized. This policy will be denoted as max(x) in the figures of this paper.

4.2.5 Maximize the satisfactions of the peers:

This policy maximizes $\sum_{l=1}^{n}\left(x_{L_l}/D_{L_l}\right)$, under the constraints $\sum_{l=1}^{n} x_{L_l} \leq C_z$ and $x_{L_l} \leq D_{L_l} \forall l$. The solution can be found in a similar way as the solution of Eq. 2. In this way the policy seeks to maximize the satisfactions of all competing peers. This policy will be denoted as max(x/D) in the figures of this paper.

4.2.6 Maximize the quality of service based on the reputations of the peers:

This policy maximizes $\sum_{l=1}^{n} R_{z,L_l} x_{L_l}$, under the following constraints $\sum_{l=1}^{n} x_{L_l} \leq C_z$ and $x_{L_l} \leq D_{L_l} \forall l$. It gives more resources to more reputed peers. This policy will be denoted as max (Rx) in the figures of this paper.

4.3 Performance metrics

As we already stated in the introduction, one of the goals of our proposed policies is to motivate peers to cooperate and improve their contribution level by guaranteing that their satisfaction from their services' requests will be proportional to their contributions. Therefore, in order to examine whether the contributions of the peers in the system are correlated with the satisfaction they obtain from it, we define the global reputation and average satisfaction of a peer as follows.

The global reputation vector of a peer i at a period p is given by $\mathbf{GR}_i^p = \{GR_{i,1}^p, ...GR_{i,M}^p\}$, where $GR_{i,m}^p = \sum_{\forall z \neq i} R_{z,i,m}^p / (N-1)$, i.e., the global reputation of peer i for service m and period p is taken by averaging the individual opinions of every other peer z in the system for peer i and service m (local reputations) at period p. The global reputation of a peer actually reflects his capability and cooperation, as captured by the whole system (aggregate of peers). We clarify that under our scheme global reputations are only used for performance analysis; peers keep and use only local reputations.

We further calculate the average satisfaction vector $\mathbf{St}_i^p = \{St_{i,1}^p, ..., St_{i,M}^p\}$ of each peer i in the system. The average satisfaction of peer i for service m until period p, is given by summing his satisfaction ratios in each one of his transactions with all other peers for service m and averaging over the total number of his requests for service m till period p. In case of a single service being exchanged in the system, aforementioned vectors are reduced to a single value.

Another goal of our policies is to help peers find the appropriate partners with whom they can mutually benefit and improve their utility from the network, by exchanging different kind of services. We define the utility of peer i at a given period p, U_i^p , as follows:

$$U_i^p = \sum_{s=1}^{M} \left[V_{i,s} \times \frac{RT_{i,s}^p - RF_{i,s}^p}{RT_{i,s}^p + RF_{i,s}^p} \right] \Bigg/ \sum_{s=1}^{M} V_{i,s} \tag{6}$$

where, $RF_{i,s}^p$ and $RT_{i,s}^p$ are the average resources per period offered and obtained respectively by a given peer i for each service s till period p, measured in the units of each service. We divide by $RT_{i,s}^p + RF_{i,s}^p$ in order to weigh units of different services. The utility of a peer represents his net benefit by exchanging services in the system, given his own valuations for the various services.

Last, we define another metric, inspired by work in [23], to test the system under severe misbehavior. The average net benefit of a cooperative peer should be higher than that of a misbehaving one in the system who does not contribute any resources, i.e.,

$$\sum_{i \in C} \sum_{s=1}^{M} \left(V_{i,s} \times \left(RT_{i,s} - RF_{i,s}\right)\right) / |C|$$
$$- \sum_{j \in B} \sum_{s=1}^{M} \left(V_{j,s} \times RT_{j,s}\right) / |B| > 0 \Rightarrow$$
$$MI = \frac{\sum_{i \in C} \sum_{s=1}^{M} \left(V_{i,s} \times RF_{i,s}\right)}{\sum_{i \in C} \sum_{s=1}^{M} \left(V_{i,s} \times RT_{i,s}\right) - \frac{|C|}{|B|} \sum_{j \in B} \sum_{s=1}^{M} \left(V_{j,s} \times RT_{j,s}\right)} < 1 \tag{7}$$

where C, B, are the set of collaborative and misbehaving peers respectively and MI stands for Misbehavior Indicator and expresses how well the system marginalize misbehaving peers. MI must be positive and smaller than 1 in order for the system to block misbehaving peers. More specifically, the smaller MI is from 1, the better the marginalization of misbehaving peers.

5 Simulation results for the single service case

5.1 Peers with different capabilities and misbehaving peers

In Fig. 1 we consider peers who wish to dedicate different bandwidth capacities for uploading either because their bandwidth capabilities or their willingness to offer are different, and compare TAP with BRR. So, in a network of 100 peers, 20% dedicate 7 Mb/s, 20% 6 Mb/s, 20% 5 Mb/s, 20% 4 Mb/s and 20% 3 Mb/s for bandwidth uploading. We consider that all peers' demands vary uniformly between 1 and 7 Mb/s during simulation time. Each peer generates three requests per period and requests are directed with equal probability to each one of the rest of the peers (random server selection policy).

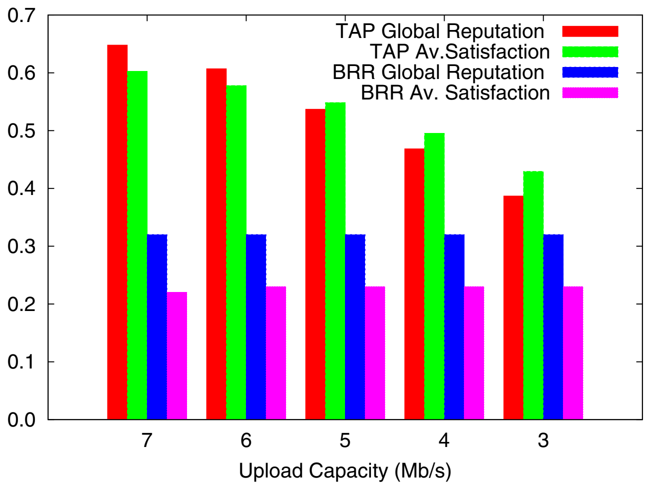


Fig. 1 Global Reputations and Av. Satisfactions of different capacity peers

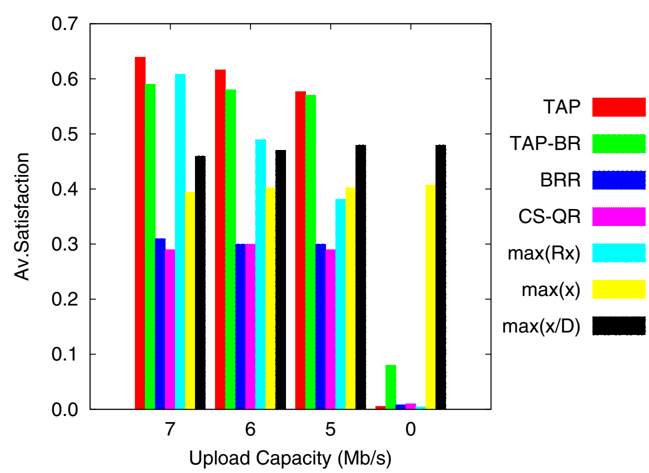
We see that with TAP the reputations of the peers are proportional to their capacities (dedicated upload bandwidth), and moreover their satisfactions are proportional to their reputations. However, BRR cannot differentiate peers according to their contribution levels and all cooperative peers have the same reputation (as defined in BRR scheme), and satisfaction.

In Fig. 2 the lower capacity (3 & 4 Mb/s) peers of the experimental case exhibited in Fig. 1 are now misbehaving by not contributing any resources to the network (0 Mb/s) and constitute 40% of the total number of the peers in the system. In Fig. 2a we can see that misbehaving peers do not succeed in gaining anything from the system (their satisfaction ratio is almost zero); however, they succeed in deteriorating the performance of the other peers who, unaware of the misbehavior, send them requests. The performance of contributive peers can significantly improve (45% improvement) if they use the trust-based server selection policy.

In Fig. 2b we see the satisfaction ratios of the different peers when the various allocation policies described in Section 4.2 are used along with the trust-based server selection policy. It is clear that only our proposed allocation scheme TAP, and max(Rx) achieve service differentiation according to the contributions of the peers. Our policy, though, achieves much better satisfaction ratios than max(Rx) for the same available bandwidth in the system. BRR scheme, although refrain misbehaving peers from exploiting the system, it cannot differentiate collaborative peers. In the same figure we can see the performance of two other policies, TAP-BR and CS-QR. The first one is the combination of our allocation policy TAP with the **B**inary **R**ating-based reputation metric of scheme in [8], while the second one is the combination of the **C**lient **S**election policy of scheme in [8] with our proposed **Q**uality-based **R**eputation metric. As we can see, the combined use of our reputation metric and proposed allocation policy is the best among all other policies.

We further evaluated our scheme for different populations N of 10^3-10^6 peers and the conclusions of the analysis remained the same. In each one of these populations the satisfactions of peers reached the same steady state with a convergence time in the order of N^2. However, even after only 1,000 rounds, for all aforementioned populations the service differentiation is succeeded. The only problem is that the average satisfactions of peers are a bit lower than the ones of the steady state. For very large populations, peers delay in "getting to know" each other and thus delay in identifying their partners. In reality, though, each one of the peers would collaborate with only a subset of the

(a)



(b)

Fig. 2 Av.Satisfaction of different capacity peers for 40% misbehaving peers and different **a** server selection and **b** allocation policies total population at a given duration of time according to its specific service requests or the number of peers who happen to be online at the particular duration time. Therefore, in reality, peers get to know their partners at different time periods. This case is simulated in the following section where new peers enter the system at different time periods.

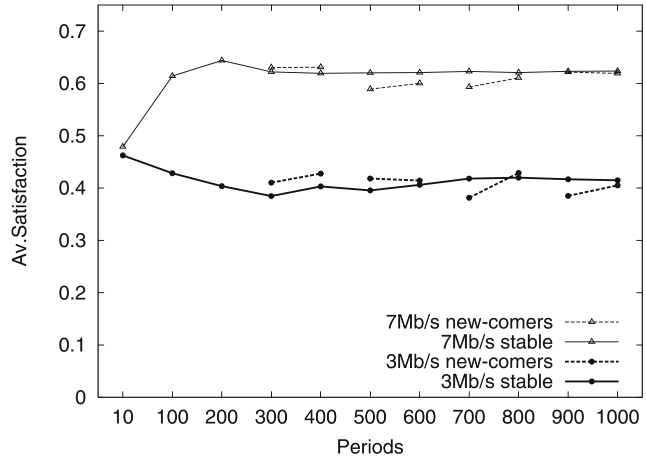
5.2 NewComers & WhiteWashers

In this section we examine the performance of the TAP policy, when peers periodically leave and join the network. As in the previous section, peers are categorized according to their upload capacity; 20% of them dedicate 7 Mb/s, 20% 6 Mb/s, 20% 5 Mb/s, 20% 4 Mb/s and 20% 3 Mb/s for bandwidth uploading and all peers generate three requests per period. However, every 200 periods 50% of each category (capacity) peers leave the system and are replaced by new identity peers of the same capacity with those who left. In this way, we maintain the same analogy of capacity peers in the system and better investigate the performance of the system compared to the static one, where no arrivals or departures occur. All peers use the TAP and the proposed trust-based server selection policies.

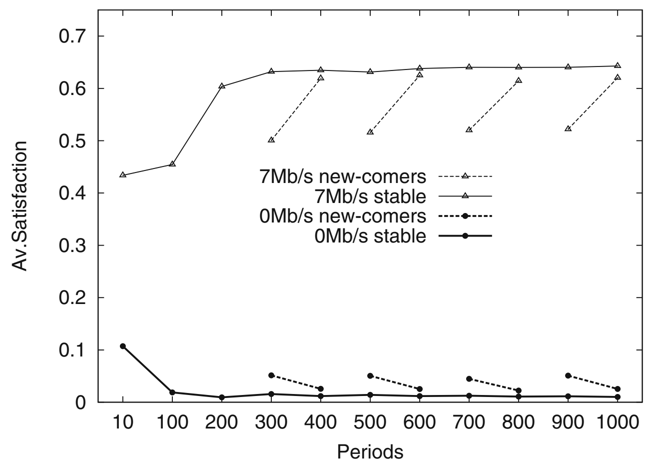
In Fig. 3a we see the average satisfactions of the newcomers and the stable peers (the ones who remain in the system) with capacities 7 and 3 Mb/s, as the system evolves in periods. For distinctness reasons, we did not exhibit the performance of the other capacity peers in the same graph since the same conclusions apply for them, as well. We can see that the performance of the permanent peers is not affected by the presence of the newcomers, while the newcomers' performance is very close to the one of the steady state, despite their short life in the system.

Next, we consider that lower capacity peers (3 & 4 Mb/s) are misbehaving (contributing zero resources) and, as above, every 200 periods 50% of each category peers are replaced by new identity peers of the same capacities. New identity peers may be misbehaving ones who have left the network and appeared with new identities (whitewashers) in order to exploit the system. However, as we see from Fig. 3b neither stable misbehaving peers nor new identity misbehaving peers can take advantage of the system; their satisfaction ratio is almost zero.

The performance of the stable 7 Mb/s capacity peers is not influenced by the presence of newly arriving misbehaving and other peers. Moreover, the newcomers of 7 Mb/s capacity can almost achieve the performance of the steady state in the end of their lives in the system. As we already noted, new peers stay for 200 periods in



(a)



(b)

Fig. 3 Average satisfaction of newcomers and stable peers of different capacities for **a** no misbehaving peers and **b** 40% misbehaving peers in the system

the network. During their acquaintance duration (100 periods), they direct their requests with equal probability among all others and their average satisfaction till the end of this acquaintance duration is only 0.51 as we see from Fig. 3b, due to the presence of the misbehaving peers. After the acquaintance duration, new peers start directing their requests to their most reputed servers and thus in the end of their lives in the system (after more 100 periods) they manage to improve their satisfaction ratio to 0.61. If they stayed in the network for a longer time, their performance would even more improve to reach the one of the steady state.

5.3 Formation of cooperation groups (Coalitions)

Next, we consider that 50% of peers have both 7 Mb/s capabilities and demands (strong peers) in each one of their requests and 50% of peers have 3 Mb/s

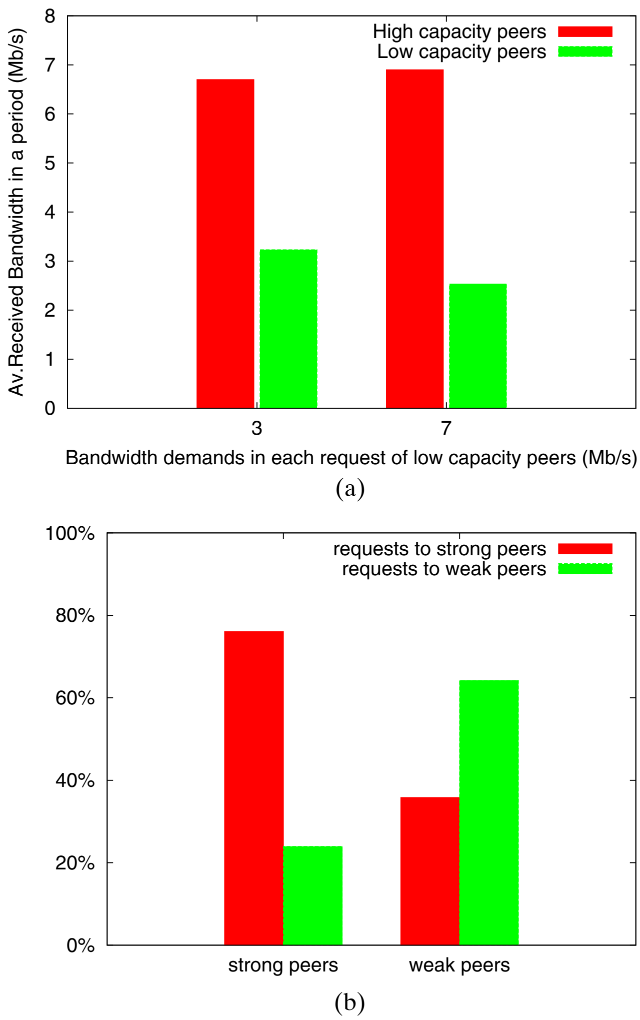


Fig. 4 **a** Av. Received Bandwidth per period for high and low capacity peers. **b** Percentage of requests directed from and to each category of peers

capabilities and demands (weak peers). All peers generate three requests per period and they use the TAP policy. In Fig. 4a we see that high capacity peers get 6.7 Mb/s on average during a period, while low capacity peers get only 3.23 Mb/s. If, however, low capacity peers had high demands of 7 Mb/s, their average bandwidth in a period would only be 2.53 Mb/s, while the average bandwidth in a period for high capacity peers would be 6.9 Mb/s (see Fig. 4a). Low capacity peers are punished for their excess demands. They should be able to provide a proportional service to what they ask for, because in possible competition with higher capacity peers they lose.

In Fig. 4b we see that peers were self organized in two coalitions; the one of the strong peers and the other of the week peers. The highest percentage of the requests of strong peers was directed to strong peers, while the highest percentage of the requests of weak peers was directed to weak peers. Strong peers can accommodate the high demands of strong peers and that is why strong peers direct their requests to other strong peers. On the other hand, weak peers when competing with strong peers, lose (get what remains from strong peers). Thus, they prefer to send their requests to other weak peers (weak peers have higher local reputation to them). It is very important that although peers are unaware of the capabilities of the other peers, they efficiently recognize the appropriate traders with whom they can improve their utility and are self-organized in coalitions. We have also detected these coalitions in all other cases of heterogeneous (capacity and demands) peers. In Fig. 4b we see that still some requests are directed to opposite groups; this is mainly due to the acquaintance duration time, under which peers send their requests randomly to one another.

5.4 Strategic peers

In this section we consider that peers are strategic, i.e., they seek to maximize their satisfaction with the least possible contributions. All peers have the same upload capabilities of 7 Mb/s and peers' demands vary uniformly between 1 and 7 Mb/s. As we already saw, by using the proposed allocation policy TAP, the average satisfaction ratio of a given peer is proportional to his global reputation. Therefore, a peer knows that as soon as his average satisfaction ratio reaches a certain high value, his global reputation is high enough and thus he can take advantage of it to start misbehaving.

We investigate two possible scenarios. In the first one, each time a peer reaches a certain average satisfaction ratio (threshold) he progressively decreases his dedicated capacity for uploading (here we consider that the capacity is decreased by steps of 1 Mb/s). Once his average satisfaction ratio falls down this threshold, he progressively increases his capacity till its maximum value (determined by the peer's physical upload capacity), or till he perceives the desired average satisfaction ratio (in this case he starts, once again, decreasing his dedicated capacity). We will refer to this scenario as the *progressive strategy*.

In the second scenario, once a peer's average satisfaction ratio reaches the specified threshold, he does not offer any bandwidth at all; however once his average satisfaction ratio falls down the threshold he starts collaborating again by contributing his maximum upload capacity. We will refer to this scenario as the *steep strategy*. In both scenarios, peers start contributing their whole capacity and dynamically adjust their contributions according to their perceived satisfaction ratio.

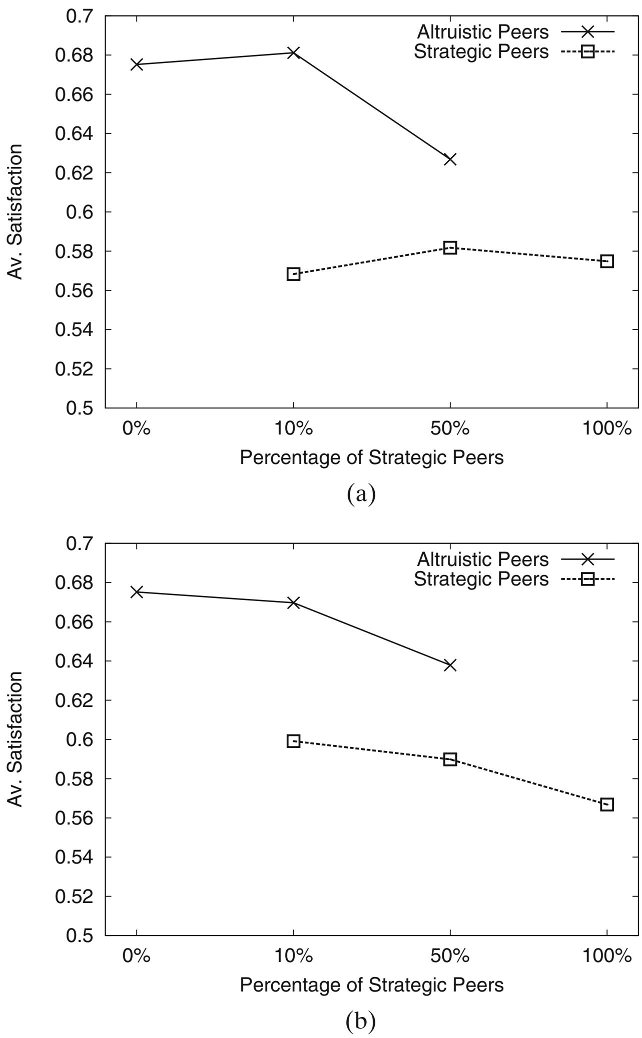


Fig. 5 Average satisfaction for different percentage of strategic peers when the latter adopt **a** the progressive strategy or **b** the steep strategy

In Fig. 5 we see the average satisfaction of altruistic and strategic peers for different percentage of strategic peers in the system when (a) the progressive and (b) the steep strategy are adopted by the strategic peers. We refer to peers who contribute their maximum upload capacity during their lifetime in the network as altruistic and to peers who vary their upload capacity as strategic ones. We further consider that the satisfaction threshold under which peers start misbehaving is 60%.

From Fig. 5 it is obvious that altruistic peers gain a much better satisfaction ratio than strategic peers, even when the latter constitute the half portion of the network. If all peers were altruistic (case 0% in Fig. 5), their average satisfaction ratio would be 17,5% and 19,1% greater than if they were all progressive strategic and steep strategic peers (case 100% in Fig. 5) respectively. It is further remarkable to observe that the average satisfaction of the strategic peers does not exceed their satisfaction threshold (60%), indicating that peers indeed cannot receive more resources than what they have contributed to the network.

5.5 Peers with heterogeneous request generation rates

In this section we investigate a system of 100 peers with heterogeneous request generation rates. All peers have the same upload capacity of 7 Mb/s and their demands are uniformly distributed between 1 and 7 Mb/s during simulation time. However, 50% of the peers are high rate (HR) peers generating four requests per period, while 50% of them are lower rate (LR) peers generating 2 requests per period. In Table 1 we can see the average bandwidth per request, the average bandwidth per period and the average satisfaction of the high rate and the low rate peers, when policies BRR, TAP and ETAP with $a = 1$ or $a = 2$ are used by all peers.

We note that with BRR and TAP, HR and LR peers gain almost the same average bandwidth per request and that HR peers gain almost twice as much bandwidth per period as LR peers (as they produce twice as much requests per period as LR peers). These policies are not fair for LR peers and actually motivate peers to produce a lot of requests in the system to improve their performance, in cost of other contributive peers. In order to cope with this case we use ETAP. It is remarkable that by using ETAP with $a = 1$, HR peers have a smaller average bandwidth per request than LR peers, leading to just a slight more bandwidth per period than LR peers. ETAP with $a = 2$ is even more restrictive for HR peers, leading to the same average bandwidth per period for both HR and LR peers.

In Table 2 we consider that (a) HR peers have a capacity of 7 Mb/s while LR peers have a capacity of 3 MB/s and that (b) HR peers have a capacity of 3 Mb/s while LR peers have a capacity of 7 Mb/s. We can see that when high rate peers are the strong capacity ones, even with ETAP and $a = 2$ they receive better service per period than lower capacity ones since they provide more resources than them. On the other hand, when HR peers are the low capacity ones, their performance

Table 1 Performance of HR and LR peers for different policies

HR Cap:7Mb/s	AvBand/Req (Mb/s)		AvBand/prd (Mb/s)		AvSatisf	
LR Cap:7Mb/s	HR	LR	HR	LR	HR	LR
BRR	2.15	1.90	8.60	3.80	0.33	0.31
TAP	2.19	1.94	8.76	3.88	0.69	0.64
ETAP (a=1)	1.70	2.90	6.80	5.80	0.57	0.82
ETAP (a=2)	1.56	3.19	6.24	6.38	0.51	0.87

Table 2 Performance of HR and LR peers for different policies and capacities

	AvBand/Req (Mb/s)		AvBand/prd (Mb/s)		AvSatisf	
	HR	LR	HR	LR	HR	LR
HR Cap:7Mb/s LR Cap:3Mb/s						
BRR	1.60	1.52	6.40	3.04	0.23	0.23
TAP	1.80	1.10	7.20	2.20	0.60	0.43
ETAP (*a*=1)	1.53	1.63	6.12	3.26	0.54	0.56
ETAP (*a*=2)	1.31	2.05	5.25	4.11	0.48	0.64
HR Cap:3Mb/s LR Cap:7Mb/s						
BRR	1.44	1.39	5.76	2.78	0.21	0.21
TAP	1.43	1.83	5.71	3.66	0.51	0.61
ETAP (*a*=1)	1.08	2.52	4.33	5.04	0.39	0.74
ETAP (*a*=2)	1.04	2.60	4.17	5.20	0.36	0.76

is much restricted by ETAP policies. It is like punishing weak peers for their excess requests, since they are not able to provide the relative services to the system.

These results reveal the fairness of the enhanced proposed allocation policy, which guarantees that peers will be able to receive resources in proportion to their contributions. We further reach the same conclusion for any other cases of request rates, peer populations and capabilities that we examined. When the request rates of all peers are the same, ETAP is almost identical with TAP. We say "almost" because when the trust-based server selection policy is used, ETAP slightly favors the most reputed peers since most of the requests are directed there.

5.6 Applying TAP and ETAP in BitTorrent

We simulate the basic version of BitTorrent (BT) [20], as described in Section 4.2.3, and examine the system as soon as a neighborhood of connected peers participating in the same torrent is set. Here, we consider a neighborhood of 50 connected peers, in compliance to the typical set of peers returned by the tracker. All connected peers send requests for blocks of one or more files, specified in the torrent, to one another. Since our focus is in the performance evaluation of the allocation policy, we omit the block selection algorithm used in BT and consider that the connected peers are always interested in blocks of each other.

Under this setup, we further evaluate the performance of peers who perform our policies. The default values used for the initial reputations and misbehavior threshold are as described in Section 4.1. Periods in this setup lasts for 10s in compliance to the time interval between successive performances of BT *choked algorithm*. Simulations both for BT peers and our reputation-aware peers were run for 100 periods, which proved to be enough to represent the steady state performance values of the peers, and the acquaintance duration time for our policies was set to 10 periods. Since the goal in BitTorrent is to maximize the average received bandwidth per period in order to minimize the file's download time, we consider that the peers' demands under our reputation-aware scheme are set by the software equal to a flag indicating that they request the maximum possible bandwidth. Under this flag, $D(t)$ are set equal to 1 in the reputation calculation formula (1), i.e., reputations are calculated based on the received amount of bandwidth. Finally, under our policies we consider that the request generation profile of peers is set from the software equal to 5, i.e., each peer sends 5 requests per period for blocks of files. This better complies with the basic version of BitTorrent, where each peer serves five peers per period.

A well-known problem of the BitTorrent's reciprocation strategy is that peers can obtain some resources even when they do not contribute anything in the system and this fosters misbehaving activity [34]. Optimistic unchoke policy is considered to be one of the reasons for that [32]. To see how our policies can overcome this deficiency of BT, we consider a scenario of 40% misbehaving peers, 20% peers who dedicate 7 Mb/s for uploading, 20% 6 Mb/s and 20% 5 Mb/s. In Fig. 6 we can see that when our policies are used (TAP and ETAP with parameter a=1), misbehaving peers cannot receive any resources from the system, in contrast to BT policy. Furthermore, under BT , some peers seem to contribute more resources (upload capacities) than those they obtain in a given period and some less, indicating unfairness. Under our policies, though,

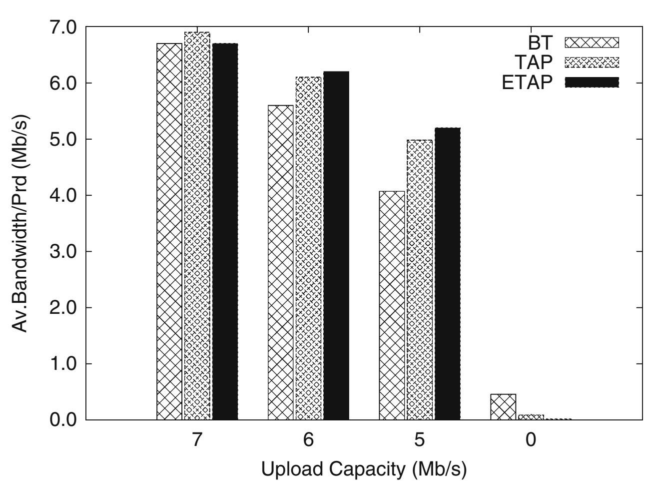


Fig. 6 Performance of misbehaving peers in a BT scenario

peers contribute on average as much resources as they obtain per period. Please note that the performance of ETAP is close to TAP, since the request generation profiles of all peers are set by the software and are the same. The slight differences are due to a higher amount of requests directed to more reputed peers, through our trust-based server selection scheme. Under ETAP, peers do not have any motivation to hamper the software and generate more requests than the others, since from Eq. 4 they know that their performance will be restricted (see also results of previous section).

Another well known deficiency of the memoryless reciprocation mechanism of BitTorrent is the fact that it does not provide incentives for seeds (peers who have downloaded a file) to remain in the system and contribute resources to others [31, 33]. Here, we show how our proposed scheme can motivate peers who have already downloaded a file to remain and contribute to the system. We study 50 connected peers with physical capacities of 7Mb/s who download blocks of files. One of these peers, though, finishes downloading a file in period 100 but remains in the system and contributes his upload capacity for another 100 periods. After this duration, i.e., in the 200th period, he starts downloading once again (e.g., other files).

In Fig. 7 we see the performance of the seed before and after his 100 uploading periods. With BT and TAP policies the seed's performance is almost the same before and after his uploading periods; thus he does not earn anything by remaining in the system and offering his total physical capacity to others. On the other hand, when ETAP is used, the seed's performance is improved when he starts downloading again and this improvement remains for many periods. This is due to the fact that during the periods that the seed uploads

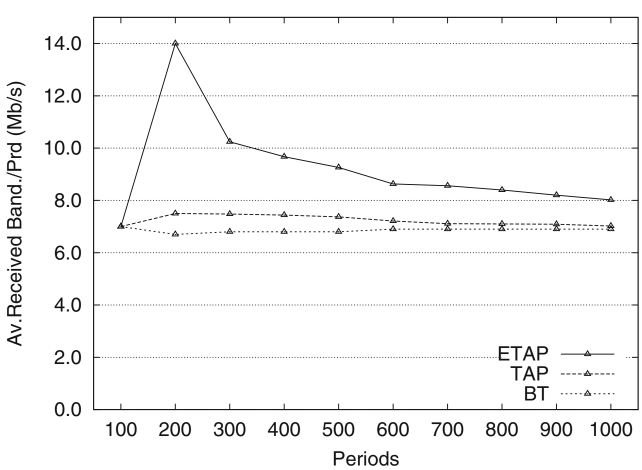


Fig. 7 Performance of seed

to other peers, the number of requests directed to him compared to the ones he has sent to others significantly increases. Therefore, from Eq. 4 it is obvious that he is going to be given high priority in his future requests.

We note, however, that Fig. 7 is only indicative of the motivation a seed can have to remain and contribute in the system, since his actual gains depend on the frequency with which he transacts with the same peers he was uploading to. In BT, for example, a new plugin could help peers select their neighbors among the ones who have the file of interest; a seed would select the ones he was uploading to, in order to improve his performance.

6 Simulation results for the multiple services case

In this section we present the results from the analysis of a system where two different services are being exchanged, as it is the most simple case and easily presented. We further experimented with different cases of more than two services being exchanged in the system, and the conclusions of the analysis remained the same.

In our first experiment we consider that the 100 peers of the system are separated in 10 categories of 10 peers. Peers of the same category have the same capabilities in providing each service. In Table 3 we see the capabilities of peers in each category i for service 1 and service 2 ($C_{i,s}$ $s = 1, 2$). Variable i represents a given category. Peers of category 1 can provide the best service 1 but no service 2, while peers of category 10 provide the best service 2 among other peers but no service 1. The rest of the peers provide one service better than the other or provide both services equally well compared with all other peers' performance (e.g., peers of category 6). We consider that all peers have the same consumption profile, i.e., demands of all peers take values with equal probability in the set {1,2,3,4,5,6,7} for service 1 and in the set $\{1,3,5,7,9,10,11,13\}\times 10^2$ for service 2 and each peer produces three requests per period for service 1 or 2 with equal probability, unless otherwise stated. Service 1 could be bandwidth, for example, measured in Mb/s, while service 2 could be CPU cycles. We just use a case of different units to test the performance of our system which, though, works for any kind of services measured in any units.

Table 3 Capabilities of each category peers

Category i	1	2	3	4	5	6	7	8	9	10
$C_{i,1}$	9	8	7	6	5	4	3	2	1	0
$C_{i,2}(\times 10^2)$	0	5	6	7	8	9	10	11	12	13

Table 4 TAP-MS Performance metrics of all categories peers for $V_i = \{1, 1\} \forall i$

Ct i	$GR_{i,1}$	$GR_{i,2}$	$St_{i,1}$	$St_{i,2}$	$r_{i,1}$	$r_{i,2}$	U_i
1	0.77	0.01	0.58	0.60	17.9	0.30	0.40
2	0.74	0.57	0.72	0.75	16.0	7.60	0.10
3	0.74	0.62	0.71	0.77	15.1	8.50	0.10
4	0.71	0.63	0.74	0.76	12.0	10.0	0.10
5	0.68	0.66	0.73	0.73	9.70	8.90	0.05
6	0.65	0.67	0.75	0.74	9.10	11.0	0.10
7	0.55	0.66	0.74	0.67	8.20	12.5	0.09
8	0.45	0.72	0.68	0.72	7.90	14.2	0.16
9	0.31	0.72	0.67	0.65	4.60	13.3	0.19
10	0.00	0.74	0.55	0.59	0.50	13.6	0.40

Table 5 ENF Performance metrics of all categories peers for $V_i = \{1, 1\} \forall i$

Ct i	$St_{i,1}$	$St_{i,2}$	$r_{i,1}$	$r_{i,2}$	U_i
1	0.39	0.49	0.10	0.10	0.27
2	0.39	0.48	0.10	0.10	−0.15
3	0.42	0.48	0.10	0.10	−0.15
4	0.44	0.48	0.10	0.10	−0.15
5	0.42	0.47	0.10	0.10	−0.15
6	0.45	0.48	0.10	0.10	−0.17
7	0.48	0.47	0.10	0.10	−0.06
8	0.47	0.50	0.10	0.10	−0.01
9	0.48	0.50	0.10	0.10	0.08
10	0.46	0.49	0.10	0.10	0.27

6.1 Homogeneous service value profiles

In Table 4 we see the various performance metrics in the steady state, when all categories have the same value profile, i.e., $V_i = \{1, 1\} \forall i$. Variables $r_{i,1}$, $r_{i,2}$ are the average percentage of requests directed to peers of category i for service 1 and 2 respectively. The following tables exhibit the average values of the various performance metrics over all peers of a given category. We see that the reputation of each category for each service is in proportion to the category's capability in providing the service. This shows that the reputation vectors satisfactorily capture the capabilities of the peers. The above can also be shown from $r_{i,1}$, $r_{i,2}$ metrics. We see that the highest percentage of requests for service 1 is directed to the first category of peers who can best provide it.

In Table 4 we further see that peers have exchanged different type of services in the system as although category 1 peers cannot provide service 2, their satisfaction from requests for this service is good since they exchanged it with service 1, indicated by $GR_{1,1}$. Moreover, the average utilities of all peers are positive, i.e, peers benefit by their transactions. In general, $St_{i,2} > St_{i,1}$; this is due to the fact that the supply for service 2 is more than for service 1 compared to the corresponding demands in this case study.

In Table 5 we see the performance metrics when ENF scheme is applied, apart from global reputations which are not defined in this scheme. By comparing the performance metrics in two tables we can see that both the satisfactions and utilities of peers are higher in our scheme. While we evaluated the two schemes under the same scenario, TAP-MS seems to make a better allocation of the system resources.

In Fig. 8 we further see the dynamics of TAP-MS metrics for category 1 peers, as the system evolves in periods. After the first 100 periods (acquaintance duration), peers start selecting their servers based on their reputation, according to the trust-based server selection policy. We see that the utility and the satisfactions of the peers progressively increase after the initial 100 periods, indicating that peers improve their coalitions to increase their benefit by their transactions.

6.2 Random service value profiles (RSP)

In Table 6a we see the average satisfactions of all categories' peers for randomly chosen service values for TAP-MS and ENF schemes and in Fig. 9a, b the utilities of all categories' peers as the system evolves in periods for TAP-MS and ENF, respectively. For TAP-MS it is very interesting to see that, although initially in the system many peers have negative utilities, after the acquaintance duration time the utility of all categories increase. Figure 9a shows that although the utility of category 8 increases, it still remains negative. This most probably happens because category 8 peers put a very small value on the service that they are weak in producing it compared to the very high value they put on

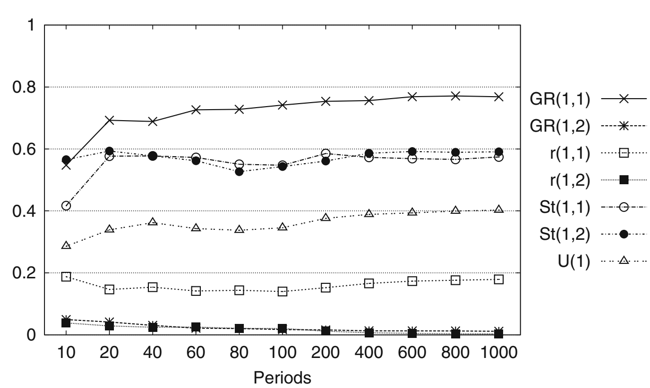


Fig. 8 Performance metrics for category 1 peers as the system evolves

Table 6 Satisfactions for (a) RSP and (b) CSP for TAP and ENF policies

Cat. i	$St_{i,1}$ TAP-MS	$St_{i,2}$ TAP-MS	$St_{i,1}$ ENF	$St_{i,2}$ ENF	$V_{i,1}$	$V_{i,2}$
(a)						
1	0.45	0.49	0.32	0.39	0,99	0,38
2	0.68	0.70	0.34	0.40	0,38	0,37
3	0.69	0.72	0.33	0.4	0,33	0,45
4	0.67	0.72	0.33	0.41	0,31	0,96
5	0.71	0.71	0.31	0.41	0,29	0,94
6	0.66	0.73	0.33	0.42	0,10	0,80
7	0.71	0.72	0.33	0.42	0,09	0,71
8	0.70	0.72	0.34	0.42	0,06	0,94
9	0.70	0.69	0.35	0.42	0,26	0,47
10	0.62	0.67	0.36	0.4	0,50	0,44
(b)						
1	0.49	0.63	0.30	0.41	1	2
2	0.65	0.76	0.30	0.39	1	2
3	0.68	0.76	0.36	0.38	1	2
4	0.66	0.76	0.36	0.39	1	2
5	0.69	0.71	0.33	0.40	1	1
6	0.72	0.71	0.36	0.39	1	1
7	0.73	0.67	0.38	0.38	2	1
8	0.70	0.67	0.38	0.39	2	1
9	0.67	0.63	0.39	0.39	2	1
10	0.61	0.55	0.38	0.37	2	1

the service that they are strong (which is quite unusual, as highlighted in Section 3.1). From Eq. 6 it is obvious that these peers will have negative utility, unless they could be able to choose among many more peers with a higher capability than theirs in providing service 2. The important conclusion of this analysis is that proposed policies help peers in recognizing trading partners with whom they can all benefit from their cooperation and improve their utility from the network. The maximum utility that a peer can gain, depends on the service value profiles of all the peers and the available resources of the system. When ENF scheme is used, the utilities of all peers are lower than those in TAP-MS even in the early peers' transactions. Actually, most of them are negative, revealing the peers' failure to transact with the appropriate partners. Our simulations have shown that the same conclusions also apply for different cases of peers' demands, capabilities and network load.

6.3 Service value profiles correlated with capability profiles (CSP)

Table 6b and Fig. 9c, d show the results for the case that the service values are related to the service capabilities, i.e., categories 1–4 peers put a double value on service 2, which they are weak in producing it, categories 7–10 put a double value on service 1, which they are weak in producing it and category 5 and 6 put equal values on both services which they can equally well provide. Once more we can see from the comparison of related figures that the utilities of all categories peers are much

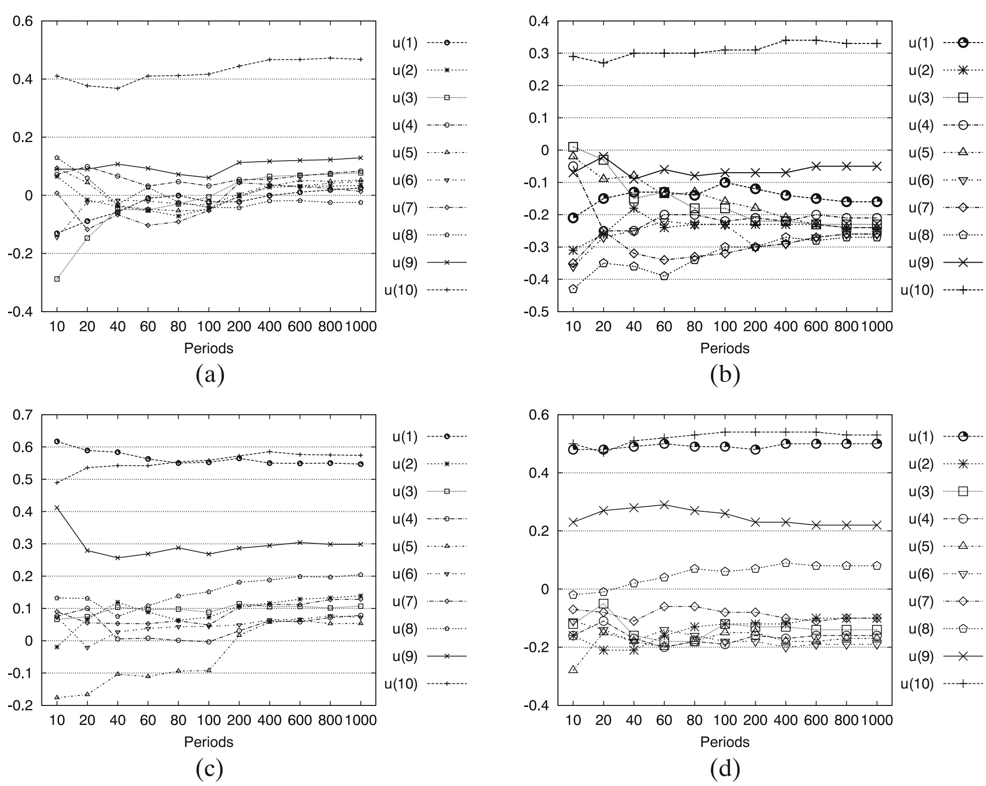


Fig. 9 Utilities of all categories peers for **a** TAP-MS and RSP, **b** ENF and RSP, **c** TAP-MS and CSP and **d** ENF and CSP

improved when our scheme is used instead of ENF, indicating that with our policies we can achieve a more efficient allocation of the system's resources.

6.4 Peers utilities in bigger populations

We were interested in examining the performance of the system and the formation of coalitions in much bigger peers' populations. Therefore, we performed experiments with a population of 10^6 peers splitting in the 10 equal size categories of Table 3, and used the same system parameters as we did for the population of 100 peers. In Table 7 we can see the utilities of all category peers after 1000 rounds both for the case of RSP examined in Section 6.2 and the case of CSP examined in Section 6.3 for the two populations. Interestingly, we can see that even in such big populations, peers succeed in finding the appropriate partners for them in order to benefit from the system (positive utilities). For the RSP case some peers' categories (6,7,8) have very small negative values of utilities close to 0. As we already reported in Section 6.2, this is due to the fact that these peers put a very small value on the service that they are weak in producing it compared to the very high value they put on the service that they are strong. For the CSP case all categories peers have positive utilities and it is notable that peers who are the strongest in providing a certain service receive the highest utility among others. This is even more pronounced in the bigger population.

6.5 Dynamic service value profiles

In previous sections we showed that under TAP-MS peers with different service evaluation and capability profiles tend to form coalitions in order to improve their utility. In this section we would like to see how these coalitions are dynamically reformatted when the peers' evaluations of the services vary through time. We consider the extreme case that all peers change their service evaluation in the 500^{th} period and we see how fast peers adapt to this change. In the beginning of the simulation time, the service evaluation profiles of all category peers have been selected randomly and can be seen in the first two columns of Table 8. From the 500^{th} period and beyond, the new service evaluation profiles of the peers, which have also been randomly selected, can be seen in the next two columns of Table 8.

In Fig. 10 we can see the utility of all category peers as the time evolves. It is indicative that after the service reevaluation in the 500^{th} period, peers' utilities adapt to the new network conditions. Just 100 periods were enough for peers' utilities to reach the new steady state of the system. We can further see that category 1 peers benefit by changing their service value profile. Actually after the 500^{th} period they give an even higher value to the service that they cannot produce by themselves, thus they tend to even more favor peers who can produce it. Not all category peers seem to benefit from the new service evaluation establishment. As we have already underlined in previous sections, peers' benefit of the system further depends on the capabilities and service evaluation of the other peers in the system. Our policies help peers to recognize and trade with partners with whom they can mutually benefit. If no such partners exist in the network, the peers' utilities can not be very high.

6.6 Performance of misbehaving peers

Next, we consider another category consisted of misbehaving peers, category 0. Category 0 consists of 50% of the total peers in a system of 200 peers, i.e., 100 peers are collaborative, splitting in each one of 1–10 categories, as in previous examples and the rest 100 of category 0 peers are misbehaving. For simplicity, all

Table 7 Utilities of each category of peers for different populations

Cat. i	RSP		CSP	
	$N=10^2$	$N=10^6$	$N=10^2$	$N=10^6$
1	0.03	0.18	0.55	0.63
2	0.03	0.02	0.14	0.08
3	0.08	0.04	0.11	0.05
4	0.08	0.03	0.08	0.02
5	0.05	0.01	0.05	0.01
6	0.05	−0.01	0.07	0.01
7	0.01	−0.03	0.13	0.08
8	−0.02	−0.01	0.20	0.14
9	0.13	0.12	0.30	0.30
10	0.47	0.47	0.57	0.64

Table 8 Satisfactions for varying random service value profiles

Cat. i	$V_{i,1}$	$V_{i,2}$	$V_{i,1}$ 500^{th} prd	$V_{i,2}$ 500^{th} prd
1	0.19	0.43	0.04	0.96
2	0.96	0.92	0.88	0.42
3	0.15	0.48	0.45	0.36
4	0.76	0.47	0.43	0.86
5	0.31	0.74	0.97	0.00
6	0.51	0.71	0.57	0.01
7	0.78	0.30	0.21	0.72
8	0.92	0.10	0.52	0.81
9	0.25	0.70	0.30	0.82
10	0.25	0.67	0.63	0.59

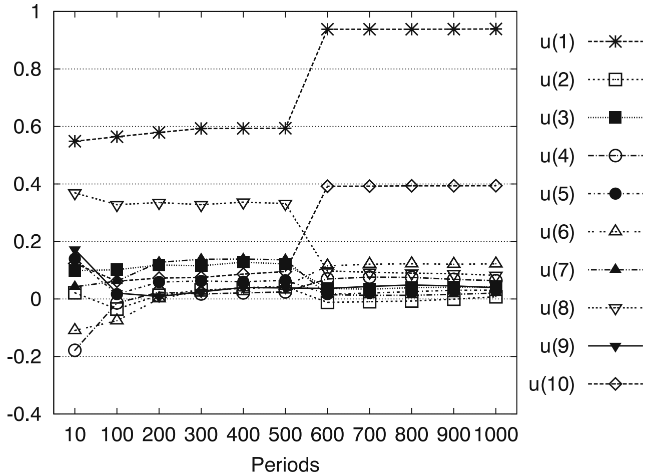


Fig. 10 Utilities of all categories peers for dynamic service value profiles

service values were set to 1. In Fig. 11, it is remarkable that although satisfactions of misbehaving peers for each service converge to 0, misbehavior indicator MI is initially very high, because several cooperative peers lose utility by sending their requests to misbehaving peers. However, after the acquaintance duration, MI systematically improves, reaching the value of 0.5. This indicates that misbehaving peers are blocked by proposed policies, even when they constitute 50% of the total population. We also note that when we evaluated the performance of ENF under this scenario we observed that in every simulation run, indicator MI had significant varying negative values, around −20. Although the satisfaction of misbehaving peers were considerably low, they were slightly better than those of cooperative peers, 0.27 versus 0.23. Misbehaving peers, under this scheme, succeed in exploiting the network. Actually, as admitted by the authors of [23]

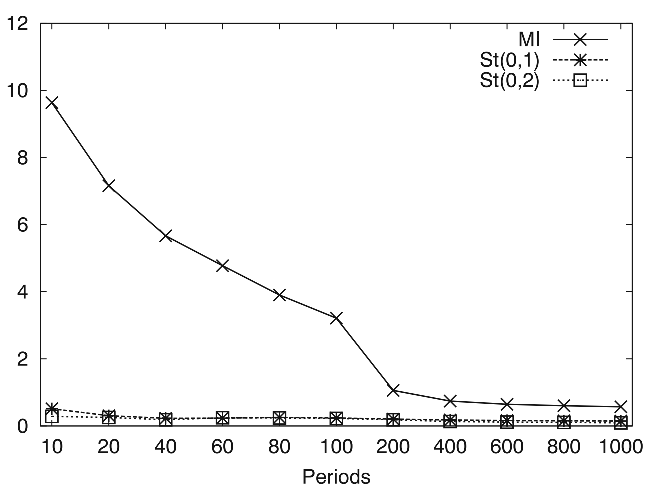


Fig. 11 Performance of misbehaving peers

themselves, when the number of misbehaving peers is high and the peers' request generation profile is low, peers under ENF have a high probability of participating in interactions that will not be profitable. We believe that this is due to two main factors: the random selection of providers and the lack of a mechanism to deny service to misbehaving peers, even when these are the sole competitors for a resource. The divergence of our conclusions than those in [23] for ENF about the marginalization of misbehaving peers arise from the fact that we evaluate their scheme under a more demanding model than the one used in [23], as we explain in detail in Section 4.2.2.

6.7 Heterogeneous request generation profiles

Till now we have considered that all peers have the same request generation profile. Next, as we did for the single service case we examine a system of peers with different request generation rates. In Table 9 we consider 100 peers, 50% of which are low rate (LR) peers generating two requests per period and 50% of which are higher rate peers (HR) generating four requests per period with equal probability for each service. All peers have the same capability profile $\mathbf{C} = \{7, 13 \times 10^2\}$, the same demand profile $\mathbf{D} = \{7, 13 \times 10^2\}$ and all service values are set to 1. We investigate the average received resources per period for each service when ENF, TAP-MS and ETAP-MS are used. When ENF or TAP-MS is used, HR peers receive twice more resources on average per period than LR peers, while ETAP-MS restricts the behavior of HR peers. For $a = 2$ both HR and LR peers receive almost the same amount of resources per period. ETAP-MS provides a fairer treatment to peers, as equally cooperative peers receive almost the same amount of resources in a loaded network. ETAP-MS could be particular helpful in case of peers trying to exploit the network by generating a huge amount of requests for the various services. It is notable to see that under this scenario which considers peers of homogeneous capability, demand and service evaluation profiles, ENF performs almost identically with TAP-MS.

Table 9 Average received resources per period for HR and LR peers

Average Resources/prd	Service 1		Service 2 ($\times 10^2$)	
	HR	LR	HR	LR
ENF	7.78	3.52	14.5	6.71
TAP-MS	7.83	3.57	14.6	6.74
ETAP-MS (α= 1)	5.93	5.40	11.4	9.96
ETAP-MS (α= 2)	5.72	5.66	10.3	10.5

7 Conclusions

In this paper we present a trust-based framework to fairly regulate the exchange of one or multiple services in p2p like systems of peers with different contribution, consumption and service evaluation profiles. Our simulation studies show that our framework outperforms previous work and leads peers to cooperation and self-organization in coalitions in order to improve their utility. In case of only one service being exchanged, peers tend to cooperate with similar capability peers, while in case of more than one services being exchanged, peers' coalitions depend on their service evaluation and capability profiles; peers favor those who can better satisfy their service needs. We also observed that the proposed policies shield the system from (a) misbehaving peers, (b) strategic peers who seek to maximize their utility with the least possible effort/contributions and (c) peers who generate much more requests than the others without contributing a proportional amount of resources. The aforementioned peers neither can exploit the system nor can harm the performance of the other peers. In fact, peers can only receive resources in proportion to their contributions; the higher demands and the more requests a peer has, the more resources he will have to contribute in order to satisfy his needs in a loaded network.

References

1. Satsiou A, Tassiulas L (2007) A trust-based exchange framework for multiples services in P2P systems. In: Proc. of the 7th IEEE international conference on P2P computing, Ireland, pp 45–52
2. Hardin G (1968) The tragedy of the commons. Science 162:1243–1248
3. Hughes D, Coulson G, Walkerdine J (2005) Freeriding on Gnutella revisited: the bell tolls. IEEE Distrib Syst Online 6(6):1
4. Vishnumurthy V, Chandrakumar S, Sirer EG (2003) KARMA, a secure economic framework for peer-to-peer resource sharing. In: Proc. of the workshop on the economics of peer-to-peer systems, Berkeley, California
5. Ma RTB, Lee SCM, Lui JCS, Yau DKY (2006) Incentive and service differentiation in P2P networks: a game theoretic approach. IEEE/ACM Trans Netw 5(14):978–991
6. Feldman M, Lai K, Stoica I, Chuang J (2004) Robust incentive techniques for peer-to-peer networks. In: Proc. of the 5th ACM conference on electronic commerce, ACM, New York, pp 102–111
7. Ma H, Leung H (2006) A demand and contribution based bandwidth allocation mechanism in P2P networks: a game-theoretic analysis. In: Proc. of the 20th international conference on advanced information networking and applications
8. Papaioannou TG, Stamoulis GD (2006) Reputation-based policies that provide the right incentives in peer-to-peer environments. Comput Netw (Special Issue on Management in Peer-to-Peer Systems: Trust, Reputation and Security) 50(4):563–578
9. Kung HT, Wu CH (2003) Differentiated admission for peer-to-peer systems: incentivizing peers to contribute their resources. In: Proc. of the workshop on economics of peer-to-peer systems, Berkeley, CA, USA
10. Andrade N, Brasileiro F, Cirne W, Mowbray M (2004) Discouraging free-riding in a peer-to-peer CPU-sharing grid. In: Proc. of the 13th IEEE international symposium on high-performance distributed computing
11. Kamvar SD, Schlosser M, Molina HG (2003) The EigenTrust algorithm for reputation management in P2P networks. In: Proc. of the 12th international conference on world wide web, Hungary, pp 640–651
12. Marti S, Garcia-Molina H (2004) Limited reputation sharing in P2P systems. In: Proc. of ACM conference on electronic commerce, pp 91–101
13. Xiong L, Liu L (2004) PeerTrust: supporting reputation-based trust for peer-to-peer electronic communities. IEEE Trans Knowl Data Eng 16(7):843–857
14. The Economist (2005) Who wants to buy a computon? 12 March
15. AuYoung A, Chun BN, Snoeren AC, Vahdat A (2004) Resource allocation in federated distributed computing infrastructures. In: Proc. of OASIS '04
16. Feldman M, Lai K, Zhang L (2005) A price-anticipating resource allocation mechanism for distributed shared clusters. In: Proc. of the 6th ACM conference on electronic commerce
17. Foster I, Kesselman C, Tuecke S (2001) The anatomy of the grid: enabling scalable virtual organizations. Int J Supercomput Appl 15(3):1–4
18. Pai V, Mohr AE (2006) Improving robustness of peer-to-peer streaming with incentives. In: Proc. of the first workshop on the economics of networked systems
19. Yan Y, El-Atawy A, Al-Shaer E (2007) Ranking-based optimal resource allocation in peer-to-peer networks. In: Proc. of IEEE INFOCOM
20. Cohen B (2003) Incentives build robustness in bittorrent. http://bittorrent.com/bittorrentecon.pdf
21. Eger K, Killat U (2006) Bandwidth trading in unstructured P2P content distribution networks. In: Proc. of the 6th IEEE international conference on peer-to-peer computing
22. Bharambe A, Herley C, Padmanbhan V (2005) Analyzing and improving BitTorrent performance. Technical Report MSR-TR-2005-03, Microsoft Research
23. Mowbray M et al (2006) A reciprocation-based economy for multiple services in peer-to-peer grids. In: Proc. of the IEEE international conference on peer-to-peer computing, pp 193–202
24. Banerjee D, Saha S, Sen S, Dasgupta P (2005) Reciprocal resource sharing in P2P environments. In: Proc. of the 4th international conference on autonomous agents and multi-agent systems (AAMAS'05), Utrecht, The Netherlands, pp 853–869
25. Papaioannou T, Stamoulis G (2005) An incentives' mechanism promoting truthful feedback in peer-to-peer systems. In: Proc. of the fifth IEEE International symposium on cluster computing and the grid, Cardiff, UK
26. Efstathiou EC, Frangoudis PA, Polyzos GC (2006) Stimulating participation in wireless community networks. In: Proc. of IEEE INFOCOM, Spain
27. Antoniadis P, Grand BL, Satsiou A, Tassiulas L, Aguiar R, Barraca J, Sargento S (2008) Community building over neighborhood wireless mesh networks. IEEE Technol Soc (Special issue on Potentials and Limits of Cooperation in Wireless Communications) 27:48–56

28. Mekouar L, Iraqi Y, Boutaba R (2009) A contribution-based service differentiation scheme for peer-to-peer systems. Peer-to-Peer Netw Appl 2:146–163
29. Basu S, Banerjee S, Sharma P, Lee SJ (2005) NodeWiz: peer-to-peer resource discovery for grids. In: Proc. fifth IEEE Int'l. symp. on cluster computing and the grid (CCGRID'05)
30. Cirne W, Brasileiro FV, Sauve J, Andrade N, Paranhos D, Santos-Neto EL, Medeiros R, Silva F (2003) Grid computing for bag-of-tasks applications. In: Proc. 3rd IFIP conference on E-Commerce, E-Business and E-Government
31. Guo L, Chen S, Xiao Z, Tan E, Ding X, Zhang X (2005) Measurements, analysis, and modeling of BitTorrent-like systems. In: Proc. 5th ACM SIGCOMM
32. Piatek M, Isdal T, Anderson T, Krishnamurthy A, Venkataramani A (2007) Do incentives build robustness in BitTorrent. In: Proc. 4th USENIX symp. networked systems design and implementation (NSDI), pp 1–14
33. Qiu D, Shrikant R (2004) Modeling and performance analysis of BitTorrent-like peer-to-peer networks. In: Proc. ACM SIGCOMM
34. Locher T, Moore P, Schmidt S, Wattenhofer R (2006) Free riding in BitTorrent is cheap. In: 5th workshop hot topics in networks (HotNets)

Anna Satsiou (S'07) received the Diploma in Electrical and Computer Engineering from the Aristotle University of Thessaloniki, Greece in 2002, the MSc in Electronic and Computer Engineering and the MSc in Production Engineering and Management from the Technical University of Crete (TUC), Greece in 2004 and 2005 respectively.

Currently, she is a Ph.D. candidate in the Department of Computer and Communication Engineering of the University of Thessaly and a Research Assistant in the Informatics and Telematics Institute of Thessaloniki.

Her research interests are in the general area of computer and communication networks and particular in peer-to-peer, wireless and mobile communications, security and quality of service (QoS) issues, and distributed multimedia content delivery networks.

For the period 2002 to 2004 she was awarded a Fellowship from Onassis public benefit foundation.

Leandros Tassiulas (S'89, M'91, SM/06 F07) obtained the Diploma in Electrical Engineering from the Aristotle University of Thessaloniki, Thessaloniki, Greece in 1987, and the M.S. and Ph.D. degrees in Electrical Engineering from the University of Maryland, College Park in 1989 and 1991 respectively.

He is Professor in the Dept. of Computer and Telecommunications Engineering, University of Thessaly, since 2002. He has held positions as Assistant Professor at Polytechnic University New York (1991–1995), Assistant and Associate Professor University of Maryland College Park (1995–2001) and Professor University of Ioannina Greece (1999–2001).

His research interests are in the field of computer and communication networks with emphasis on fundamental mathematical models, architectures and protocols of wireless systems, sensor networks, high-speed internet and satellite communications.

Dr. Tassiulas is a Fellow of IEEE. He received a National Science Foundation (NSF) Research Initiation Award in 1992, an NSF CAREER Award in 1995 an Office of Naval Research, Young Investigator Award in 1997 and a Bodosaki Foundation award in 1999. He also received the INFOCOM 1994 best paper award and the INFOCOM 2007 achievement award.