Dynamic Cooperation Enforcement through Trust-based Allocation Policies in P2P Systems

Anna Satsiou, Leandros Tassiulas
Dept. of Computer Engineering and Telecommunications
University of Thessaly
Volos
{satsioua, leandros}@ uth.gr

Abstract—In this paper we propose distributed trust-based resource allocation policies to provide fairness in p2p-like systems. According to these policies, the portion of the available resource received by each competing peer depends on (a) his trust/reputation, (b) his resource demands and (c) his request generation rate. The reputation of a peer is quantified and an approach for continuously updating the reputation is proposed such that it reflects the contributions of the peer in the transactions he engages. We study both homogeneous and heterogeneous systems of peers with different resource capabilities and needs and we see from theoretical analysis and simulation results that proposed policies motivate peers to contribute resources in the network by guarantying that peers will only receive resources in proportion to their contributions; thus misbehaving (non contributive) peers cannot exploit the system. Furthermore, proposed policies lead to the dynamic formation of coalitions (cooperation) between peers who mutually benefit by their transactions, according to their capabilities and needs. Peers’ coalitions are adaptive to network changes and self-organized as new peers enter the system or strategic peers vary their contributions.

Keywords—Cooperation enforcement, incentive mechanisms, p2p systems, trust-based resource allocation policies

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. Autonomic peers can simultaneously act as clients and servers taking part in a collaborative workflow 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” [1] 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 [2] in 2005 indicated that 85% of Gnutella users are free riders.

One way to alleviate the aforementioned problem is to give incentives to users to cooperate. Many methods were proposed in the literature in order to enforce users to cooperate, like pricing-based schemes [3], game theoretical methods [4]-[6], and trust and reputation based methods [7]-[10].

On one hand, the need of trusted virtual banks to 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, including private information of other peers like their bandwidth capacities, necessitated the use of simpler and more practical methods to foster users to cooperate.

Trust and Reputation systems have been extensively investigated in file sharing systems to distinguish and avoid malicious peers by applying suitable provider selection mechanisms [7]. Aforementioned mechanisms, though, do no 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] provide the appropriate 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 (non contributive) peers are punished by not being served. However most of the aforementioned work does not consider quality of service issues. Nodes are simply distinguished to altruistic and egotistic according to whether they provide services or not, while the different quality of service offered and received or the different capabilities and needs of the peers are not considered.

Aforementioned shortcomings motivated us to propose distributed trust/reputation-based resource allocation policies in systems of peers who may have different service capabilities, needs and request generation profiles. In this paper we consider that one single service is being exchanged, as bandwidth in file sharing systems. Our proposed policies determine the quality of service given to all competing for resources peers at a given time according to their trust/reputation demands and request generation profile. In a real implementation this could be realized by placing different amounts of data in the socket buffers of the TCP connections or by using more sophisticated traffic control to schedule the traffic directly at the queue of the network interface card [11].

We quantify a peer’s tendency to contribute by a novel reputation metric and an update mechanism is proposed such that the capabilities/contributions of the peers are tracked dynamically as the system evolves. The terms trust, reputation and contribution level are used interchangeably in this paper. Both theoretical analysis and simulation results have shown that proposed policies motivate peers to
contribute resources in the network. In fact, 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. A trust-based server selection policy is also proposed, which discourages peers from seeking service by misbehaving ones. The proposed policies lead to the dynamic formation of coalitions (cooperation) between peers of similar service needs and capabilities, in order to improve their utility from the network.

II. RELATED WORK

From our knowledge there is no previous work in proposing bandwidth allocation policies based on a reputation system. There are several papers, though, that propose bandwidth allocation policies in p2p systems based on an implicit contribution level of the users. In [12] 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 and different needs of the competing peers. In [4] and [6] a peer’s tendency to contribute is represented by the “contribution value”, while in [13] “ranking” is a metric of the behavior of a peer. Allocation decisions are taken 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 on the contrary, reputation is affected by the allocation decisions taken by the peers and dynamically adapts to network changes as the system evolves, representing a more realistic system.

Finally, there are reciprocation-based mechanisms proposed for BitTorrent-like networks according to which 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 [14]. One drawback of these schemes is that they do not motivate peers who already downloaded their files (seeds) to stay in the system and keep on contributing resources to other peers. A reputation-based allocation could prompt such peers to stay and contribute in the system with the profit of gaining reputation which they could use in their future transactions to improve their received quality of service.

Aforementioned work does not explore the possible 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; proposed policies can help peers to find the appropriate traders with whom they can mutually benefit. In [15] 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.

III. TRUST-BASED POLICIES

A. Model Description

We consider a p2p network of N peers who provide and consume a particular service. In this paper we consider that the service is bandwidth for content sharing. Each peer i can provide bandwidth with different capability, according to the upload capacity $C_i$ (measured in bits per second) of the access link with which he is connected to the Backbone Network. Capabilities of peers not need to be global information in our system, but considered private/hidden information, which is revealed through the reputation mechanism. We consider that the capacity of the backbone network is enough to accommodate all the traffic between the peers. This assumption is quite realistic as previous studies have shown that the utilization in the core of the network is low [16]. We assume that the physical upload capacity of each peer is the bottleneck in the network, while the download capacity $dC_i$ of each peer $i$ is sufficient to accommodate every incoming flow. This is a quite reasonable assumption as the download capacity is usually much higher than the upload capacity of the peers, (e.g. in ADSL connections). The network model is depicted in Fig.1. In the rest of this paper we will refer to the upload capacity of a peer $i$ as his bandwidth capacity or capability and we denote it by $C_i$.

We consider that our system progresses in periods of $T$ time units. At each period each peer $i$ generates $Q_i$ requests according to a uniform distribution with mean value $Q$. In each one of his requests he reports his demands $D_i$ (in terms of bandwidth) for the given request. The bandwidth requested by $i$ could be for any kind of content/files which are considered available by any other than $i$ peer in the system, but provided with different quality according to their capabilities and availability. $D_i$ is Zipf distributed between $[\min D_i, \max D_i]$ with skew $sk$, according to each peer’s demand profile. In each one of $Q_i$ requests a peer may report different demands. Peers act both as servers and clients simultaneously. A peer as a client may select the servers to which he will direct his requests either arbitrarily or based on some attributes that indicate the quality of service he may expect. At each period, peers collect the requests that have been directed to them and allocate their available upload capacity according to the proposed allocation policies.

![Fig. 1. The Network Model](image-url)
The service is not granted for more than a period of $T$ time units, in order to give the chance for other peers to access the link. At each period peers redirect their requests to the same or possibly other servers aiming at improving their received quality of service. The periodic readjustment of the peers’ strategies is also adopted in other file sharing systems like BitTorrent protocol [15], where peers during periods of typically 10 seconds evaluate the three peers that have most contributed to themselves and then favour them in the next period by serving their requests. However, unlike BitTorrent protocol where peers use a simple memoryless reciprocation mechanism, our trust-based allocation scheme aims at providing fairness in the system by guaranteeing that peers will be able to receive resources in proportion to their cumulative contributions in the system. Under this case, even seeds are motivated to remain in the system and contribute resources.

B. Reputation

If in a given transaction a peer $i$ demands a certain amount of service $D_i$ from a peer $s$ and finally receives $x_i \leq D_i$, the ratio $x_i/D_i$ represents how much server $s$ satisfied his needs and accounts for the reputation of peer $s$ in the eyes of peer $i$. This is actually the local reputation of $s$ to $i$. At each new transaction with the same server, the local reputation of the server to peer $p$ is recalculated. So, the reputation of peer $s$ to any peer $i$ at a given period $p$ is given by $R_{i,s}^p = \frac{1}{\text{Req}_{i,s}^p} \sum_{t \in [1,p]} x_i(t)/D_i(t)$, where $\text{Req}_{i,s}^p$ is the total number of bandwidth requests that peer $i$ has sent to peer $s$ till period $p$ and $t$ denotes a given transaction with peer $s$. In a similar way, each peer in the system calculates the local reputation of all other peers. It is obvious that the reputation metric can take any value from 0 to 1. We see that the proposed reputation metric expresses the average satisfaction, in terms of quality of service, offered by the given peer and not just the number of requests served, as in most of work for reputation.

We consider that new peers in the system are awarded an initial small reputation, in order to have the chance to receive some resources from the network. If they proved to be collaborative enough, their reputation will be increased, permitting them to receive more resources. We have seen from our simulations that a history of 10 past and most recent transactions is enough to evaluate one’s reputation. From our simulations that a history of 10 past and most recent transactions is enough to evaluate one’s reputation.

We further calculate the average satisfaction of each peer $i$ in the system by summing his satisfaction ratios in each one of his $t$ transactions with all other peers and averaging over the total number of his requests till period $p$, $\text{St}_{i}^p$, that is $\text{St}^p_i = \frac{1}{\text{Req}_{i}^p} \sum_{t \in [1,p]} x_i(t)/D_i(t)$.

D. Trust-based Allocation policy (TAP)

Suppose that a peer $s$ receives in a given period $p$, $n$ requests for bandwidth and that his upload capacity is $C_s$. We use a vector $L = \{L_1, \ldots, L_n\}$, each element of which represents a specific requester of peer $s$ with demands expressed by $D_{s,l}$, $l \in \{1, \ldots, n\}$. Then, we consider that $s$ will allocate $C_s$ in order to solve the maximization problem:

$$\max \sum_{l \in [1,n]} R_{s,l}^p \frac{1}{D_{s,l}} x_{s,l} \quad \text{s.t} \quad \sum_{l \in [1,n]} x_{s,l} \leq C_s \quad \text{and} \quad x_{s,l} \leq D_{s,l} \quad \forall l, \quad (1)$$

where $R_{s,l}^p$ represents the local reputation of peer $L_l$ to peer $s$ at period $p$, and $x_{s,l}$ is the allocated resource from peer $s$ 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 the competing peer does not receive any resource, as a punishment for his zero contributions.

Both the objective function and the constrains in the maximization problem (1) are linear, so the solution in a period $p$ can be easily found by sorting peers in decreasing order according to their ratio $R_{s,l}^p/D_{s,l}$. Then, the server will satisfy the needs of 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 peers, till all resources are exhausted or all peers are completely satisfied. TAP actually maximizes the satisfactions \( x_i / D_{t_i} \) of the competing peers, according to their reputations.

Applying TAP, peers with the higher ratio \( R^p_{s,t_i} / D_{t_i} \) 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.

It is evident from the solution of (1) that for any two competing peers \( i \) and \( j \) with \( (R^p_{i,t} / D_{t}) \leq (R^p_{j,t} / D_{t}) \) their satisfactions for the given period will satisfy \((x_i / D_{t}) \leq (x_j / D_{t})\). This means that peers with the highest contribution level per unit resource request will receive the highest satisfaction. This final point clearly shows that TAP provides incentives for cooperation. Actually, the higher demands a peer has, the more contributive/reputed he has to be in order to satisfy them.

E. Enhanced Trust-based Allocation Policy (ETAP)

Proposed allocation policy described in section III.D is fairly suitable in cases where all peers’ requests follow the same uniform distribution with the same average value per period \( \bar{D} \). However, in case some peers produce many more requests than others, they should be properly handled so as not to absorb most of the resources in the system, in cost of other collaborative users. The following enhanced allocation policy (ETAP) is proposed to cope with this case. Peer \( s \) will share his available resources to \( n \) competing peers, by solving:

\[
\begin{align*}
\max \sum_{l=1}^{n} \left( \frac{R^p_{s,l}}{D_{l}} \times \frac{\text{Req}^p_{s,l}}{\text{Req}^p_{s,l}} \right) x_{s}^{l},
\end{align*}
\]

\[
\text{s.t} \quad \sum_{l=1}^{n} x_{l} \leq C_{l}, \quad x_{l} \leq D_{l}, \quad \forall l,
\]

where \( \text{Req}^p_{s,l} \) is the total number of requests that peer \( s \) has sent to each peer \( l \) 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 collaborations with these peers, and on the other hand low rate peers will be able to restrict the excess requests of high rate peers, satisfying even other collaborative peers whom they need. From the solution of (2), which can be found by a similar procedure with the one for (1), we can easily see that high rate peers need to be more contributive than low rate peers in order to satisfy all of their requests.

F. Trust-based Server Selection Policy

Along with the proposed allocation policy, a proper trust-based server selection policy is needed in order to block misbehaving peers. So, each time a peer would like to send a request for bandwidth he should direct his request to one of his most reputed transacted peers, avoiding in this way misbehaving peers. However, when a peer enters the system for the first time he does not have information about the behavior of other peers. So, for a short duration of time (acquaintance duration) new peers direct their requests to all peers with equal probability (random selection policy), till they obtain a spherical view of the network. Similarly, preexisted peers in the system will use the random selection policy among their top reputed and new peers in the system for 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 \( j \) at a given period \( p \), is directly proportional to \( j \)’s local reputation to \( i \), as

\[
p^r_{i,j} = R^p_{i,j} / \sum_{s \in S_i} R^p_{s,i},
\]

where \( S_i \) is the set of all peer \( i \)’s transacting peers.

IV. PERFORMANCE EVALUATION

A. Simulation Model

We conducted an event-driven simulation study in C++ to evaluate the performance of our policies and investigate whether they satisfactorily motivate peers to cooperate. We simulate a peer-to-peer network of 100 peers who share bandwidth for file/content sharing and have different upload capacities and varying demands for bandwidth. Each peer produces 3 requests per period, unless otherwise stated. Peers acting as servers allocate their bandwidth to their competing peers based on TAP or ETAP and peers acting as clients select their servers, following the random or the trust-based server selection policy.

All peers start with an initial small reputation equal to 0.07 in order to bootstrap the system. Misbehaviour reputation threshold, below which a peer does not receive service, even when he is the only one competing for resources, is 0.01. Acquaintance duration under which new peers send their requests with equal probability to all others, in order to have a global picture of the network is set to 100 periods, while total simulation time is 1000 periods. We also run the system for different random initial reputation metrics for all peers and our simulation results have shown that reputations quickly converge to the same steady state, as with any initial reputation conditions and depend solely on the contributions of the peers. The performance of the system scales even to larger overlay networks; however in larger networks the convergence time of the reputations increases.
B. Comparison with other Allocation Policies

In this section we further describe some alternative allocation policies for the case of competing peers requesting service from server $s$, in order to highlight the benefits of our proposed scheme over them.

1) max($x$): Under this policy a server allocates his upload capacity by maximizing $\sum_{i=1}^{n} x_i$, under the constrains $\sum_{i=1}^{n} x_i \leq C_s$ and $x_i \leq D_i \forall i$. He uses a progressive filling algorithm by increasing 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.

2) max($x/D$): This policy maximizes $\sum_{i=1}^{n} (x_i / D_i)$, under the constrains $\sum_{i=1}^{n} x_i \leq C_s$ and $x_i \leq D_i \forall i$. The solution can be found in a similar way as the solution of (1). 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.

3) max($Rx$): This policy maximizes $\sum_{i=1}^{n} R_{i,s} x_i$, under the constrains: $\sum_{i=1}^{n} x_i \leq C_s$ and $x_i \leq D_i \forall i$. It gives more resources to more reputed peers. This policy will be denoted as max ($Rx$) in the figures of this paper.

V. SIMULATION RESULTS

A. Peers of Different Capabilities and Misbehaving Peers

In Fig. 2, we consider peers who wish to dedicate different bandwidth capacities for uploading either because their physical bandwidth capacities or their willingness to offer are different. 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. Peers’ requests are directed with equal probability to each one of the rest of the peers (random server selection policy).

From Fig. 2 we can clearly see how the reputations of the peers are analogous to their capacities (dedicated upload bandwidth), and moreover their satisfactions are analogous to their reputations. Higher capacity peers gain more reputation and satisfaction from the network than lower capacity ones. We further tried our policies for different capacities, demands, load and peer populations and the conclusions of the analysis remained the same.

In Fig. 3 the lower capacity (3 & 4 Mb/s) peers of Fig. 2 are now misbehaving, by not contributing any resources in the network (capacity=0) and constitute 40% of the total number of the peers in the system. We can see that, although selfish peers do not succeed in gaining anything from the system (their satisfaction ratio is almost zero), they do succeed in deteriorating the performance of others peers who, ignoring the misbehavior, direct some of their requests to the selfish peers. The performance of contributive peers can significantly be improved (45% improvement) if they use the trust-based server selection policy described in section III.F.

Fig. 3. Average Satisfactions of different capacity peers, when different server selection policies are used and 40% are selfish peers

Fig. 4. Average Satisfactions of different capacity peers, when different allocation policies are used and 40% are misbehaving peers
In Fig. 4 we see the satisfaction ratios of the different peers, when the various allocation policies described in section IV.B are used along with the trust-based server selection policy. It is clear that only our proposed allocation scheme, which we denote by max(Rx/D) in this figure, and the scheme max(Rx) achieve service differentiation analogous to the contributions of the peers and refrain misbehaving peers from exploiting the system. Our policy, though, achieves much better satisfaction ratios than max(Rx) for the same available bandwidth in the system.

It is worth noted here that the proposed allocation and server selection policies relieve system from other type of misbehavior, malicious attacks like Sybil attack and whitewashing, according to which a malicious peer can switch among different identities and exploit the system. With our policies, when a new identity appears in the system it is given a small initial reputation from other peers in order to prove its collaboration, as explained in section III.B. However, if this new peer does not provide enough resources to the system, he will not be able to obtain any, since its initial reputation which is very small to be competed with the reputation of already existed contributed peers will even further decrease. In this way, malicious peers can not exploit the network by creating one or more new identities and, furthermore, cannot deteriorate the performance of other peers, when proposed trust-based server selection policy is used. Our simulations show that newcomers really have to prove themselves as contributive peers in order to savor the acknowledgment (contributions) of the rest of the peers (see next section).

B. NewComers

In this section we examine the performance of the system when peers periodically leave and join the network. As in section V.A we consider a network of 100 peers where 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. However, every 200 periods 50% of each category (capacity) peers leave the system and are replaced by new identity peers of the same capacities. This also simulates the case under which misbehaving peers leave the network in purpose and appear with new identities in order to exploit the system. However, as we see from Fig. 6 neither stable misbehaving peers nor new 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, while the newcomers of 7 Mb/s capacities can achieve almost the performance of the steady state in the end of their lives in the system, indicating that they quickly self organize and find the appropriate partners with whom to trade. As we said new peers stay for 200 periods in the network. During the acquaintance duration (100 periods) they direct their requests with equal probability among all others and their performance in the end of this acquaintance duration is only 0.51 as we see from Fig. 6 due to the presence of the misbehaving peers. After the acquaintance duration new peers start directing their requests to the more reputed ones and thus in the end of their lives (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 be even more improved to reach the steady state. Peers could possibly speed their perception of the network by using recommendations from other peers. However, recommendations may induce a lot of problems as we already stated in section III.B and their investigation is out of the scope of this paper.
C. Formation of Coalitions

Next, we consider that 50% of peers have both 7 Mb/s capabilities and demands (strong peers) and 50% of peers have 3 Mb/s capabilities and demands (weak peers). In Fig. 7 we see that peers were self organized into two coalitions; the one of the strong peers and the other of the weak 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), and thus they prefer to send their requests to other weak peers (weak peers have higher local reputation to them). This formation of collaborating groups introduces fairness in the network since weak peers cannot receive more resources than stronger ones (which would be the case if they could wastefully use the resources of the strong peers).

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 into coalitions according to their needs and capabilities, by using the proposed distributed policies. We have detected these coalitions in all other cases of heterogeneous (capacity and demands) peers. In Fig. 7 we see that still some requests are directed to opposite groups; this is mainly due to the acquaintance duration under which peers send their requests randomly to one another.

D. Peers with Heterogeneous Request Generation Rates

In this section we investigate the case of peers with heterogeneous request generation rates. We consider a network of 100 peers, where they all have the same capabilities of 7 Mb/s and their demands are uniformly distributed between 1 and 7 Mb/s during simulation time. However, we consider that 50% of the peers are high rate (HR) peers generating 4 requests per period, while 50% of them are lower rate (LR) peers generating 2 requests per period. In Table I we can see the average bandwidth per request, the average bandwidth per period (T) and the average satisfaction of high rate and low rate peers, for TAP and ETAP with \(a = 1\) or \(a = 2\).

We note that with 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). However, this policy does not function very fair for LR peers and actually motivates 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, ETAP is used. 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 slightly more bandwidth per period than LR peers. ETAP with \(a = 2\) is even more restrictive to HR peers, leading to the same average bandwidth per period for both HR and LR peers.

In Table II we consider that (a) HR peers have a capacity of 7 Mb/s while LR peers have a capacity of 3 Mb/s and (b) that 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 is much restricted by ETAP policies. It is like punishing weak peers for their excess requests, since they are not able to provide the analogous 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 reach the same conclusion for any other cases of request rates, peer populations and capabilities that we examined.

When 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.

E. Strategic Peers

In this section we consider that peers are strategic, i.e. they seek to maximize their satisfaction with the least

---

**Table I**

<table>
<thead>
<tr>
<th>LR Cap.: 7Mb/s</th>
<th>AvBand./Req</th>
<th>AvBand./T</th>
<th>Av Satisf.</th>
</tr>
</thead>
<tbody>
<tr>
<td>HR Cap.: 7Mb/s</td>
<td>HR LR HR LR HR LR</td>
<td>2.19 1.94 8.76 3.88 0.69 0.64</td>
<td>ETAP (a=1) 1.70 2.90 6.80 5.80 0.57 0.82</td>
</tr>
<tr>
<td>HR Cap.: 7Mb/s</td>
<td>HR LR HR LR HR LR</td>
<td>1.56 3.19 6.24 6.38 0.51 0.87</td>
<td></td>
</tr>
</tbody>
</table>

**Table II**

<table>
<thead>
<tr>
<th>LR Cap.: 7Mb/s</th>
<th>AvBand./Req</th>
<th>AvBand./T</th>
<th>Av Satisf.</th>
</tr>
</thead>
<tbody>
<tr>
<td>HR Cap.: 7Mb/s</td>
<td>HR LR HR LR HR LR</td>
<td>1.80 1.10 7.20 2.20 0.60 0.43</td>
<td>ETAP (a=1) 1.53 1.61 6.12 3.26 0.54 0.56</td>
</tr>
<tr>
<td>HR Cap.: 7Mb/s</td>
<td>HR LR HR LR HR LR</td>
<td>1.31 2.05 5.25 4.11 0.48 0.64</td>
<td></td>
</tr>
<tr>
<td>LR Cap.: 3Mb/s</td>
<td>HR LR HR LR HR LR</td>
<td>1.43 1.83 5.71 3.66 0.51 0.61</td>
<td>ETAP (a=2) 1.08 2.52 4.33 5.04 0.39 0.74</td>
</tr>
<tr>
<td>LR Cap.: 3Mb/s</td>
<td>HR LR HR LR HR LR</td>
<td>1.04 2.60 4.17 5.20 0.36 0.76</td>
<td></td>
</tr>
</tbody>
</table>
possible contributions. As we already saw by using the proposed allocation policy the average satisfaction ratio of a given peer is proportional to his global reputation. Therefore, a peer knows that as soon as his satisfaction ratio reaches a certain high value, his global reputation is high enough and thus he can take advantage of it to start misbehaving.

Each time a strategic peer reaches a certain average satisfaction ratio (threshold) he progressively decreases his dedicated capacity for uploading (here we consider that the capacity is decreased unitarily by steps of 1 Mb/s). Once his average satisfaction ratio falls down this threshold, he progressively increases his capacity until it reaches its maximum value (determined by his physical upload capacity or the desired average satisfaction ratio) in this case he starts decreasing his contributions once again. Peers initially contribute their whole capacity and dynamically adjust their contributions according to their perceived satisfaction.

In Fig.8 we see the average satisfaction of altruistic and strategic peers for different percentage of strategic peers in the system. We refer to the peers who contribute their maximum upload capacity during their life 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%. It is obvious that altruistic peers deplete 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.8), their average satisfaction ratio would be 17.5% higher than if they were all strategic peers (case 100% in Fig.8), respectively.

It is remarkable to observe that the average satisfaction ratio of the strategic peers is under their satisfaction threshold (60% in this experiment) for all cases of percentage of strategic peers in the system, indicating that peers indeed cannot profit by strategically reducing their contributions. We further reached the same conclusion for other satisfaction thresholds and possible strategies (e.g. peers stop contributing when their average satisfaction is equal to or over their satisfaction threshold).

VI. CONCLUSION

The trust based resource allocation policies proposed in this paper motivate peers to contribute resources in the network and guarantee fairness, as peers can only receive resources in proportion to their contributions. In fact, the higher demands and the more requests a peer has, the more resources he will have to contribute in order to satisfy his needs; therefore misbehaving or strategic peers cannot take advantage of such a system. Peers benefit by using the proposed trust-based policies as they tend to favor/satisfy those peers, who have been more contributive to them, and with whom more future collaboration is expected. Consequently, peers of similar capabilities and needs are dynamically self grouped, in order to improve their perceived satisfaction from the network. The proposed allocation scheme is adaptive to network changes (e.g. peer arrivals and departures) and is implemented in a distributed manner at each peer independently of the others. The only information that is passed from one peer to another is the requested amount of resources.

REFERENCES