UNIVERSITY OF THESSALY

Cooperation and Fair Resource Allocation in Communication Networks

A Dissertation submitted in partial satisfaction
of the requirements for the degree of

Doctor of Philosophy
in
Computer and Communication Engineering
by
Anna Satsiou

August 2010

UNIVERSITY OF THESSALY

Cooperation and Fair Resource Allocation in Communication Networks

A Dissertation submitted in partial satisfaction
of the requirements for the degree of

Doctor of Philosophy
in
Computer and Communication Engineering
by
Anna Satsiou
August 2010

Dissertation Committee:

Prof. Leandros Tassiulas, Chairperson
Prof. Leonidas Georgiadis
Prof. Georgios Pagkalos
Prof. Aikaterini Houstis
Associate Prof. Spiros Lalis
Assistant Prof. Iordanis Koutsopoulos
Lecturer Dimitrios Katsaros

The Dissertation of **Anna Satsiou** is approved by:

Committee Chairperson

University of Thessaly

Dedicated to my family:
my father Yannis, my mother Maria, and my brothers Kostas and Vaggelis

Acknowledgments

I would like to express my gratitude to all those who made this dissertation possible, and first of all to my supervisor, Professor Leandros Tassiulas, who entrusted this dissertation to me and provided his moral support and wise advising during its completion.

Most grateful to my beloved family for their love and invaluable support, and my friends for the courage and support they were constantly giving me. Especially, I would like to mention Anastasios Chatzianastasiou and Irene Karapistoli.

I would also like to express my sincere thanks to the members of my Dissertation Committee for generously sharing their expertise and time in reviewing this thesis, as well as my colleagues in the University of Thessaly and the Institute of Informatics and Telematics who accompanied me with their own way in this long and difficult journey, making it fruitful, enjoyable and memorable.

Special thanks to Thanasis Korakis and Apostolis Apostolaras for their valuable help and support in the implementation issues over UTH testbed.

Most of all, I express my gratitude to God and His Holly Mother Whom presence and love guides all my life.

Finally, I would like to acknowledge the support of the 03ED918 research project within the framework of PENED, co-financed by E.U.-European Social Fund (75%) and the Greek Ministry of Development-General Secretariat for Research and Technology (GSRT) (25%).

Thank you all!!!

Anna Satsiou

Thessaloniki, August 2010

Abstract

ABSTRACT OF THE DISSERTATION

Cooperation and Fair Resource Allocation in Communication Networks
Anna Satsiou

Doctor of Philosophy, Graduate Program in Computer & Communication Engineering
University of Thessaly, August 2010
Prof. Leandros Tassiulas, Chairperson

Last few years have witnessed a dramatic growth of communication networks, with peer-to-peer (p2p) overlay communities becoming a new network paradigm for diverse applications and services. 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 phenomenon is known as "free riding" and leads to the degradation of the system performance. In this thesis we propose distributed reputation-based resource allocation and server selection policies to motivate cooperation among users of p2p networks and provide fairness by guarantying that more contributive peers can receive a better QoS from their community.

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, reputation-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 reputation-based allocation policies determine the quality of service given to all competing for resources peers at a given time according to their reputations, demands, and profiles. On the other hand, our reputation-based server selection policy is used to choose suitable providers of the requested service and avoid misbehaving (non contributive) ones. Our simulations in various different setups of peers with different resource capabilities, consumption and service evaluation profiles present how proposed policies outperform previous work both for single and multiple services systems and lead to the dynamic formation of coalitions (cooperation) between peers who mutually benefit by their transactions.

We further study systems of peers who use their capacity-limited access links both for their upstream and downstream connections (*single capacity limited-link systems*). Such peers have to share their available resources between their own and other peers' needs. In the selfish approach, each peer would like to exploit the full capacity of his access link only for his downloads. However, if all peers acted selfishly, the system would collapse. In order to motivate peers to cooperate under such systems, we combine our distributed reputation-based system with a novel capacity adaptation algorithm. In this way, each peer has to trade off the capacity he will dedicate for uploading in order to increase his reputation and therefore his revenue and the capacity he will dedicate for his downloads. All peers act rationally, trying to maximize their utility. Our proposed policies lead rational peers to cooperation while promoting fairness, as peers receive resources in proportion to their contributions. Our policies outperform existing work in this area in which the slowest link becomes the bottleneck of a heterogeneous system of different link capacity peers. In the contrary, no such bottleneck appears when our policies are used, improving the performance of the system. Moreover, we apply our reputation-based framework in a BitTorrent-like file sharing system and we highlight the performance gains both for ADSL-type and single capacity-limited link systems.

Finally, we study a wireless internet sharing community and we propose easily applicable distributed reputation-based policies to motivate users to share their Internet connection in a p2p fashion. The goal is the provision of free and good quality Internet access anytime and anywhere inside the community. We study general systems of users with different contribution and consumption profiles and show that when our reputation-based allocation scheme is applied, mobile users enjoy a QoS Internet connection from their community in proportion to their cooperation level; the more greedy for community resources users are, the more contributive they should be to satisfy their needs. Our allocation scheme is further implemented in 802.11 APs and clients with no modifications at existing standards at the MAC or network layers and experimental results present that the throughputs of the community members are in proportion to their reputations for any channel conditions, type of traffic sent or received, and number of competing clients.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Cooperation, as a critical issue, in P2P Communication Networks

A dramatic increase in the usage of the internet has been witnessed in the recent years, with peer-to-peer (p2p) networks becoming a new prevalent architecture for diverse applications and services, including content storage and sharing (file-sharing, content distribution, backup storage), resources' exchange (grids) and communication (voice, instant messages), just as a few examples.

The p2p network paradigm is essentially about distributively performing processing, bandwidth, storage or human-resource intensive tasks between end-users who simultaneously act as clients and servers, without the necessity of an expensive dedicated infrastructure of central service providers.

The effectiveness of existing p2p systems, however, relies on the cooperation of the users and the contributions of their resources. Several studies, though, have shown that most of the users of p2p systems are incited to consume resources without contributing any. This is a social phenomenon reported as "tragedy of the commons" [3] that most of the users are reluctant to cooperate, and only a small number of them are willing to share their resources. The same phenomenon is known as "free riding" in the context of file sharing systems, like BitTorrent, Gnutella, Napster and Kazaa. In Gnutella, for example, it was reported in 2000 [4] that 70% of all users do not share files and nearly 50% of all file requests are satisfied by the top 1% of sharing nodes. In 2005, a new report [5] 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 [6], game theoretical methods [7–9], and trust and reputation based methods [1, 10–12].

Credit-based approaches treat file sharing and other resources offering as a service that can be priced with a virtual or real currency, which is used to regulate the various transactions between the users. However, virtual banks (trust authorities) are needed to deal with the transactions, which in case of non-fixed infrastructures, like peer-to-peer networks is not applicable. If however, the role of the virtual bank is distributed among nodes [6], significant communication and infrastructure overhead is induced and authentication issues become critical. Moreover micro-payment schemes may deter users from participating, as highlighted in [13].

Game theoretical methods, on the other hand, increase the complexity of the networks and usually games need to be played repeatedly between users in order to have the desired effects. However, the churn of nodes causes a lot of difficulties in the repetition of such games between the same users. Moreover, game methods presuppose that all nodes in the system are fully aware of system information. In [7], for example, it is considered that all peers know the bandwidth capacity and contributions of each other peer in the network. In [14] peers determine whether they will join a peer-to-peer network if their estimated benefit exceeds a threshold. However, that benefit is determined by a presumed preexisting knowledge of the significance to them of all the other peer's contributions.

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 the other peers like their bandwidth capacities, necessitated the use of simpler and more practical methods to foster users to cooperate, like reputation-based systems.

1.2 Reputation as an Efficient Means to Foster Cooperation

Trust and Reputation systems have been extensively investigated in p2p systems to distinguish and avoid malicious peers by applying suitable provider selection mechanisms [15–17]. These mechanisms, though, do not provide incentives for cooperation, since malicious peers have no repercussions and they can still savor the available resources of the network, while non-malicious peers are not motivated to improve their contributions.

Acknowledging such concerns, recent work in reputation systems [1, 10–12] provides simple incentive mechanisms to enforce collaboration between peers by controlling not only the provider selection policy but also the client selection policy. Reputed, thus contributive, peers are rewarded by receiving preferential treatment, while misbehaving 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.

More specifically, schemes [1,10–12] explicitly define a reputation metric and take binary allocation decisions (not serve or serve a peer by offering him all the available resources) and therefore are limited to categorize cooperative and non cooperative peers, while being incapable of differentiating cooperative peers in terms of the different QoS they offer. In this way, they fail to motivate cooperative peers to improve their provided QoS. Moreover, the peers' particular demands are not considered in the allocation decision, while it is assumed that the capabilities of the servers are always enough to satisfy the most reputed peer's demands.

Aforementioned shortcomings motivated us to propose a reputation-based allocation policy to decide about the different levels of offered resources to different users, in contrast to just decide whether to serve one peer or not, and to account for their different needs (expressed service demands). In section 3.3.2 we define our reputation-based resource allocation policy, RA, which determines the portion of the available resources offered to each competing peer depending on his reputation and his resource demands, while in section 3.3.3 we propose an enhanced policy, called ERA, which further considers the request generation profiles of the peers in the allocation decision. The benefits of our allocation policies become more obvious by the comparison results we provide of RA and ERA with scheme in [1] in Chapter 3.

From our knowledge, there is no previous work to determine 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, this mechanism does not apply for the individual needs of the competing peers.

In [7] and [9] 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, we propose and explicitly define a reputation metric which is dynamically updated and affected by the allocation decisions taken by the peers as the system evolves, corresponding to a more realistic system.

Furthermore, our ERA policy accounts for the different heterogeneous request generation profiles, in contrast to aforementioned policies, so that it can differentiate peers not only based on their contributions levels but also on their consumption profiles; greedy users can no longer exploit the network by absorbing much more resources than other similar contributive ones.

In this thesis, we propose a novel reputation metric and an update mechanism in order to dynamically capture the contributions of a peer in the transactions he engages. Reputation in previous schemes is usually defined as a function of the number of times a given peer helped another one, where "help" may correspond to offered files, CPU cycles, etc. For example, in [1] reputation is extracted by the ratings (binary values) about the outcome of a given transaction. In [12], reputation is defined as a function of satisfied uploads and downloads. In [2], the reputation of a given peer for a certain type of service is a function of the number of times he offered this service.

Our reputation metric is novel because it reflects the satisfaction in terms of QoS offered by the given peer, in contrast to just a function of the number of served requests. This is an important difference since a reputed user in our scheme is one that not only provides his help (e.g., offer a file or not) but also provides a high QoS in order to satisfy his requesters' demands (e.g., upload a file with high speed). In this way we can distinguish peers in terms of different capabilities/contribution levels in contrast to the binary categorization to altruistic (contributors/ reciprocative) and egotistic (free riders/selfish) peers. Following our approach and distinguishing users in more layers (in terms of different QoS offered), we can motivate them to improve their contributions and thus their reputation and perceived QoS.

There is also work [2, 11] that defines the reputation of a peer as a function of the total amount of provided and received resources. However, corresponding metrics fail to capture the tendency of a peer to satisfy the particular needs of his requesters. 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 at different times of the day, e.g., during congested periods a medium QoS can be considered more satisfying than in less congested ones. Most of the times, it is really difficult to determine the different levels of satisfaction of various peers, because of their different criteria for judging a service. However, by considering the peers' reported demands in our allocation decisions, we can account for their different degrees of satisfaction.

Summarizing, on one hand we propose a novel reputation metric capturing the ability of one peer to satisfy the demands of other peers, a reputation update mechanism to track the behavior of a peer dynamically, reputation-based resource allocation policies to determine the portion of the available resource offered to each competing peer and on the other hand, we propose a reputation-based server selection policy to help peers select among the more reputed/contributive servers and avoid misbehaving (non contributive) ones. The combination of our policies leads to bonds between peers of similar capabilities and needs. 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. Although simple enough, our policies were designed to consider many important issues, like peers of different capability and consumption profiles, newcomers, strategic and misbehaving peers. As far as we know, no previous work has attempted to encompass all the above under a common design framework. Some existing policies do not account for peers of different capability profiles ([1, 2, 8, 10, 12, 20]), some do not account for peers of different request generation profiles ([1, 7, 9]) and some do not account for peers of different demands ([1, 2, 8, 10–12, 20]).

1.3 Local versus Global Reputation Systems

Reputation-based incentives schemes can be categorized in two groups, one that use global reputations based on voting/ratings of others peers and one that use local reputations based only on the opinion of the peer that has been involved in a transaction. Work in [1, 8, 10] is based on the existence of a global reputation system, which is responsible (i) to collect the ratings of the peers for their transactions, (ii) to calculate the reputations of all the peers based on these ratings and (iii) inform the others about these reputations. In all these papers it is considered that peers send reports about their transactions to some specified peers that are responsible to calculate the reputation of a particular peer according to a

DHT mechanism. However, it is not guaranteed that peers will say the truth about their transactions. Analogous questions about the truthful feedback of nodes arise also in ad hoc networks, as reported in [21, 22].

Although recommendations and votes to determine one's reputation can speed up peers' perception of the network, there should be mechanisms to guarantee the trustworthiness/sincerity of the peers who vote in order for the reputation-based system to be efficient. Work in [23] proposes a mechanism to motivate peers to truthfully report their exchanges. According to [23], both transacting peers have to report the result of their transaction to the respective peers that are responsible for calculating and maintaining their reputation values. If their reports contradict, they will be both punished and the rating's feedback will be disregarded. Moreover, if only one of the peers sends his rating, both peers will be punished. Peers under punishment do not transact with others during their punishment period, while their ratings for such transactions are not taken into account. The problem with proposed mechanism is that unaware peers may be punished because another peer reported a fake transaction with them (that never really happened), which is very unfair. What's more, peers have always to ask which ones are punished in order not to ask service from them, increasing the overhead.

In general, it is very difficult in voting and rating sharing schemes not only to guarantee the trustworthiness of the peers who vote and share their ratings but also the trustworthiness of the peers who are responsible to collect the ratings and report peers' reputations. Such peers may lie for their own benefit, since they are part of the system. Furthermore, if these responsible peers exit the network, they must transfer their information to another neighboring peer and inform the network about this change. The whole procedure adds a lot of complexity to the system.

On the other hand, use of local reputations can significantly relieve the system from network load and complexity. Local reputation / information systems have been used in many practical systems as peer-to-peer streaming systems [18, 24] or p2p grid networks [11], where they have been proved to be efficient for resource allocation decisions.

In our work we use local reputations to, first of all, avoid the load from communication exchanges due to recommendations, as well as other possible problems like misreporting, reputation pollution and collusion which appear in reputation rating and voting systems like the ones used in [1,10,15,16].

Furthermore, by 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 (e.g., BitTorrent), for example, high capacity peers are better off by trading with other high capacity peers, while 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) [25]. Therefore, a low capacity peer who is considered as “low reputed” by high capacity peers, may be a good trader and considered as “high reputed” for some other low capacity peers. In this thesis, we exhibit how peers under our local reputation-based framework form coalitions according to their capabilities and demands, improving their performance.

Based on this local reputation-based framework and principles that we initially designed to foster cooperation in single service systems, i.e., systems where only one resource is being exchanged (e.g., bandwidth in file sharing systems), we propose extensions and modifications for various different setups highlighted in the following sections.

1.4 Reputation-based Exchange in Multiple Services Environments

Multiple Services systems, like p2p grids, where more than one services/resources are being exchanged between end-users, pose different challenges to the establishment of a cooperation scheme than the ones faced in single service systems. First of all, it is necessary to form a common exchange framework to determine the transactions between peers with different capabilities and need for different resources, like memory, CPU, storage capacity, bandwidth, etc. Some peers may have storage capabilities and CPU needs and some other peers may be processing powerful and have storage needs. An efficient framework should lead these these groups of peers to cooperate in order to improve their performance.

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 [26] 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 [27,28] have been proposed for distributed computing infrastructures, like Planetlab [29] or computational grids [30] 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 order to provide an efficient framework for transactions between different services and at the same time foster cooperation among users, we propose to use the notion of reputation as the “exchange currency” between different services. We define the local reputation vector of a peer which consists of his local reputations in providing each service of the system, while we extend our reputation based allocation policy RA to consider the reputation vectors of the competing peers and the service evaluation profile of the server, forming the RA-MS policy. Under RA-MS, 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. In this way the server tends to favor collaborative peers who best satisfy his requests in the services he values more, creating bonds with them. Our allocation policy along with our adapted to multiple services environments server selection policy lead to the formation of coalitions between users who mutually benefit by transacting and exchanging their resources, as can be seen from our simulation studies.

Apart from the market-based model, the only studies, as far as we know, that deal with the exchange of multiple services are [2] and [20] which base on reciprocity mechanisms and are described in detail in section 2.5. However, both schemes face the same limitations. The individual peers' demands and capabilities are not considered in the allocation decisions, rather it is considered that the capabilities of the peers are always enough to satisfy the demands of their requesters. Moreover, they do not use a reputation-based server selection policy in order to avoid requesting service from misbehaving peers and last, they do not investigate the possible formation of peers' coalitions.

Our scheme is evaluated, via simulations, under various setups of peers with different contribution, consumption and service evaluation profiles, newcomers and misbehaving peers, and present how it outperforms previous work [2] over multiple services environments to boost cooperation among peers and improve their performance.

1.5 Cooperation in Single Capacity-Limited Link Systems

As *single capacity-limited link systems (SC)* we define systems where the access technology does not separate upstream and downstream flows, like in wireless or non-wireless connections to the backbone network through a local hub with an Ethernet-like protocol. Examples of such access technologies include WiFi, WiMAX, Ehternet LANs, etc.

As we have already pointed out, many methods have been used in the literature to motivate users of a file sharing system to cooperate. Aforementioned work, though, implies that each peer in the system has separate capacities for his own needs (e.g., download capacity) and the needs of the other peers (e.g., upload capacity), as in the case of ADSL connections to the backbone network. However, in systems where the access technology does not separate upstream and downstream flows, each peer acting simultaneously as a client and as a server has to share his capacity-limited link that connects him to the network between his own and other peers' needs.

These systems differ from those examined in [1, 10, 15, 16, 18–20, 25, 31], in that the contributions (uploads) of each peer have a direct negative impact on his download performance. Therefore, each peer would rather exploit his full capacity only for his own downloads, in order to maximize his utility. It is apparent that appropriate incentives are needed to motivate peers to cooperate and contribute their resources in such p2p systems, since if each peer acted selfishly the system would collapse.

The incentives used in previous work [1,10,15,16,18–20,25,31] cannot be effective or even apply to single link capacity peers, as we explain in the following. Schemes in [1, 8, 10, 20] do not reckon quality of service issues; the different quality of service offered and received or the different upload capacities and needs of the peers are not considered in the allocation decisions. These schemes take as granted that the capacities of the peers are enough to satisfy a given peer (the most reputed) at a given period. Therefore, the incentives used in this work cannot apply to SC peers who have to decide how to allocate their limited capacity among incoming and outgoing flows according to their own and other peers' demands.

On the other hand, schemes in [7,19,32,33] propose bandwidth allocation policies based on peers' contribution levels. These policies regulate the allocation solely of the upload capacity of the peers, since it is considered that the peers' download capacity is infinite or at least enough to accommodate their demands. If these policies, though, were to apply for SC peers, the latter should somehow incorporate their own demands in their bandwidth allocation decisions which should be bounded by their total link capacity. Clearly, there is no obvious way how to do this, maintaining the incentives for cooperation.

The most relevant work, to our knowledge, that studies incentive mechanisms for systems of peers who share their single access link both for uploading and downloading files is [34], which, though, is imposed to several restrictions, discussed in section 2.3.

In order to motivate peers to cooperate under such systems, we propose a novel capacity adaptation algorithm and rational strategies to work along with our reputation-based framework. The capacity adaptation algorithm is performed by each peer and works rationally seeking to maximize the peer's benefit by trading off the capacity that the peer will dedicate for uploading in order to increase his reputation and therefore his revenue and the capacity he will dedicate for his downloads.

Our simulation studies have shown that our proposed policies lead rational peers to cooperation while promoting fairness, as peers receive resources in proportion to their contributions, while outperforming existing work in this area [34] in which the slowest link becomes the bottleneck of a heterogeneous system of different link capacity peers. In the contrary, no such bottleneck appears when our policies are used, improving the performance of the system.

1.6 Reputation-based Cooperation in BitTorrent

BitTorrent (BT) is a popular peer-to-peer content distribution protocol, designed for bulk data transfer. It enables scalable and efficient content replication by exploiting the upload capacity of the downloading peers. The basic idea in BitTorrent is to divide a single large file into equal-sized pieces and deliver them in a non-linear way. The set of peers attempting to download the file do so by connecting to several other peers simultaneously and download in parallel different pieces of the file from multiple peers.

In order to confront free-riding effect, BitTorrent uses a tit-for-tat (TFT) strategy to determine which peers to upload data to. However, several studies [25, 31, 35, 36] exhibited that BT's reciprocation policy (i) fails to prevent unfairness across peers in terms of volume of content served, (ii) is vulnerable to strategic clients and (iii) fails to shield the system from misbehavior. That is why there were several studies that proposed improvements over it [31, 35, 37–39].

However, proposed incentive mechanisms for BitTorrent-like networks do not motivate peers who already downloaded their files (seeds) to stay in the system and keep on contributing resources to other peers. Our reputation-based allocation scheme can prompt such peers to remain and contribute in the system, as we will see in Chapter 6, with the profit of improving their performance in their future transactions by gaining reputation. The presence of seeds in the system have proved to be a crucial factor in the success of BT [40] and so far no previous work has successfully provided incentives for seeds. Furthermore, the gains of our reputation system does not only affect the long term interactions among peers but also the short term. Peers select whom and how to serve based on competing peers' reputations, instead of performing optimistic unchokes, which are considered to be one of the main reasons of unfairness in BT [36, 41]. That is why our framework outperforms BT reciprocation mechanism in terms of fairness, as the simulation study of Chapter 6 indicates.

Finally, neither BitTorrent nor the proposed improvements over it [24, 31, 35, 37–39, 42], support an algorithm to dynamically determine how peers over single capacity-limited link systems should allocate their capacity among their uplink and downlink streams. Therefore BT peers' performance is critically based on their arbitrary decisions as we will see in section 6.3.3. In order to improve BT performance under single capacity-limited link systems, we propose two extensions over it. The one that cannot be hacked by a software uses our capacity adaptation algorithm combined with the reciprocity-based incentive mechanism of BitTorrent. As we will see in section 6.3.3, our capacity adaptation algorithm improves the performance of BT under such systems, especially when combined with our reputation-based framework, instead of the reciprocation mechanism of BitTorrent.

1.7 Reputation-based Internet Sharing in Wireless Neighborhood Community Networks

The vision of fourth generation (4G) networks is ubiquitous wireless access, anytime, anywhere. Towards this vision Wireless Mesh Networks (WMNs) have emerged to confront the various shortcomings of cellular networks (GSM/GPRS), providing a cheaper solution for higher data rates and with no special spectrum licences. In recent years a growing number of WiFi hotspots has been established in airports, hotels and cafes to offer Internet connectivity. Mesh networks provide the means to inexpensively and simply link all these hotspots to cover entire municipalities. Several municipalities, like Philadelphia and San Francisco, already provide metropolitan wireless access through hotspots to connect their citizens and offer public services.

Apart these commercial efforts, though, which are centrally managed and may charge for the service, there are also several initiatives building wide area wireless networks which are implemented and managed by volunteers. The participation of end users in the network formation is highly beneficial as it can reduce the costs and increase the network coverage. Such networks are known as Wireless Community Networks (WCNs) and include several cities (e.g. NYCWireless, SeattleWireless, Southampton Open Wireless Network, Athens Wireless, etc), as well as universities (e.g. MIT's Roofnet, etc).

The most promising service of all these projects is the potential provision of Internet access to members of the community carrying Wi-Fi enabled devices. This could be realized, if community users were willing to share the Internet access bandwidth of their Wi-Fi access points (APs) with other community members. Nevertheless, it is crucial to motivate users to cooperate and share their Internet connection, as there are intrinsic factors to discourage them from participating. Firstly, the sharing of the Internet connection with other users would reduce the owner's perceived quality of service (QoS) and secondly, the negative effect of exposure to radiation and possible security attacks would lead many users to close their AP as soon as they stop using it. Therefore, it is crucial to provide users with the appropriate incentives to cooperate and share their Internet connection, in order to maintain an Internet sharing community.

Interestingly, there are several commercial companies which make profit of the Internet sharing scenario. FON [43] is an example of a one-hop Internet sharing community. FON users just have to buy and activate in their home a specialized Wi-Fi router to share the broadband Internet service and in return gain free WiFi access at FON Spots worldwide; however, there are no mechanisms to solve the asymmetry issues in potential contribution and consumption and there are no QoS guarantees. Broadband provider Speakeasy introduced WiFi NetShare [44] to allow Speakeasy subscribers to easily (and legally) extend their broadband connectivity to their neighbors via WiFi, while reducing their monthly broadband bill. Finally, Skyrove [45] users can plug a Skyrove router into their network and offer Internet service to anyone who buy credits online; in return, they receive monthly payments based on the usage of their hotspot.

In contrast to aforementioned commercial companies which use a centralized payment or reciprocity based approach to motivate users to contribute their Internet connection, in this thesis we propose simple distributed reputation-based incentive mechanisms to foster cooperation among users of an internet sharing community. The objective of such a community is to both provide free and good quality Internet access to its users anywhere inside it, and protect their home connection resources by letting them the allocation control. Our policies guarantee that the better QoS internet connection users offer to their community members, the better QoS connection they enjoy, when they are mobile inside their community. This work focuses on neighborhood wireless communities which have been recently argued [52] to provide a strong candidate solution for creating and efficiently operating the underlying network and provides an ideal framework for the design of our policies.

Thereafter, the goal is to implement our internet sharing policy in IEEE 801.11 APs and clients, with no modifications at existing standards at the MAC or network layer. However, the unreliable nature of the wireless medium constitutes a difficult task to provide QoS in wireless networks. 802.11e standard can coordinate the channel access within a based station by implementing four queues with different access characteristics for each class of following services: BK (Background), BE (Best Effort), VI (Video) and VO (Voice). However, the QoS provisioning is done based only on the traffic characteristics of the existing streams and cannot differentiate different stations. Work in [47] resolves this issue by using a second prioritization level on top of the one that is implemented in IEEE 802.11e, defining priorities for different groups of associated stations; however, it is designed to differentiate only two groups of users. Both 802.11e and [47] control the AIFS (inter frame period to consider the medium idle) and the CWmin (minimum contention window size), but not the specific allocated bandwidth to each station according to his priority.

In order to control the specific amount of bandwidth offered to each competing user according to his reputation, based on current 802.11 protocol, we use Hierarchical Token Bucket (HTB) tool [48]. We, therefore, implement our reputation-based allocation policy based on HTB, which we call R-HTB. HTB is is a class based queue discipline, located in the IP layer, to control how the packets are delivered to MAC layer. In R-HTB the algorithm that determines the order and times that the packets are queued and dequeued is determined by our reputation-based allocation policy. In section 7.4 we provide indicative experimental results both in UTH testbed and individual ORBIT nodes, which show that the throughputs of the competing users are proportional to their reputations for any channel conditions, type of traffic and number of clients.

1.8 Outline of this Dissertation

The rest of the dissertation is organized as follows:

In Chapter 2 we describe previous cooperation enforcement and reputation-based schemes for various different setups and case studies that we examine in this thesis.

In Chapter 3 we present our reputation-based polices for systems of peers who share one single service, like bandwidth in file sharing systems, and provide comparison results with related work. Part of the work described in Chapter 3 is also published in [32, 49].

In Chapter 4 we extend our reputation-based framework for systems of peers who exchange multiple services (e.g., p2p grids), and compare our scheme with previous work over such systems. Part of this work is also published in [49, 50].

In Chapter 5 we propose extensions over our framework for systems of peers who use their capacity-limited access link both for their upstream and downstream connections. We will refer to these systems as *single capacity-limited link systems*. Part of this work is also published in [51].

In Chapter 6 we examine the popular file sharing application BitTorrent, as a case study for our proposed reputation based framework. We further present the performance gains both for systems of peers with ADSL-type connections and for single capacity-limited link systems, by comparing with the current BitTorrent incentive mechanism. Part of this work is also published in [51].

In Chapter 7, following the principles of our reputation-based framework, we propose an internet sharing scheme for wireless neighborhood communities, evaluated via simulations. We further implement proposed scheme and present experimental results both in UTH testbed and individual ORBIT nodes. Part of this work is also published in [52, 53].

In each of chapters 3-7, we (i) highlight the contributions and outline of the chapter, (ii) introduce the different problems posed by the individual system that is examined in the chapter, (ii) present our policies and their performance gains over previous work and last, (iv) conclude the chapter.

Finally, we conclude the dissertation in Chapter 8.

Chapter 2

Related Work

In this section we describe previous cooperation enforcement and reputation-based schemes for various different setups and case studies that we examine in this thesis.

2.1 Single Service Systems

As we already highlighted in the introduction, we can distinguish previously related work in two main categories. In the first category there are those reputation-based schemes that explicitly define a reputation metric and take binary allocation decisions (serve or not to serve a peer), ignoring QoS issues [1, 10–12], and in the second category the are those policies which differentiate peers according to their contribution levels [7, 9, 18, 19]; however, the latter do not determine how the "contribution level" will be measured and kept, rather "contribution levels" are used as abstract ideas and given random values in order to be evaluated. In the following, we delve into more details in the most representative schemes.

Scheme in [1] 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. We note that the 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 k that request the same service from a particular server j, the probability for a specific requester i to be selected equals $R_i / \sum_k R_k$, where R_i is the global reputation of peer i. Authors examine three provider/server selection policies. In the first policy, peers select the one with the highest reputation, in the second policy, peers select one whose performance level is comparable to theirs, and in the third policy peers select one that is not in their Black list which consists of peers with reputation value below a certain threshold.

Their simulation study considers peers that are divided in two types: altruistic and egotistic. Altruistic peers provide services successfully with probability 0.9, accounting for other unpredictable factors, like network congestion, while egotistic peers with probability 0.1. These probabilities remain constant during simulation time, accounting for fixed strategies. Authors also consider milking strategies, according to which peers with high reputation decide to misbehave in the end of their life in the system and dynamic strategies, according to which a peer increases or decreases his probability of providing a service according to the utility that he receives from the system during a predefined service period. The results of their simulations in Java conclude that the higher performing peers have the highest success ratio even when the peer population is renewed with a high rate.

Scheme in [10] considers a content sharing system. Each node monitors his success rate, which is the percentage of his content and transport of content requests that has been satisfied by other nodes. If a given node's success rate is above a desired level for a certain period of time, then the node will decrease his willingness to serve probability. When a node A receives a request from node B for content, initially he will consider this request based on his current willingness to serve probability. If this probability is high enough, node A will determine whether he will serve node B or not. Above decision is taken based on comparison with two benchmark nodes whose reputations are determined by a predefined set of nodes. Periodically, the set of nodes and two benchmark nodes are updated.

One of the two benchmark nodes has a high usage reputation (a lot of his content requests have been satisfied by other nodes) and the other one has a high service reputation (has served a lot of other nodes). If B has a higher usage reputation than the first benchmark node and smaller service reputation than the second benchmark node, he is denied service.

Authors of [10] evaluate the performance of their scheme by simulating a p2p system of 128 nodes, each constantly making content requests to other nodes. They simulate 10000 transactions per round and C value changes every 100 rounds. These simulations have shown that peers adapt their willingness to serve probability in order to find a minimum level of services they need to provide, while still being able to receive services provided by others. The amount of resources consumed and provided is not considered in their decisions. Moreover, no multiple requests for the same service are considered at a given time slot. Both models of [10] and [1] do not consider quality of service issues; a node either serves his requester by providing him the file, or rejects his request.

In [9] a distributed resource allocation protocol is proposed, which is called RBM-IU, based on which the download bandwidth offered by a server to his requesters is determined by the latter's bandwidth biddings and contribution levels. The theoretical analysis of this algorithm proves that the proposed allocation scheme is Pareto optimal and provides incentives for cooperation, as the peers who have the highest contribution per unit resource request among all other competing peers will receive the highest utility. This work does not consider a specific metric to define the contribution level, but it rather considers an abstract idea of it. It is further considered that the information about the contribution levels of all peers can be found by a centralized auditing authority, which is aware of all realized transactions in the system.

Paper [19] considers a resource allocation and admission control mechanism for p2p networks. It is assumed that each user is characterized by his ranking, which reflects his contribution level. The higher his ranking is, the worse his contribution level. However, neither "ranking" nor its adaptation to peers' dynamic behavior are explicitly defined, rather "ranking" is used as an abstract idea and given random values in order for the scheme to be numerically evaluated. It is taken as granted that users are aware of all other users' rankings in the system (e.g., by a trusted third party). The model consists of a single server and multiple competing peers. Each competing peer expresses both his demands for good quality service (target resources) and his minimum needs in bandwidth to sustain the service (minimum acceptable quality). The server will initially examine the requests of the competing peers and see whether his capacity is enough to accommodate their target resources or at least their minimum resource requirements. If, however, his capacity is not enough not even for their minimum requirements, he denies service to peers with rankings above a certain threshold, which is a function of all competing peers' rankings. Thereafter, the server allocates his available capacity among remaining competing users, following an allocation policy that is proved to achieve max min fairness. Some numerical examples are presented for the case of one server accommodating four competing peers with known rankings. The allocation decisions are exhibited showing that higher ranking peers achieve worse service than the other ones.

2.2 Multiple Services Systems

In order to regulate the transactions in multiple services environments, several market-based resource allocation systems [**?**, 27, 28] have been proposed for distributed computing infrastructures. One such representative scheme has been proposed by the authors in [?]. Proposed model consists of m users and n machines. Each machine (its computational capacity) can be continuously divided for allocation to multiple users. Each user places a bid to each machine, the price of which is determined by the total bids placed. Finally each peer receives the ratio of his bid to the sum of bids for that resource. It is considered that each user has different and relatively independent preferences for different machines. Users are selfish and seek to maximize their own utility by solving an optimization problem. Analytical results exhibit that the scheme converges to a Nash equilibrium under which efficiency and fairness is provided to the system. Fairness is evaluated through the measures of utility uniformity and envy-freeness.

One serious disadvantage of proposed model is that users need to send and receive $O(n)$ messages every time because they must be aware of the total bids on each host in order to solve the optimization problem. Furthermore, proposed scheme does not give incentives for cooperation, as users' own contributions (machines) are not considered in the allocation scheme. Actually, users are independent from machines in this work. It would be very interesting to consider users owning machines and schemes to prioritize service to those, who contribute the more machines in the system.

In general, market models 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. More practical systems [2, 20] rely on reciprocity-based mechanisms to regulate the transactions and encourage cooperation between users.

Work in [20] examines a reciprocity-based mechanism to promote cooperation between peers who have different expertise in providing several types of resources. A simulation study is performed and in each iteration, the peers randomly issue a request in an area of which they are not experts and ask other peers for help. Peers then decide to help or not depending on their strategies and their expertise. An agent cannot help in an area in which it is not an expert. It is assumed that when an expert helps a non-expert incurs a cost of 10 and the non-expert saves a cost of 1000. Inside the network there are both selfish and reciprocative peers. Selfish peers do not serve other peers' requests. 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. A reputation mechanism is also proposed in order to help peers take interaction decisions in the early stage of their lifetime, when they do not know the behavior of the others. The simulation results have shown that under proposed framework reciprocative users' wealth (number of requests satisfied) is much better than the one of selfish users. This study does not consider the various capabilities or demands of all peers, rather it is considered that the capabilities of the peers are always enough to satisfy the demands of their requesters.

In [2] a reciprocation-based economy is studied for multiple services in p2p grids. Peers' resource requirements are combinations of two different basic services (e.g., processing power and storage). Peers can either be collaborative or free riders. Simulations proceed in turns and at each turn a peer can be either in consuming or in non-consuming state with some probability. When not in consuming state, collaborative peers donate the use of their spare resources, while free riders go idle. 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* where peers are totally unaware of other peers' service evaluation profiles. In [2] 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 turn the candidate peer with the highest cost balance. As far as the server selection policy is concerned, peers at a given turn select all the servers in a random order, seeking someone to serve them.

Both [2] and [20] face the same limitations. The individual peers' demands and capabilities are not considered in the allocation decisions, rather it is considered that the capabilities of the peers are always enough to satisfy the demands of their requesters. Moreover, they do not apply a proper reputation/trust-based server selection policy in order to avoid requesting service from misbehaving peers and last, they do not investigate the possible formation of peers' coalitions.

2.3 Single Capacity-Limited Link (SC) Systems

Single capacity-limited link systems have been previously studied in [34]. This work implicitly provides incentives for cooperation by setting auctions in which peers buy and sell to each other their access link capacity. Two request generating strategies are proposed that determine the bandwidth rate that each peer will request. Proposed strategies are combined with two different auction types which determine the rate that is given to the requester, as soon as his demands are smaller than those of the currently served peer; otherwise the requester is not served. The numerical results showed that the combination of their Greedy Rational request strategy with a second price auction lead users to the optimal operation of the network, in which peers offer half of their capacity for uploading and they use the rest for downloading.

The model in [34], though, is restricted to the case of every peer performing at most one download and one upload at a time. Moreover, the study is only limited to a homogeneous system of peers with the same capacity. As the authors admit, in the general case of heterogeneous links, their theorem indicates that the slowest link will be the bottleneck of the system. This is a serious drawback of the proposed scheme, as even powerful peers with high capacity will not be able to download with a rate higher than $\min_{i \in N}\{C_i\}/2$, where C_i is the link capacity of peer i and N is the set of all peers in the system. This limitation is not present when our policies are used, which exhibit that peers will receive resources in proportion to their contributions even in heterogeneous systems of peers with different capacities and request generation profiles. We further consider the general case of multiple uploads and downloads.

Other work in SC systems include [54] which proposes an optimization framework into reputation mechanisms for single capacity-limited link p2p systems. A separate utility maximization problem is solved by each peer, who allocates a portion of his link bandwidth to his own downloads and the remaining bandwidth for serving incoming requests by other peers. The optimization is carried out under a constraint on the level of *dissatisfaction* the peer intends to cause by not fulfilling others' requests. Proposed scheme is evaluated in a toy system of 5 peers and exhibit the peers' tension to obtain utility in accordance to their intention to dissatisfy others.

Another utility optimization framework for single capacity-limited link systems is proposed in [55] where peers of different utilities functions are investigated. However, the objective of this framework is the *socially welfare*, by identifying a socially optimal operating point in terms of bandwidth allocation and therefore it seeks to solve a different problem than the one investigated in this thesis.

2.4 BitTorrent

BitTorrent (BT) was created in 2001 by Bram Cohen who also developed the first BT implementation, the official BT client. Since then, BT has gained a tremendous popularity among content distribution community. There is a vast amount of work that has studied BT either analytically or via experiments and extensive simulations.

The first analytical study of BitTorrent was given by [56] where the authors considered a simple fluid model to conclude that the scalability and efficiency of the protocol are very good. An interesting point that they found is that although there is no Nash equilibrium point for a system of peers with heterogeneous upload capacities, there does exist one in which peers upload their total physical capacity, when the network consists of groups of peers who have the same uploading and downloading capacities. Although this model provides valuable insight, it is based on unrealistic assumptions, such as that all participants share global system knowledge (uploading rates) and that the pieces are uniformly distributed among peers. Moreover, experimental study of [57] exhibited that the arrival, abort, and departure processes of downloaders do not follow a Poisson distribution as considered in this analysis. Finally, several important BT parameters have not been considered, such as the peer set and active set sizes.

Another very interesting analytical study that followed is the one in [58] which presents the fundamental trade-off between performance and fairness of BT peers. It is argued that the improvement on fairness among peers comes at a cost of overall performance. Once again, the applicability of this model in real scenarios is dubious since several unrealistic assumptions are made, like the assumption that all peers of the same type (same uploading and downloading capacities) get the same uploading and downloading rates at any time.

Work in [59] was among the first experimental ones which used the logs of a tracker as well as one instrumented client, participating in a very popular torrent, the 1.77GB Linux RedHat 9 distribution, for its 5 months of activity. Experimental results exhibited a significant variation in the download rates of the various participating peers and showed that the upload and download throughputs of peers are not correlated, providing the first hints for unfairness in BitTorrent.

One of the most recent experimental studies was from Legout et al [40] which took place on Planetlab by monitoring all peers participating in a torrent. An interesting finding of theirs is that the initial seed upload capacity is critical to the performance of the system. When the seed is well-provisioned, peers are clustered according to their upload capacity and higher capacity peers download faster. On the other hand, when the initial seed is underprovisioned, clustering is no longer the case while most peers download at approximately the same time irrespective of their upload capacity, indicating unfairness in the system.

Work in [25] was the first study which addressed the fairness weaknesses of BitTorrent's TFT reciprocation policy. A thorough simulation study of theirs, considering real world traces, exhibited that BT's reciprocation policy fails to prevent unfairness across peers in terms of volume of content served. This becomes more evident in heterogenous settings where high bandwidth peers receive much less resources than low bandwidth ones. Another interesting point of their investigation is that fairness is balanced in the system when the tracker groups the peers according to their capacities, such as high capacity peers connect to other high capacity ones and lower capacity with lower capacity ones; in this way the uplink capacity of the high capacity peers is much better utilized. However, this concept is based on the assumption that peers will truthfully report their upload capacities to the tracker as soon as they connect to it, which can not be guaranteed.

Another work in [36] open questioned the widely held belief till then that "incentives build robustness in BitTorrent" [56, 60, 61]. They found out that the success of BitTorrent relies on the significant altruism of participating peers and exhibited how their modified BiTorrent client, called BitTyrant, can strategically game BitTorrent and significantly improve its download performance for the same level of upload contribution and degrade the performance of other BT clients. BitTyrant can accomplish this by dynamically adapting both the uplink capacity and the active set size, while preferentially selecting peers according to their contributions, instead of randomly optimistically unchoking them.

There were several other studies that investigated strategic clients in BT. Work in [62] investigated three exploits over BT clients, i.e., a) download only from seeds, b) download only from the fastest peers and c) advertise false pieces. They showed that although these kind of strategic peers can improve their performance, BitTorrent is quite robust against these exploits. Work of [63] proposed another BT strategic peer which can significantly improve his performance by allowed to open much more connections than the official client and by using random piece selection policy in order to ensure that an unchoke period is never left unused; however the effect such peers could have in the system performance is not explored.

A following work of Carra et al [64] has studied, in publicly available simulator GPS [65], the performance of BitTyrant [36] and demonstrated that although its existence in a BT system can have a surprisingly positive impact on the content distribution process, the system's performance deteriorates when many such peers appear. In the same setting, a new released version of the mainline BT has been tested which presents a significant performance improvement in terms of download times, due to an allowed large number of active connections for high capacity peers.

The performance improvement of peers who have a larger than normal view of a BitTorrent swarm and maintain much more connections, has also been pointed by work in [41]. Their modified freeriding BT client who, on one hand, does not share any pieces of the file and, on the other hand, uses more connections than the default in mainline BT, succeeds in significantly improving his downloading completion time. More specifically, these free riders outperform compliant clients on average, except when free-riders dominate the swarm, in which case both compliant and free-riders incur substantial performance degradation.

From all aforementioned studies, it is clear that BT is vulnerable to strategic clients and fails to shield the system from misbehavior. Responsible for these deficiencies of BT protocol was mainly considered to be the reciprocation strategy of BitTorrent [31, 37, 41, 42, 62, 63]; on one hand, the optimistic unchoke policy and on the other hand, the incapability to find the right match, allow low capacity peers to gain more than they invest, compared to high capacity peers. For this reason, there were several studies that proposed improvements over it. One of these proposals was from Eger et al [31]. Their proposed reciprocity mechanism was inspired by their previous resource pricing mechanism introduced in [66], according to which a peer adjusts his bandwidth rates to his various customers based on the amount of bandwidth he receives from them, which is used as the pricing information. It is proved that the equilibrium in the steady state is a Nash equilibrium where each peer uses his full upload capacity, and downloads from a specific peer with the same rate that he uploads, preserving fairness.

Simulation studies are also conducted in order to evaluate the performance of the proposed scheme, although the authors do not explain how they bootstrap the system. An overlay topology is constructed according to the original BT implementation. Several probability functions are used to model the BT behavior, like the probability that a specific peer ID is returned by the tracker. The block selection algorithm is omitted and it is assumed that connected peers are always interested in blocks of each other. Based on these assumptions, the performance of BT TFT mechanism is compared with the performance of the proposed policy. It is generally shown that the proposed policy exhibits a fairer allocation of resources than the reciprocation mechanism of BT with which peers with small upload capacities receive considerably more than what they contribute, compared to others. Especially when peers have a small number of connections, download rates vary considerably. The algorithm, though, is restricted to peers that did not complete the download already. Seeds cannot perform this algorithm and still lack incentives to contribute resources in the network.

Jun et al. [37] is among other works that argue that the BT choking algorithm is susceptible to free-riding and proposed a new incentive mechanism based on the iterated prisoner's dilemma. Peers take allocation decisions, which prove to be more robust than current TFT mechanism of BT, based on a "deficit" defined as the upload amount minus the download amount for a given link. However, their experimental study considers only free riders who contribute much less than others and do not examine cases of misbehaving peers who provide bogus blocks or try to exploit the system in a different way.

Similar to [37], Sherman et al. [38] use a deficit counter which represents the amount of data their modified FairTorrent client owes to another one. FairTorrent clients decide whom to upload the next data block based on such deficit counters. Their results demonstrate that FairTorrent provides up to two orders of magnitude better fairness; however, seeders still lack incentives for cooperation.

A recent study of Li et al [39] argues that free riding effect on BitTorrent is mainly due to the altruistic behavior of seeds and not so much due to the BT reciprocation policy. They proposed a seed allocation policy according to which seeds favor only the most cooperating peers in order to mitigate free riding. However, this study is based on the truthfull reporting of neighboring peers about their downloading rates from the competing requesters of seeds, which may not be the case in a real system of antagonistic peers.

Work in [35], on the other hand, revealed through extensive trace analysis and modeling that the severe fluctuations in the peers' download performance is due to the exponentially peer arrival rate. However, they showed that multi torrent collaborations can alleviate these effects and provide strong motivation for peers who have already downloaded a file to stay in the system and exchange it with a new file they are interested in. Nevertheless, still this policy cannot motivate a seed to remain and contribute in the system if he is no longer interested in another file, while a reputation mechanism could boost the reputation of a such a peer who could use it in his future requests to improve his performance.

Work in [24] is the first attempt to incorporate a reputation mechanism in BitTorrent in order to motivate peers to remain in the system and contribute their resources as soon as they have downloaded a file. Upload and download statistics are used to to compute a local reputation for each peer, based on which peers decide whom to assign upload slots. Peers are distinguished to *freeriders* who immediately leave the swarm after completing their download and *sharers* who share every downloaded file for 10 hours after completion. However, in practice a *freerider* is not only a peer who disconnects as soon as he completes download but also provides bogus blocks and has a low sharing ratio even when he is connected to the swarm. The policies in this work do not shield the system from this kind of behavior. Moreover, only homogenous settings of peers with the same pairs of upload-download capacities are considered and the results are not so satisfactory, since although the performance of *freeriders* deteriorates, so does the performance of the *sharers*.

In order to confront free-riding in BitTorrent, work in [42] proposed a trust management system for BT according to which each peer (including seeds) decides which peers to unchoke according to their global trust level which depends both on the contribution level of the peers and their behaviors (whether they upload bogus blocks). Their simulation study in GPS [65] shows that their policies improve the fairness ratio (number of blocks uploaded over those downloaded by each peer) compared to BT, as soon as peers can obtain the global trust values of the other peers from the tracker. The global trust values are computed by the tracker by using the local opinion of 4 randomly chosen peers. In this way, though, their policy is susceptible to false reporting from misbehaving peers. Moreover,it induces a severe communication overhead, as the tracker should communicate at regular intervals with 4 peers for the trust reports for each peer in the system and with all peers to inform them about the trust levels of others. Furthermore, if seeds are also using the proposed allocation policy, as hinted by the authors, they would cease servicing peers as soon as the total number of blocks they downloaded from them minus the total number of blocks they uploaded to them falls below a certain threshold. This tactic, though, would deter seeds from contributing their resources, instead of motivating them.

Finally, work in [67] considers resource allocation policies for BT peers which improve peers' download rates in cooperative settings of peers who seek to reach a common goal, e.g., P2P clients of an organization who participate and cooperate in a private torrent that is set up by a system administrator to disseminate a software update or a new application. This work is totally different than aforementioned papers whose focus is how to enforce peers to cooperate inside an antagonistic framework.

2.5 Wireless Internet Sharing Community Networks

Trust and Reputation systems have been extensively investigated in file sharing systems to foster cooperation among users and improve their performance as we saw in previous sections. However, the Internet sharing context is new and reputation mechanisms haven't been tested under such settings yet. Among the work in most related settings is the one in [68], which proposes a reputation-based scheme to motivate cooperation among different Wireless Internet Service Providers (WISPs) in order to provide seamless roaming provision for their users. Proposed mechanism is controlled by a Trusted Central Authority which signs the reputation records of the WISPs. Mobile nodes use these records along with other information (i.e., requested prices and advertised QoS) to decide to which neighboring WISP to associate with and pays for the service using a credit-based micropayment scheme. Reputations are only used to choose the appropriate provider and not to determine the QoS offered to competing users asking for connection, which is the main focus of our approach. Moreover, in contrast to this work, we seek to foster cooperation among users without the need of a centralized authority or micropayment schemes.

In [69] the owners of the APs in a WCN are motivated to allow mobile users to access their home WLAN/DSL connection by the existence of a reciprocity mechanism. It is considered that WCN participants divide into teams of few tens of members each and the members of the same team must know and trust each other and have a leader. Members sign digital receipts when they consume service from another team. The receipts form a receipt graph, which is used as input to the reciprocity algorithm that identifies contributing teams using network flow techniques. The receipts are stored either in a central server, or they are distributed among multiple team servers with a gossiping protocol and are used in order to decide whether to grant service in a given requester or not.

The aforementioned described work, as a first attempt to stimulate participation in a WCN, faces some limitations. Firstly, it is based on the unrealistic assumption that the aggregate team consumption rates are homogeneous. Secondly, it is considered that the cost for a contributor to share his Internet connection is the same, irrespective of the duration of the service and the QoS offered. Our scheme, on the other hand, considers users of different consumption profiles, while the duration and QoS of offered services are important cost factors considered in the allocation decisions. More importantly, it does not examine whether users receive resources in proportion to their contributions. The reciprocity algorithm solely determines whether a given requester will be grant an Internet connection or not, and not the specific QoS that he will receive.

An extended work [70] seeks to handle this issue by proposing an enhancement on the reciprocity mechanism; however, it is still based on the unrealistic assumptions of [69] and the scheme is not evaluated in order to examine whether better contributors indeed receive a better QoS when connecting to other AP owners in the WCN. The fact that it does not consider the different consumption profiles of the users, may lead some users to receive unfairly much more resources than some others. Last, users of different cooperation levels are not modeled, in order to test their performance in a dynamic scheme.

Chapter 3

Reputation-based Framework for P2P *Single Service (SS)* Systems

3.1 Introduction

This chapter refers to systems of peers who share a single computing resource, defined as service, like bandwidth in file sharing systems (e.g., BitTorrent, Gnutella), or storage in distributed information storage systems (e.g., Freenet), etc. As we already stated in chapter 1, a reputation-based scheme seems the most appropriate mechanism to motivate users of such systems to cooperate.

The main contributions of this chapter can be summarized as follows:

a) we propose a novel reputation metric and update mechanism such that it gradually exposes the cooperation level of each peer in terms of the quality of service it provides and dynamically adapts to possible peer's behavior variations,

b) we propose an overall allocation framework according to which each peer independently decides how to allocate his available resources. Reputation-based allocation policies are followed (RA and ERA), based on which each peer determines the quality of service that he will offer to each one of his requesters according to their reputations, resource demands and request generation profiles. 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 [48]. Our proposed allocation scheme can be 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,

c) We propose a reputation-based server selection policy (RS) to help peers select among the most reputed/contributive servers and avoid misbehaving (non contributive) ones and finally,

d) we evaluate our scheme via simulations to various setups of peers with different contribution and consumption profiles, newcomers, misbehaving and strategic peers, and present the performance gains compared to previous work.

The remainder of this chapter is organized as follows. Section 3.2 describes the considered model and section 3.3 presents the proposed reputation-based exchange framework for one service, its robustness over various attacks and potential application scenarios. In section 3.4 we discuss the performance evaluation details and in section 3.5 we exhibit the results for the single service case, providing comparison results with related work. Finally, section 3.6 concludes the chapter.

3.2 Model Description

We consider a p2p network of N peers who provide and consume a particular service. We consider that the service is bandwidth for content sharing. Each peer i can provide bandwidth with different capability, according to the upload capacity ${}^{u}C_i$ (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

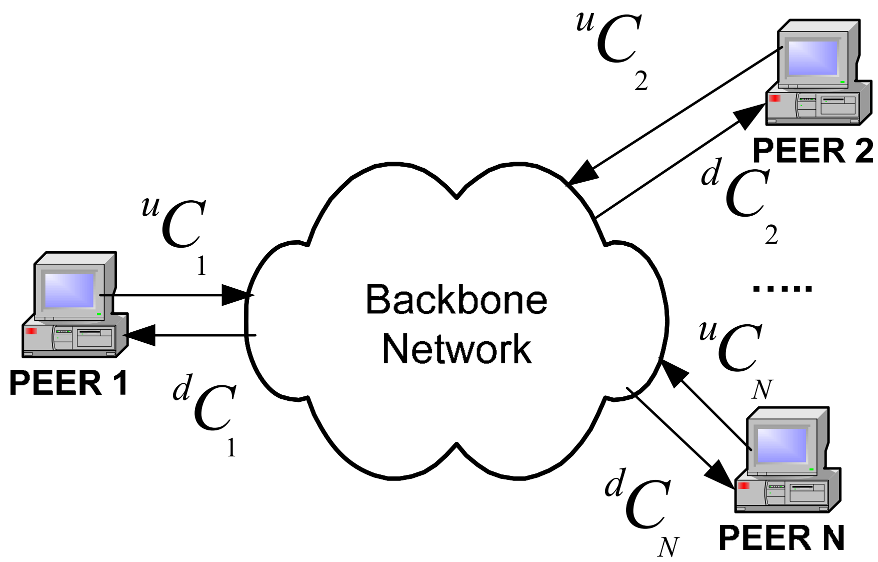


Figure 3.1: *Network Model.*

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 [13]. We assume that the physical upload capacity of each peer is the bottleneck in the network, while the download capacity ${}^{d}C_i$ 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. 3.1. In the rest of this chapter 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 fixed time units. At each period each peer i generates g_i requests and directs them to other peers. In each one of his requests he reports his demands (in terms of bandwidth), expressed by D^{p}_{ij} where j denotes the server for the given request of peer i and p denotes the period in which the request took place. 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. The demands of peer i are uniformly distributed between $[\min D_i, \max D_i]$, according to each peer's demand profile. A peer may report different demands in each one of his requests.

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, as we will see in the sequel. At each period, peers collect the requests that have been directed to them and allocate their available upload capacity among their requesters.

The service is not granted for more than a period, 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 [71], as we describe in detail in chapter 6.

3.3 Reputation-based Policies for SS systems

3.3.1 Reputation System

In this section we describe the reputation system that we propose. Each peer keeps a reputation value for each one of his transacting peers, based on the amount of resources that he receives from each one of them. In particular, if in a given period p a peer i demands a certain amount of service D_{ij}^p from a peer j and finally receives an amount of $x_{ij}^p \leq D_{ij}^p$ during this period, then the ratio x_{ij}^p / D_{ij}^p represents how much peer j satisfied his needs in this period and accounts for j's reputation in the eyes of peer i. This represents the subjective (local) reputation of peer j to i. At each new transaction with the same server the local reputation of the server to the peer is recalculated. So, the reputation of peer j to any peer i at a given period p is given by:

$$R_{ij}^p = \frac{1}{Req_{ij}^p} \sum_{t \in T_{ij}^p} \frac{x_{ij}^t}{D_{ij}^t}, \tag{3.1}$$

where Req_{ij}^{p} is the total number of i's past bandwidth requests from peer j till period p and T_{ij}^{p} denotes the set of the specific periods in which peer i requested service from peer j, till period p. In a similar way, each peer in the system calculates the local reputation of all other peers. The reputation of each peer can take any value from 0 to 1 and expresses the average satisfaction in terms of quality of service that he has offered.

Please note that the reputation of a given peer reflects the average satisfaction, in terms of quality of service, he offers to his requesters, as opposed to previous work [1, 10–12], 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.

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 newcomers prove to be collaborative enough, their reputation will quickly increase, 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 local reputation. More specifically, we performed simulation experiments with different peers' populations and profiles and saw that a 10-log most recent history was the smallest log achieving similar 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 any other peer j, he starts replacing the oldest rating transactions with newer ones, always keeping a history of, at the most, 10 past transactions with each peer j, maintaining the overhead low. Moreover, in this way the system removes age-bias and quickly adapts to peer behavior changes, e.g., when peers strategically vary their contributions through time.

3.3.2 Reputation-based Allocation Policy (RA)

Suppose that a peer j receives in a given period p, n requests for bandwidth and that his upload capacity is C_j. We use an identifier vector $\mathbf{I} = \{I_1, ...I_n\}$, each element of which represents a specific requester of peer j at period p with demands expressed by $D^p_{I_i j}$, $i \in \{1,..,n\}$. Then, we propose that j will allocate his upload capacity C_j among the n competing peers by following the maximization problem:

$$max \sum_{i=1}^{n} R^p_{jI_i} \frac{x^p_{I_i j}}{D^p_{I_i j}},$$
$$\text{s.t.} \ \sum_{i=1}^{n} x^p_{I_i j} \leq C_j \ \text{ and } \ x^p_{I_i j} \leq D^p_{I_i j} \ \forall i, \tag{3.2}$$

where $R^p_{jI_i}$ represents the local reputation of peer I_i to j at period p, and $x^p_{I_i j}$ represents the allocated resource of peer j to peer I_i during period p. We will refer to this policy as RA.

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.

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 (3.2) are linear[1], so

[1] We also experimented with a non-linear policy, i.e., $max \sum_{i=1}^{n} R^p_{jI_i} log(\frac{x^p_{I_i j}}{D^p_{I_i j}} + 1)$ with the same constrains as in (3.2); we preferred the simpler linear one, as both policies perform equally well when our proposed reputation

the solution can be easily found by sorting peers in decreasing order according to their reputation to demands ratio $R^p_{jI_i}/D^p_{I_ij}$. 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. If two or more peers have the same reputation to demands ratio and the residual capacity of the server j is not enough to accommodate all of their demands, it will be equally split among them, unless this portion exceeds the demands of some of them and thus the residual is allocated to the others.

Applying RA, peers with the highest ratio $R^p_{jI_i}/D^p_{I_ij}$ will be prioritized. 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).

It is obvious from the solution of (3.2) that for $C_j < \sum_{i=1}^{n} D_{I_ij}$ and for any two competing peers i and k with $\frac{R^p_{ji}}{D^p_{ij}} < \frac{R^p_{jk}}{D^p_{kj}}$, their satisfaction ratios from their bandwidth requests will satisfy $\frac{x^p_{ij}}{D^p_{ij}} < \frac{x^p_{kj}}{D^p_{kj}}$. This statement proves that when the resources of a given server j cannot satisfy the demands of all the competing peers, the ones with the highest contribution level per unit resource request will perceive 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 in a loaded network. As an example, consider 3 peers A, B and C who request service from peer D with $C_D = 6Mb/s$. Their demands are $D_{AD} = 3Mb/s, D_{BD} = 5Mb/S, D_{CD} = 5Mb/s$ 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.

system is used.

The benefits of RA policy are also exhibited through the simulation results of section 3.5. 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 3.5.1) and strategic peers fail to maximize their obtained resources when they do not offer a proportional amount of resources to the network (section 3.5.4). Our simulation results further exhibit the performance stability of the system in the presence of newly arriving peers (section 3.5.2) and the formation of coalitions between peers of similar capabilities and needs (section 3.5.3).

3.3.3 Enhanced Reputation-based Allocation Policy (ERA)

Although RA policy is suitable for systems of peers with similar request generation profiles, it fails to be fair when there are peers who produce many more requests than others, in order to exploit the network. Such peers should be properly handled, so as not to absorb most of the resources of the system, in cost of other collaborative users. The following allocation policy (ERA) is proposed to cope with this case. According to ERA policy, peer j will share his capacity C_j to n competing peers by solving:

$$\begin{aligned} & max \sum_{i=1}^{n} \left[(\frac{Req^p_{jI_i}}{Req^p_{I_i j}})^a \times R^p_{jI_i} \right] \frac{x^p_{I_i j}}{D^p_{I_i j}}, \\ \text{s.t.} \; & \sum_{i=1}^{n} x^p_{I_i j} \leq C_j \;\text{ and }\; x^p_{I_i j} \leq D^p_{I_i j} \;\forall i, \end{aligned} \tag{3.3}$$

where $Req^p_{jI_i}$ is the total number of requests that peer j has sent to each peer I_i till period p. For any new peer in the system, these variables are set to 1 and change through time. Variable a can be adjusted by the mechanism designer to balance the fairness in the p2p system. The bigger a is, the more restrictive the policy is for the high rate peers.

With ERA policy we provide better quality of service to those peers to whom we have sent more requests than what they have sent to us, in order to somehow repay them. 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, 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.

The solution of (3.3) can be easily found by following a similar procedure with the one in (3.2), indicating that high rate peers need to be more contributive than low rate peers, in order to satisfy their requests in a loaded network. This is also exhibited through the simulation results of section 3.5.5.

3.3.4 Reputation-based Server Selection Policy (RS)

In this section we describe a reputation-based server selection policy in order to help peers avoid requesting service from misbehaving (non contributive) ones. 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. So, for a short period of time, which we call "acquaintance duration", new peers direct their requests to all peers with equal probability (random selection policy), till they obtain an overview of the network. Similarly, preexisted peers in the system will use the random selection policy among their old providers and new peers in the system 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 j at a given period p, is directly proportional to j's local reputation to i, as:

$$p_{ij}^{p} = R_{ij}^{p} / \sum_{k \in S_i^p} R_{ik}^{p}, \tag{3.4}$$

where S_i^p is the set of all peer i's transacting peers till period p. A similar policy is used in [1] and [16], where though corresponding probability is proportional to a global reputation and not a local one. As we already explained in a previous section, by using local reputations peers tend to select the appropriate traders for them, according to their particular needs.

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 potential servers for future peers' requests.

3.3.5 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 3.3.1 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 3.3.2. Moreover, according to our policies, as soon as peers' 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 reputation-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 the evaluation results of sections 3.5.1 and 4.5.6.

Whitewashing and sybil attacks

Under both Sybil and whitewashing attacks, peers create many pseudonymous entities in order to exploit the network. Sybil attack is common in global reputation systems which are based on peers' votes. By creating many pseudonymous entities, a peer can provide a large number of positive votes for himself cheating the others. In our work, 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 3.5.2.

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 3.5.4.

3.3.6 Application Scenarios

Our proposed reputation-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 and 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 [5] 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 reputation 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 reputation-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 chapter 7 we explain how our scheme can be incorporated in BitTorrent, to overcome well-known deficiencies of its current reciprocation mechanism.

3.4 Performance Evaluation

3.4.1 Simulation Model

We have developed an event-driven simulator in C++ to evaluate the performance of our policies. 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. The performance of the system scales even to larger overlay networks; however in larger networks the convergence time of the reputations increases.

All peers start with a small initial reputation 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.

3.4.2 Comparison with other schemes

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

Scheme in [1]

Scheme in [1] 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 k that request the same service from a particular server j, the probability for a specific requester i to be selected equals $R_i/\sum_k R_k$, 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 [1] to our model, because the model in [1] does not account for peers of different capabilities and demands. Described scheme will be denoted as BRR in the figures of this chapter standing for **B**inary-**R**ating **R**eputation-based scheme.

The following alternative allocation policies share similarities with RA and are described below for the case of n requests directed to peer j at period p. Their performance is exhibited in section 3.5.1 in comparison with the performance of RA.

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 chapter.

Maximize the satisfactions of the peers:

This policy maximizes $\sum_{i=1}^{n}(x^p_{I_i j}/D^p_{I_i j})$, under the constraints $\sum_{i=1}^{n} x^p_{I_i j} \leq C_j$ and $x^p_{I_i j} \leq D^p_{I_i j} \forall i$. The solution can be found in a similar way as the solution of (3.3). 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 chapter.

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

This policy maximizes $\sum_{i=1}^{n} R^p_{jI_i} x^p_{I_i j}$, under the following constraints $\sum_{i=1}^{n} x^p_{I_i j} \leq C_j$ and $x^p_{I_i j} \leq D^p_{I_i j} \forall i$. It gives more resources to more reputed peers. This policy will be denoted as max (Rx) in the figures of this chapter.

3.4.3 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 average contribution level of a peer is reflected on his global reputation in the network. The global reputation of peer i at a given period p is given by averaging the individual opinions of every other peer in the N-peers system for the given peer i at period p (local reputations), i.e.,

$$GR^p_i = \sum_{\forall j \neq i} R^p_{ji}/(N-1), \tag{3.5}$$

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 St_i^p of each peer i in the system. The average satisfaction of peer i until period p, is given by summing his satisfaction ratios in each one of his transactions with all other peers and averaging over the total number of his requests till period p.

3.5 Simulation Results

3.5.1 Peers with Different Capabilities and Misbehaving Peers

In Fig.3.2 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 RA 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 3 requests per period and requests are directed with equal probability to each one of the rest of the peers (random server selection policy).

We see that with RA 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.3.3 the lower capacity (3 & 4 Mb/s) peers of the experimental case exhibited in Fig.3.2 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.3.3a 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)

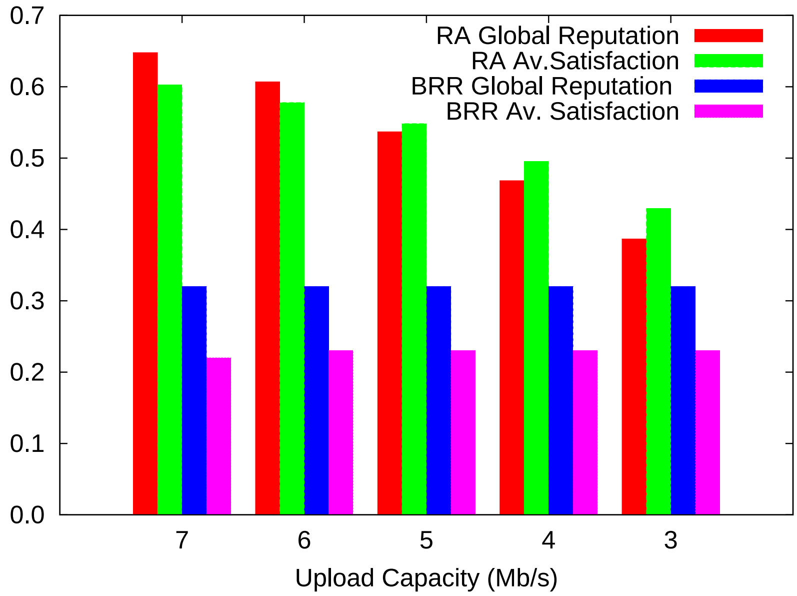


Figure 3.2: *Global Reputations and Av. Satisfactions of different capacity peers.*

if they use the reputation-based server selection policy.

In Fig.3.3b we see the satisfaction ratios of each category of peers when the various allocation policies described in section 3.4.2 are used, combined with the reputation-based server selection policy. It is clear that only our proposed allocation scheme RA, 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 refrains misbehaving peers from exploiting the system, it cannot differentiate collaborative peers. In the same figure we can see the performance of two other policies, RA-BR and CS-QR. The first one is the combination of our allocation policy RA with the **B**inary **R**ating-based reputation metric of scheme in [1], while the second one is the combination of the **C**lient **S**election policy of scheme in [1] 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

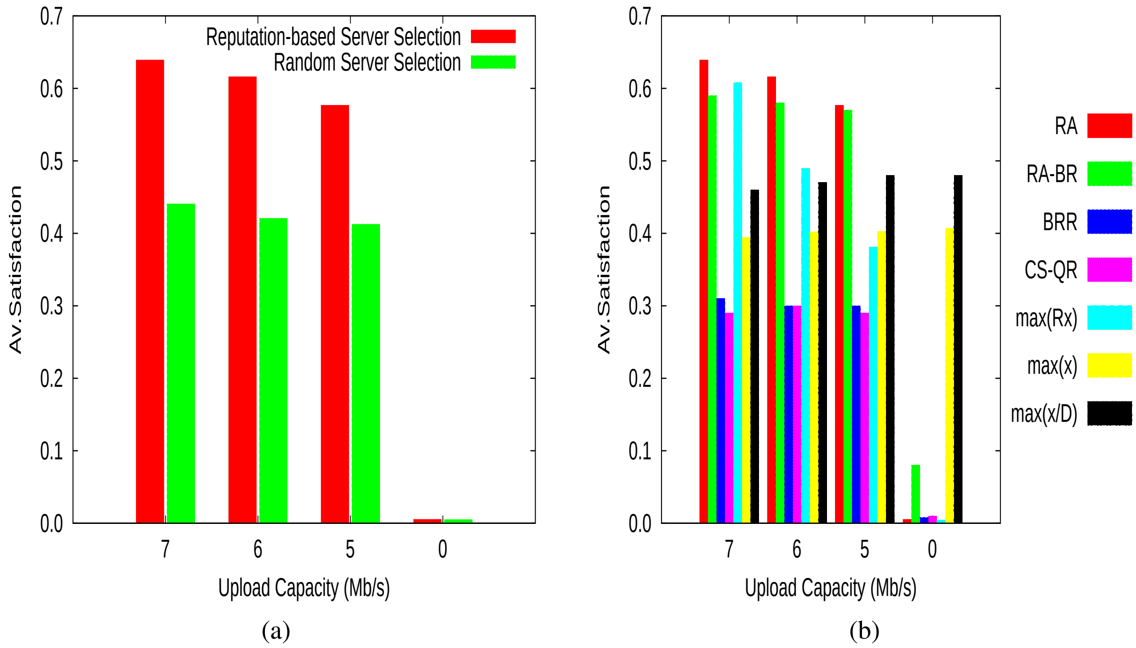
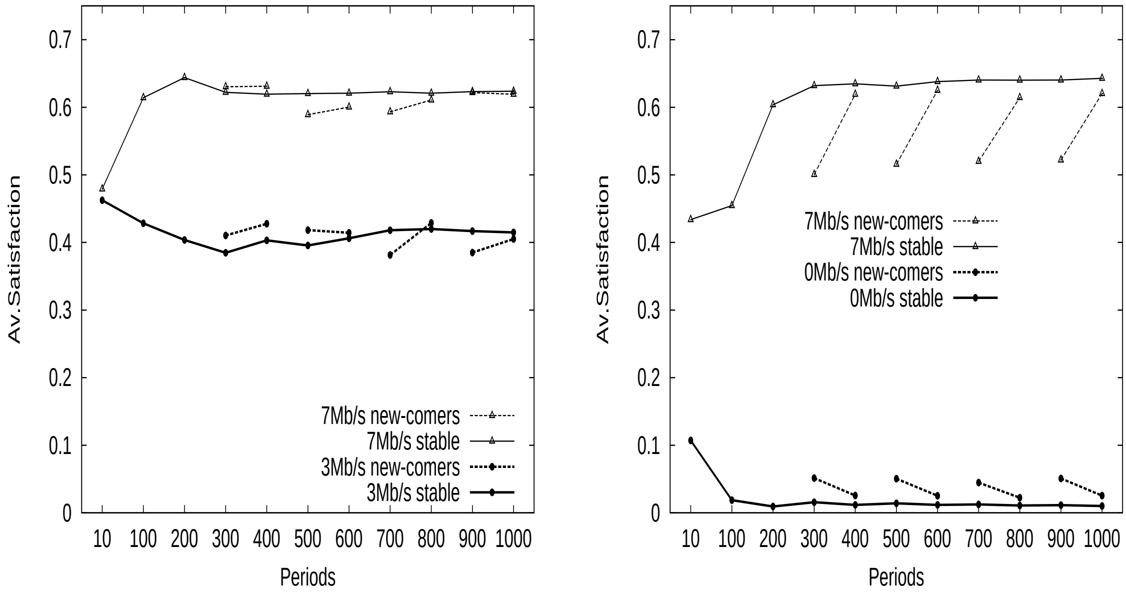


Figure 3.3: *Av.Satisfaction of different capacity peers for 40% misbehaving peers and different (a) server selection and (b) allocation policies.*

reached the same steady state with a convergence time in the order of N^2. However, even after only 1000 rounds, for all aforementioned populations the service differentiation is succeeded. The only difference 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 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 practice, 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.



(a) (b)

Figure 3.4: *Average Satisfaction of newcomers and stable peers of different capacities for (a) no misbehaving peers and (b) 40% misbehaving peers in the system.*

3.5.2 NewComers & Whitewashers

In this section we examine the performance of the RA 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 3 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 our reputation-based allocation and server selection policies (RA and RS).

In Fig.3.4a 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 0 resources) and, as above, every 200 periods 50% of each category peers are replaced by new identity peers of the same capacities. New identity misbehaving peers may be whitewashers who have left the network and appeared with new identities in order to exploit the system. However, as we see from Fig. 3.4b 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 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.3.4b, 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.

3.5.3 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 capabilities and demands (weak peers). All peers generate 3 requests per period and they use the RA policy. In Fig.3.5a 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,

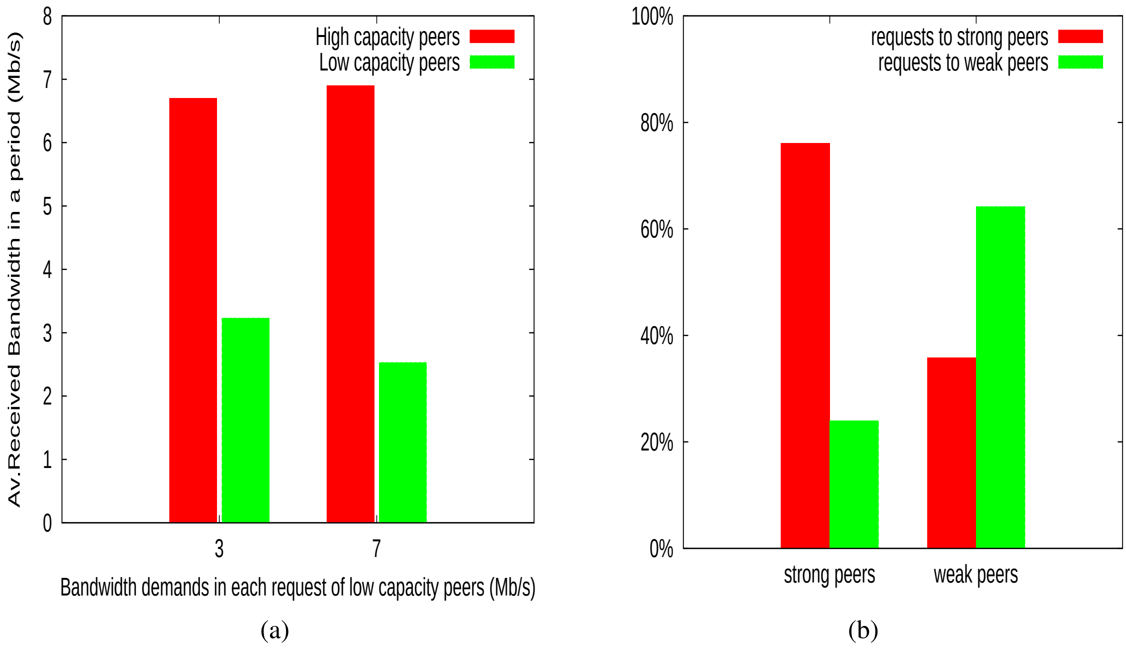


Figure 3.5: *(a) Av. Received Bandwidth per period for high and low capacity peers (b) Percentage of requests directed from and to each category of peers.*

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 (please refer to Fig.3.5a). 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.3.5b 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.3.5b 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.

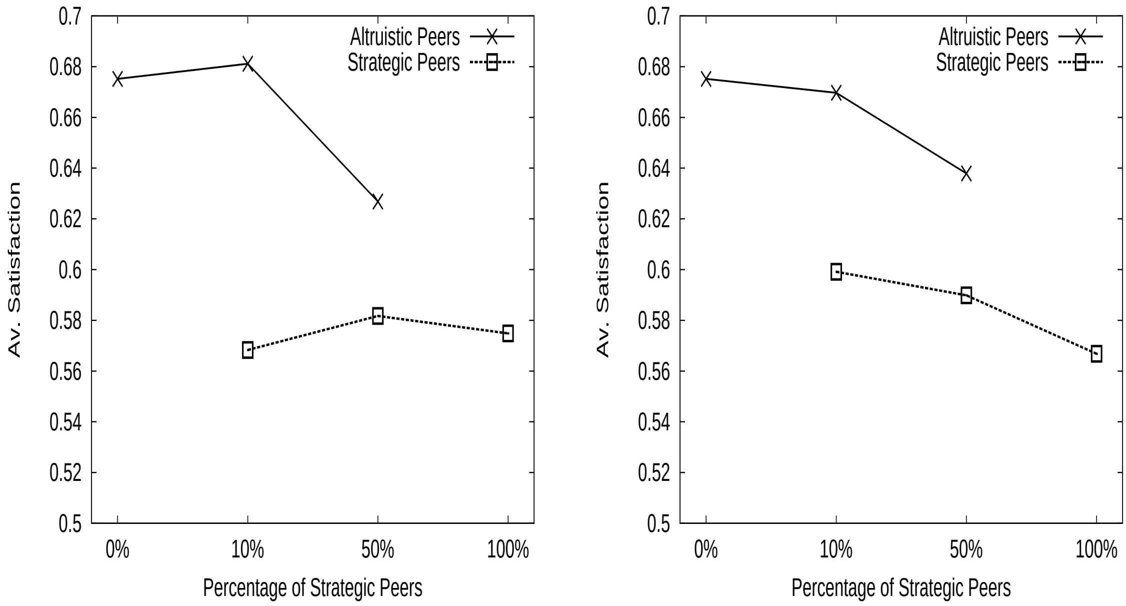
3.5.4 Strategic Peers

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 RA, 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.

In Fig.3.6 we see the average satisfaction of altruistic and strategic peers for different percentage



(a) (b)

Figure 3.6: *Average Satisfaction for different percentage of strategic peers when the latter adopt (a) the progressive strategy or (b) the steep strategy.*

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.3.6 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.3.6), 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.3.6), 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.

Table 3.1: *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
RA	2.19	1.94	8.76	3.88	0.69	0.64
ERA (a=1)	1.70	2.90	6.80	5.80	0.57	0.82
ERA (a=2)	1.56	3.19	6.24	6.38	0.51	0.87

3.5.5 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 4 requests per period, while 50% of them are lower rate (LR) peers generating 2 requests per period. In Table 3.1 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, RA and ERA with $a = 1$ or $a = 2$ are used by all peers.

We note that with BRR and RA, 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 ERA. It is remarkable that by using ERA 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. ERA 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 3.2 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 ERA and $a = 2$ they receive better service per period than lower capacity ones since they provide more resources

Table 3.2: *Performance of HR and LR Peers for Different Policies and Capacities.*

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

than them. On the other hand, when HR peers are the low capacity ones, their performance is much restricted by ERA 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, ERA is almost identical with RA. We say “almost” because when the reputation-based server selection policy is used, ERA slightly favors the most reputed peers since most of the requests are directed there.

3.6 Summary

In this chapter we presented a reputation-based framework to fairly regulate the exchange of one service in p2p like systems of peers with different contribution and consumption profiles. We observed that the proposed policies shield the system from (a) misbehaving peers and whitewashers, (b) strategic peers who seek to maximize their satisfaction 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. Moreover, we presented how our policies greatly outperform previous representative reputation scheme which cannot differentiate peers according to their contribution levels, but solely binary categorize them in altruistic and egotistic. Under such schemes, all cooperative peers, irrespective of the QoS they offer, have the same reputation and finally perceived satisfaction from their requests, lacking incentives to improve their contributions.

Peers benefit by using our proposed 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 can be 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.

Part of the work described in this chapter has been published in the following journal and conference proceedings:

* A.Satsiou, L.Tassiulas, Dynamic Cooperation Enforcement through Trust-based Allocation Policies, in Proc. of the 4th EURO-NGI International Conference on Next Generation Internet Networks, Krakow, Poland, 28-30 April 2008.

* A.Satsiou, L.Tassiulas, Trust-Based Exchange of Services to Motivate Cooperation in P2P Networks, accepted for publication in Peer-to-Peer Networking and Applications Journal, Springer.

Chapter 4

Reputation-Based Framework for P2P *Multiple Services (MS)* Systems

4.1 Introduction

Chapter 3 and related work reported in section 2.1 are focused in the case that one single service is exchanged in the network (e.g., processing power [11], bandwidth [9], 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.

In this chapter we extend our reputation-based policies introduced in chapter 3 to apply to systems where more than one services may be exchanged and to overcome the various difficulties arisen in such systems, as already underlined in section 1.4. The goal is to foster cooperation and help peers recognize and transact with other peers with whom they can mutually benefit.

The main contributions of this chapter can be summarized as follows:

a) we propose a reputation vector consisting of the reputations of the peers in providing each system service, based on which allocation decisions are taken,

b) we extend our allocation policies to form RA-MS and ERA-MS in order to incorporate the reputation vectors of the competing peers, as well as the service evaluation profile of the server in the allocation decisions,

c) we adapt our server selection policy to multiple services systems (RS-MS) and finally,

c) we evaluate our scheme via simulations to various setups of peers with different contribution, consumption and service evaluation profiles, newcomers and misbehaving peers, and present the performance gains compared to related work [2] in multiple services systems.

The remainder of chapter 4 is organized as follows. Section 4.2 presents the system model and section 4.3 describes the proposed reputation-based exchange framework for multiple services, and potential application scenarios. In section 4.4 we discuss the performance evaluation details and in section 4.5 we exhibit the results, providing comparison results with related work in multiple services systems. Finally, section 4.6 summarizes this chapter.

4.2 Model Description

We consider a p2p network of N peers who provide and consume multiple computing resources, defined as 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 S services in the system, the capability profile of a peer i is $\mathbf{C}_i = \{C_{i1}, ..., C_{iS}\}$, where C_{is} represents the capability of peer i in providing service s.

Each peer is further characterized by his service value profile $\mathbf{V}_i = \{V_{i1}, ..., V_{iS}\}$, where V_{is} represents the value put from peer i on a unit of service s, 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 s, he incurs a cost of V_{is}, whilst each time he receives a unit of resource s, he has a profit of V_{is}. 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 $\mathbf{M}_i = \{\{D_{i1}, G_{i1}\}, ..., \{D_{iS}, G_{iS}\}\}$, where D_{is} represents the average demands of peer i for service m and G_{is} represents the average request generation rate of peer i for service s. The demands of peer i for service s are uniformly distributed between $[\min D_{is}, \max D_{is}]$, according to the particular needs of each peer i for service s.

We consider that our system progresses in periods. In each period, each peer i generates g_i requests. Each one of the g_i requests is for a service s with probability p_{is}, such that $g_i \times p_{is} = G_{is}$. In each one of his requests, peer i reports his demands for the requested service s which can be certain bandwidth, storage, CPU cycles, etc, for Bag-Of-Tasks [72] or other applications and are expressed by D^p_{ijs}, where j denotes the server for the given request and p the period during which the request took place. As explained in chapter 3 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.

4.3 Reputation-based Policies for MS systems

4.3.1 Reputation vector

In this section we describe the reputation vector we propose for MS systems. 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., $\mathbf{R}_i^p = \{R_{ijs}^p, \forall j, s\}$, where R_{ijs}^p represents the local reputation of peer j to peer i for service s at a given period p, and is calculated as follows: if in a given period p, a peer i demands a certain amount of service s, D_{ijs}^p, from a peer j and finally receives an amount of $x_{ijs}^p \leq D_{ijs}^p$ during this period, then the ratio x_{ijs}^p / D_{ijs}^p represents how much peer j satisfied his needs for service s in this period. Then, he can calculate the local reputation of peer j in providing service s at a given period p by following the formula:

$$R_{ijs}^p = \frac{1}{Req_{ijs}^p} \sum_{t \in T_{ijs}^p} \frac{x_{ijs}^t}{D_{ijs}^t}, \tag{4.1}$$

where Req_{ijs}^p is the total number of i's past bandwidth requests from peer j for service s till period p and T_{ijs}^p denotes the set of the specific periods in which peer i requested service s from peer j, till period p. In the same way, the reputation of any peer is calculated.

Each peer i further keeps a vector with the overall local reputations of all other peers $\widehat{\mathbf{R}}_i^p = \{\widehat{R}_{ij}^p, \forall j\}$. The overall reputation of peer j to peer i at a given period p, $\widehat{R}_{ij}^p$, is given by averaging the satisfaction ratios of peer i from all his service requests from server j 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. As for the SS systems studied in chapter 3, we have seen 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. Therefore, each time a peer i completes 10 transactions with peer j for service s, 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 j and service s, keeping the overhead low. If the log has fewer inputs (e.g., due to limited interactions with peer j for service s), reputations are calculated based only on those. In order to clarify these issues, we describe the following example. Suppose that there are 2 services provided in the system, $s1$ and $s2$. If peer i has the following recent log for peer j 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 j 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 j 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 j as $(2.4+0.4)/14 = 0.2$.

4.3.2 Reputation-based Allocation Policy for MS systems (RA-MS)

In this section we consider that peers may exchange more than one services. We extend the RA policy for multiple services p2p systems, regarding the peers' service evaluation profiles. Suppose that a peer j receives in a given period p, n requests for a given service s and his capacity for service s is C_{js}. We use the vector $\mathbf{I} = \{I_1, ...I_n\}$ for representing the different requesters of peer j with demands expressed by $D^p_{I_i js}, i \in \{1,..,n\}$. Then, we propose that peer j will allocate C_{js} in order to solve the maximization problem:

$$max \sum_{i=1}^{n} \sum_{m=1}^{S} (V_{jm} \times R^{p}_{jI_i m}) \frac{x^{p}_{I_i js}}{D^{p}_{I_i js}}$$
$$\text{s.t. } \sum_{i=1}^{n} x^{p}_{I_i js} \leq C_{js} \text{ and } x^{p}_{I_i js} \leq D^{p}_{I_i js} \forall i, \tag{4.2}$$

where $x^{p}_{I_i js}$ represents the allocated resource of peer j to peer I_i for service s during period p.

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 RA-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 j provides more resources to those peers who 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 4.5 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 4.5.1), (b) peers have random service value profiles (section 4.5.2), (c) their service value profiles are correlated with their capability profiles (section 4.5.3) and finally, (d) their service value profiles change through time (section 4.5.5).

As in RA, 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 (4.2) can be found following the same procedure as in RA. Similarly, it can be proved that peers with the highest weighted reputation per unit resource request will receive the highest satisfaction.

4.3.3 Enhanced Reputation-based Allocation Policy for MS systems (ERA-MS)

ERA policy can also be extended for the multiple services case by applying (4.3). Peer j will share his available resources to I_i competing peers for service s by solving:

$$max\sum_{i=1}^{n}\sum_{m=1}^{S}\left[(\frac{Req^{p}_{jI_im}}{Req^{p}_{I_ijm}})^{a}\times V_{jm}\times R^{p}_{jI_im}\right]\frac{x^{p}_{I_ijs}}{D^{p}_{I_ijs}},$$
$$\text{s.t}\ \sum_{i=1}^{n}x^{p}_{I_ijs}\leq C_{js}\text{ and }x^{p}_{I_ijs}\leq D^{p}_{I_ijs}\forall i, \tag{4.3}$$

where $Req^{p}_{jI_im}$ is the total number of requests for service m that peer j has sent to peer I_i till period p.

Compared to RA-MS, under ERA-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 4.5.7 demonstrate the benefits of ERA-MS compared to RA-MS in systems of peers with heterogeneous request generation profiles.

4.3.4 Reputation-based Server Selection Policy for MS systems (RS-MS)

In a similar way as in SS systems after the acquaintance duration, the probability with which a peer i directs his request to a peer j for a service s at a given period p is directly proportional to peer j's local reputation to i, for service s, as

$$p^{p}_{ijs}=R^{p}_{ijs}/\sum_{k\in S^{p}_{i}}R^{p}_{iks}, \tag{4.4}$$

where S^{p}_{i} is the set of all peer i's transacting peers till period p.

Service discovery issues are out of the scope of this dissertation but we rather consider that an underlying resource discovery mechanism, like NodeWiz [73] 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.

4.3.5 Application Scenarios

Our extended framework for multiple services systems can be used in p2p grids, like OurGrid [74], which support Bag-of-Tasks (BoT) applications. BoT [72] 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 [73]. In section 4.5, 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 [43, 75]. The cooperation between the existed wireless access points in a close proximity neighborhood could lead to the formation of wireless neighborhood communities [52]. 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 left for future work.

4.4 Performance Evaluation

4.4.1 Simulation Model

We have developed an event-driven simulator in C++ to evaluate the performance of our policies over 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. 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.

4.4.2 Comparison with other schemes

In order to exhibit the benefits of our proposed allocation framework for the multiple services case, we make comparison results with another related scheme. In the following we briefly describe it and discuss how it corresponds to our model.

Scheme in [2]

The scheme in [2], similarly with RA-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 [2] 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 RA-MS, and direct them to randomly chosen servers. Described scheme will be denoted as ENF in the figures of this chapter, standing for ExtNetFav.

It is important to note that our model under which ENF and RA-MS are compared in this chapter, is a more demanding model than the one considered in [2]. First, [2] 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 with RA-MS. Second, in [2] 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 reputation-based server selection policy. However, this is not the case for ENF, where servers are selected randomly. Third, [2] 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, [2] 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.

4.4.3 Performance Metrics

As for the SS systems, the goals of our proposed policies for MS systems 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. Towards this goal we define the global reputation and average satisfaction vectors.

The global reputation vector of a peer i at a period p is given by $\mathbf{GR}_i^p = \{GR_{i1}^p, ...GR_{iS}^p\}$, where $GR_{is}^p = \sum_{\forall j \neq i} R_{jis}^p / \ (N-1)$, i.e., the global reputation of peer i for service s and period p is taken by averaging the individual opinions of every other peer j in the system for peer i and service s (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 $\mathbf{St}_i^p = \{St_{i1}^p, ..., St_{iS}^p\}$ of each peer i in the system. The average satisfaction of peer i for service s until period p, St_{is}^p, is given by summing his satisfaction ratios in each one of his transactions with all other peers for service s and averaging over the total number of his requests for service s till period p.

Another goal of our policies for MS systems 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:

$$U_i^p = \sum_{s=1}^{S} \left[V_{is} \times \frac{RT_{is}^p - RF_{is}^p}{RT_{is}^p + RF_{is}^p} \right] / \sum_{s=1}^{S} V_{is} \tag{4.5}$$

where, RF_{is}^p and RT_{is}^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 $RT_{is}^p + RF_{is}^p$ 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 [2], 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.,

$$\begin{gathered} \sum_{i \in C} \sum_{s=1}^{S} (V_{is} \times (RT_{is} - RF_{is})) / |C| \\ - \sum_{j \in B} \sum_{s=1}^{S} (V_{js} \times RT_{js}) / |B| > 0 \Rightarrow \\ MI = \frac{\sum_{i \in C} \sum_{s=1}^{S} (V_{is} \times RF_{is})}{\sum_{i \in C} \sum_{s=1}^{S} (V_{is} \times RT_{is}) - \frac{|C|}{|B|} \sum_{j \in B} \sum_{s=1}^{S} (V_{js} \times RT_{js})} < 1 \end{gathered} \tag{4.6}$$

where C, B, are the set of collaborative and misbehaving peers respectively and MI stands for Misbehavior Indicator and expresses how well the system marginalizes 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.

Table 4.1: *Capabilities of each category peers.*

Category i	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
C_{i1}	9	8	7	6	5	4	3	2	1	0
$C_{i2}(\times 10^2)$	0	5	6	7	8	9	10	11	12	13

4.5 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 4.1 we see the capabilities of peers in each category i for service 1 and service 2 (C_{is}, $s = (1,2)$). 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' capabilities (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 3 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 4.2: *RA-MS Performance metrics of all categories peers for $V_i = \{1,1\}\forall i$.*

Cat. i	GR_{i1}	GR_{i2}	St_{i1}	St_{i2}	r_{i1}	r_{i2}	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

4.5.1 Homogeneous Service Value Profiles (HSP)

In Table 4.2 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_{i1}, r_{i2} 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_{i1}, r_{i2} 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 4.2 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_{11}. Moreover, the average utilities of all peers are positive, i.e, peers benefit by their transactions. In general, $St_{i2} > St_{i1}$; 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 4.3 we see the performance metrics when ENF scheme is applied, apart from global reputations which are not defined in this scheme (ENF uses the concept of *cost balance*). By comparing

Table 4.3: *ENF Performance metrics of all categories peers for* $V_i = \{1,1\} \forall i$.

Cat. i	St_{i1}	St_{i2}	r_{i1}	r_{i2}	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

the performance metrics in the 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, RA-MS seems to make a better allocation of the system resources.

In Fig.4.1 we further see the dynamics of RA-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 reputation-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 form coalitions which benefit their transactions.

4.5.2 Random Service Value Profiles (RSP)

In Table 4.4a we see the average satisfactions of all categories' peers for randomly chosen service values for RA-MS and ENF schemes and in Fig.4.2a and Fig.4.2b the utilities of all categories' peers as the system evolves in periods for RA-MS and ENF, respectively. For RA-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. Fig.4.2a 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

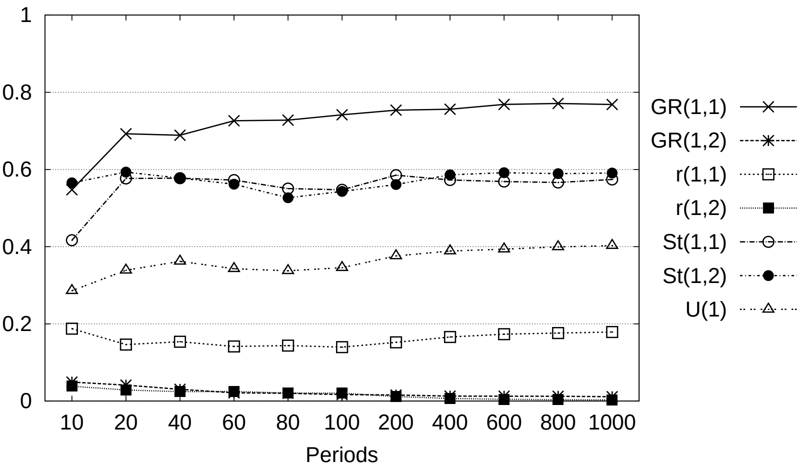


Figure 4.1: *Performance metrics for category 1 peers as the system evolves.*

put on the service that they are strong (which is quite unusual, as highlighted in section 4.2). From eq.(4.5) 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 RA-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.

4.5.3 Service Value Profiles Correlated With Capability Profiles (CSP)

Table 4.4b, Fig.4.2c and Fig.4.2d 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

Table 4.4: *Satisfactions for (a) RSP and (b) CSP for RA-MS and ENF policies.*

(a)

Cat. i	St_{i1} RA-MS	St_{i2} RA-MS	St_{i1} ENF	St_{i2} ENF	V_{i1}	V_{i2}
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)

Cat. i	$St_{i,1}$ RA-MS	$St_{i,2}$ RA-MS	$St_{i,1}$ ENF	$St_{i,2}$ ENF	$V_{i,1}$	$V_{i,2}$
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

we can see from the comparison of related figures that the utilities of all categories peers are highly 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.

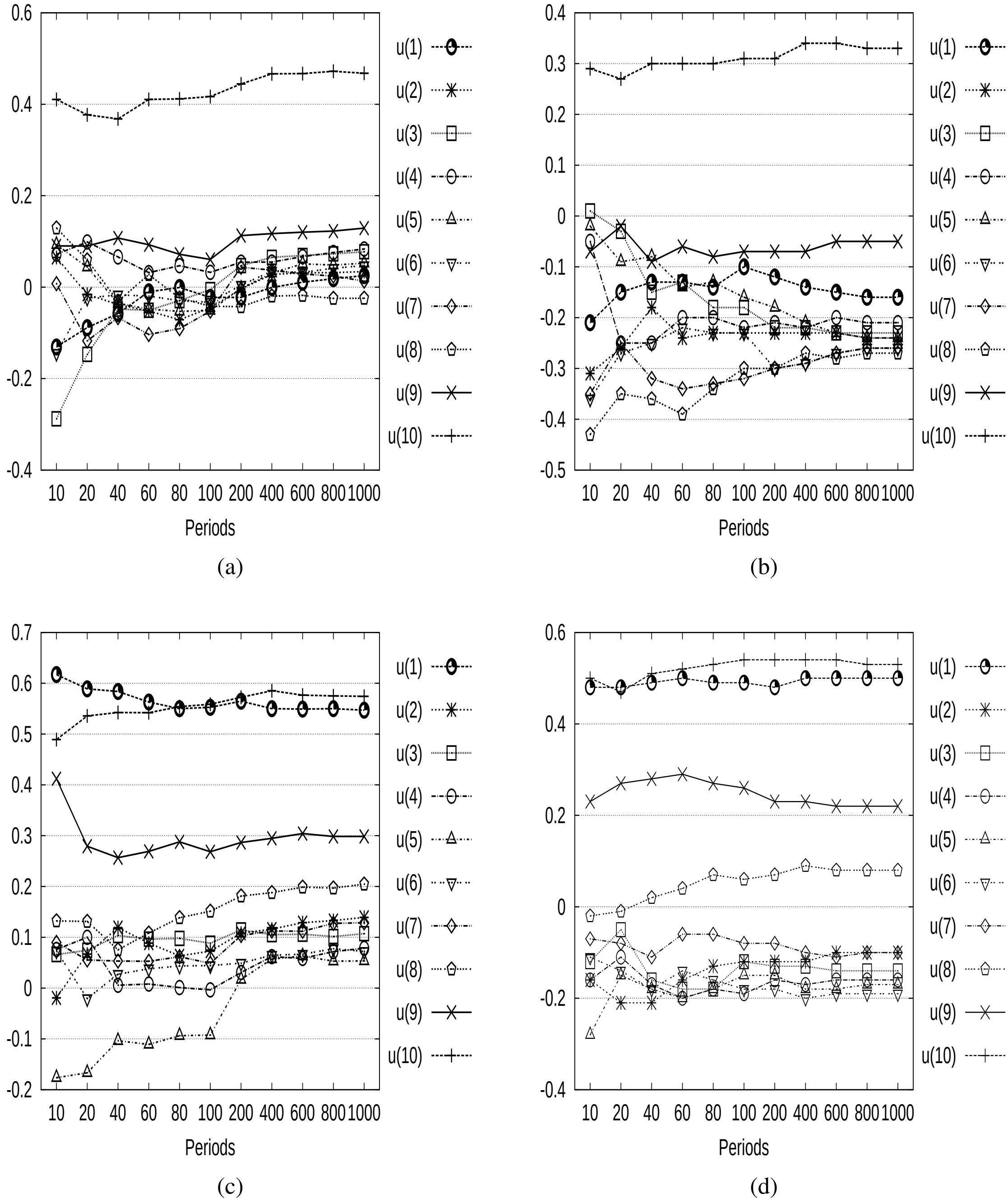


Figure 4.2: *Utilities of all categories peers for (a) RA-MS and RSP, (b) ENF and RSP, (c) RA-MS and CSP and (d) ENF and CSP.*

Table 4.5: *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

4.5.4 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 4.1, and used the same system parameters as we did for the population of 100 peers. In Table 4.5 we can see the utilities of all category peers after 1000 rounds both for the case of RSP examined in section 4.5.2 and the case of CSP examined in section 4.5.3 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 4.5.2, 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.

Table 4.6: *Dynamic service value profiles.*

Cat. i	V_{i1}	V_{i2}	V_{i1} 500^{th} prd	V_{i2} 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

4.5.5 Dynamic Service Value Profiles

In previous sections we showed that under RA-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 4.6. 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 4.6.

In Fig.4.3 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

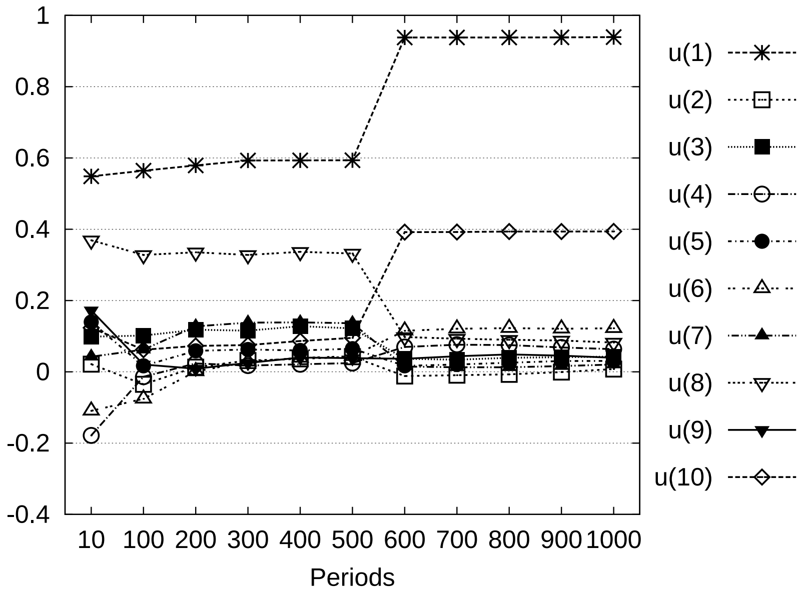


Figure 4.3: *Utilities of all categories peers for dynamic service value profiles.*

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.

4.5.6 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. That is, 100 peers are collaborative, splitting in each one of the 10 categories of table 4.1 and the rest 100 peers belong to category 0 and misbehave. For simplicity, all service values were set to 1. In Fig.4.4, 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. How-

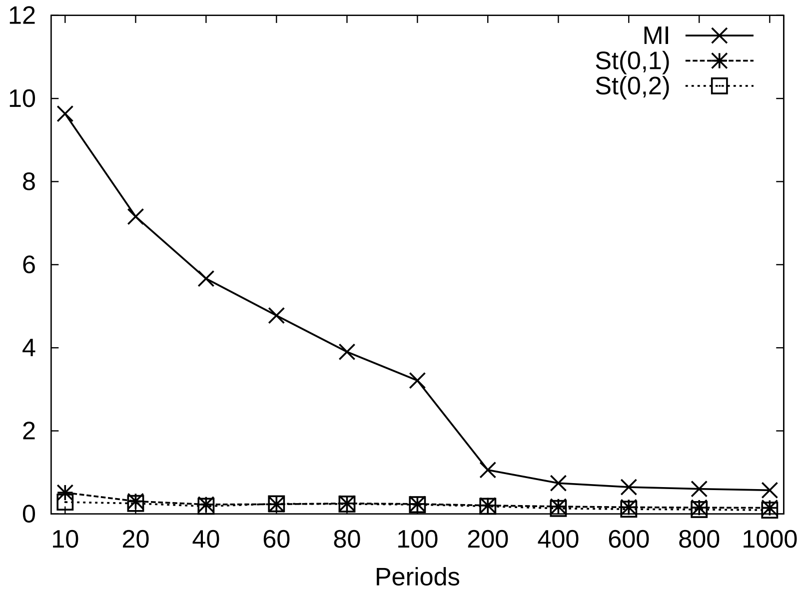


Figure 4.4: *Performance of misbehaving peers.*

ever, 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 satisfactions 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 [2] 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 [2] 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 [2], as we explain in detail in section 4.4.2.

Table 4.7: *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
RA-MS	7.83	3.57	14.6	6.74
ERA-MS (α=1)	5.93	5.40	11.4	9.96
ERA-MS (α=2)	5.72	5.66	10.3	10.5

4.5.7 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 4.7 we consider 100 peers, 50 % of which are low rate (LR) peers generating 2 requests per period and 50 % of which are higher rate peers (HR) generating 4 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, RA-MS and ERA-MS are used. When ENF or RA-MS is used, HR peers receive twice more resources on average per period than LR peers, while ERA-MS restricts the behavior of HR peers. For $a = 2$ both HR and LR peers receive almost the same amount of resources per period. ERA-MS provides a fairer treatment to peers, as equally cooperative peers receive almost the same amount of resources in a loaded network. ERA-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 RA-MS.

4.6 Summary

In this chapter we extended the reputation-based framework, presented in chapter 3, to fairly regulate the exchange of *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. Peers' utilities progressively improve, after the acquaintance period, irrespective of the peers' service values and capability profiles. This indicates the clever formation of coalitions between peers, who mutually benefit by exchanging different type of services. Peers form coalitions autonomously, following the proposed policies, without knowing one another's service capabilities and evaluation profiles. Moreover, cooperation is enforced, as more contributive peers can better satisfy their needs, while misbehavior is blocked.

Part of the work described in this chapter has been published in the following journal and conference proceedings:

* A.Satsiou, L.Tassiulas, A Trust-Based Exchange Framework for Multiple Services in P2P Systems, in Proc. of the 7th IEEE International Conference on P2P Computing, Galway, Ireland, 2-5 September 2007.

* A.Satsiou, L.Tassiulas, Trust-Based Exchange of Services to Motivate Cooperation in P2P Networks, accepted for publication in Peer-to-Peer Networking and Applications Journal, Springer.

Chapter 5

Reputation-based Framework for P2P *Single Capacity-Limited Link* (SC) Systems

5.1 Introduction

In this chapter we give the appropriate incentives for cooperation in systems of peers who use a single capacity-limited access link both for uploading and downloading content. For brevity's sake, we will refer to these kind of systems as *single capacity-limited link systems (SC)*. The main contributions of this chapter can be summarized as follows:

a) we propose an overall allocation framework according to which each peer independently decides how to allocate his available resources. On one hand, the reputation-based allocation policies described in chapter 3 are followed, based on which each peer determines the quality of service that he will offer to each one of his requesters according to their reputations and demands. On the other hand, a novel capacity adaptation algorithm and rational strategies are proposed, based on which peers, being aware of the reputation system's existence, dynamically adapt the capacity that they dedicate for uploading and downloading in order to improve their utility. The allocation scheme is combined with our reputation-based server selection policy, described in section 3.3.4 to help peers select among the more reputed/contributive servers and avoid misbehaving (non contributive) ones,

b) we evaluate our scheme via simulations to general setups of loaded file sharing networks even for cases of heterogeneous peers and multiple uploads/downloads, exploring the fairness and efficiency properties of our policies and demonstrating a much better performance than previous work in this area [34].

The remainder of this chapter is organized as follows: Section 5.2 describes the system model. In section 5.3 we propose a capacity allocation algorithm for rational peers along with three possible peers' request strategies and in section 5.4 we present the performance evaluation of our proposed policies in general setups of p2p networks. Finally section 5.5 concludes the chapter.

5.2 Model Description

We study a p2p network of N peers, who provide and consume bandwidth for content sharing. Peers act both as servers and clients simultaneously. We consider that the access technology does not provide strict separation between upstream and downstream flows; therefore peers have to share their link capacity between their uplink and downlink connections. Examples of such access technologies include WiFi, WiMAX, Ehternet LANs, etc. There are implementation tools [77] which can be used to appropriately adjust the uplink and downlink bandwidth for each connected user without modifications in the access protocol, but these issues are out of the scope of this chapter and are investigated for wireless networks in chapter 7.

We consider that each peer i is connected to the Backbone Network through an access link with capacity C_i (measured in b/s), reflecting his available bandwidth for uploading and downloading con-

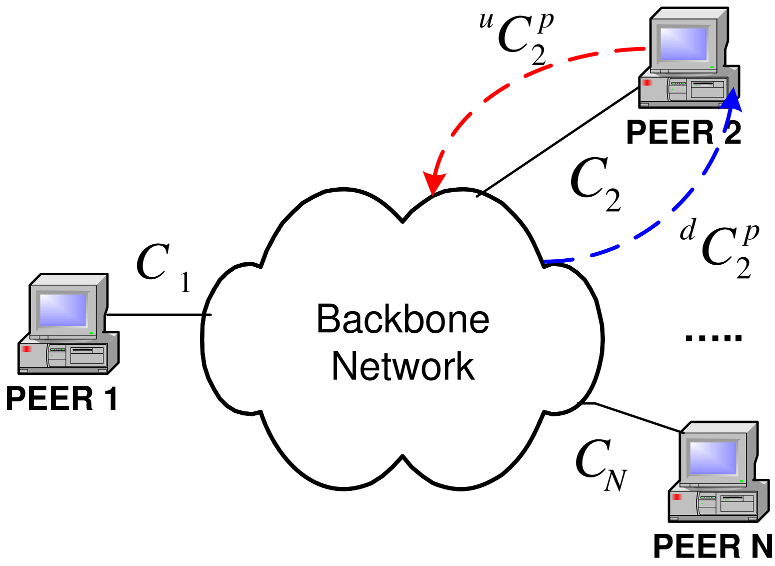


Figure 5.1: *Network Model for Single Capacity-limited Link Systems.*

tent. The capacity of the backbone network is regarded sufficient to accommodate all the traffic between the peers. The network model is depicted in Fig.5.1.

Our focus in this chapter is how to dynamically allocate in a distributed way the resources (bandwidth in this case) of the p2p community among their members in a way to provide fairness and efficiency in the system by guaranteeing that peers will be able to receive resources in proportion to their contributions, and that all available resources will be fully exploited. These issues are more meaningful in heavy loaded networks, where peers fight for resources. In light loaded networks, where the available resources exceed the needs of the peers, there is no obvious necessity for incentive mechanisms to force peers to contribute their resources, although in some cases peers may wish to free ride for other reasons (e.g., uploading media files is illegal in some countries). In the latter cases, our policies can be applied to ensure that peers will only be able to receive resources in proportion to their contributions, motivating them to cooperate at least as much as it is needed to satisfy their own needs. However, here we model an overloaded network as in [34], where there are always requests for downloading and content to be downloaded, i.e., a peer community where an infinite number of chunks of the same file are shared, where we can investigate the efficiency of our proposed policies in the worst possible conditions (overloaded networks). Content segmentation into chunks or content discovery issues are out of our scope, but we rather consider that chunks can be found by any other than the requester peer in the system.

We consider that our system progresses in periods of a fixed number of time units. Each peer decides about the capacity that he will dedicate in a given period p for uploading to others (we will refer to it as upload capacity in the rest of the chapter),${}^{u}C_i^p$, and the capacity that he will use for his own needs (we will refer to it as download capacity in the rest of the chapter), ${}^{d}C_i^p$. It is obvious that ${}^{u}C_i^p + {}^{d}C_i^p = C_i$ for any p and i. We model rational peers who aim at maximizing their utility (the amount of resources that they receive) by strategically adjusting their capacities. In the beginning of each period each peer i connects to g_i different servers simultaneously in order to receive chunks of the file. Variable g_i represents the request generation profile of each peer i; peers with higher g try to receive more resources than other peers, during a period p.

In each one of his connections peer i reports his demands (in terms of bandwidth) for the given request. The peers acting as servers will allocate their resources among competing peers according to their reputations and demands, following one of our allocation policies. The service is not granted for more than a period, in order to give the chance to other peers to access the link. Every new period, peers redirect their requests to the same or possibly other servers aiming at improving their received quality of service.

The combination of our proposed policies and strategies can form a resource allocation protocol incorporated in a client/server application software. The framework of the proposed protocol can be briefly described in the following steps for a random period p.

- **Step 1**: At the beginning of a period p, each peer i decides where to send his g_i requests (randomly or according to our reputation-based server selection policy described in section 3.3.4) and the amount of bandwidth he will request from each one of his servers under Basic, Greedy or Adaptive strategy described in the following section. During his first download period in the system, a peer sets ${}^{d}C_i^p = C_i$ and ${}^{u}C_i^p = 0$. This corresponds to either newcomers, i.e., peers who joined the system for the first time in order to download something or old peers who were idle for some time (not downloading) and decided to begin downloading again. We will refer to both newcomers and such old peers as “beginners”.

- **Step 2**: Each peer i allocates his current upload capacity ${}^{u}C_i^p$ over the peers’ requests that were directed to him. If the peer is a beginner, he does not serve any request (${}^{u}C_i^p = 0$). Allocation is performed through RA or ERA described in sections 3.3.2 and 3.3.3 respectively.

- **Step 3**: At the end of the period p each peer i calculates the reputation of his servers based on the service he received from them during this period and he adjusts the capacities ${}^{u}C_i^{p+1}$ and ${}^{d}C_i^{p+1}$ for the next period p+1, based on the capacity allocation strategy described in Algorithm 1.

Our model considers duplex connections, as it determines how a peer allocates his physical capacity among his downlink and uplink streams which take place simultaneously. A more sophisticated model could be built upon ours to further consider the bandwidth that is consumed by the acknowledgments for the TCP uplink and downlink traffic. However, since this is much smaller than the actual traffic, it is expected to have low impact on the performance of our policies.

For ease of reference, a list of notation used in this chapter is given in Table 5.1.

5.3 Rational Strategies

5.3.1 Capacity Allocation Strategy

We consider that each peer in the network seeks to maximize his utility, i.e., the amount of resources he receives from the network. Therefore, each “beginner” will initially dedicate his whole capacity

Table 5.1: *Notation and Variable Definitions*

Notation	Definition
N	Number of peers in the network
C_i	Link capacity of peer i
${}^{d}C_i^p$	Download capacity of peer i during period p
${}^{u}C_i^p$	Upload capacity of peer i during period p
g_i	Number of requests per period of peer i
b_i	Capacity step of peer i
R_{ij}^p	Reputation of peer j in the eyes of peer i during period p
GR_i^p	Global reputation of peer i during period p
D_{ij}^p	Bandwidth demands of peer i from peer j during period p
x_{ij}^p	Allocated bandwidth to peer i from peer j during period p
Req_{ij}^p	Total number of requests from peer i to peer j till period p
S_i^p	Set of transacting peers of peer i till period p
B_i^p	Total Bandwidth that peer i received during period p
a	Variable in ERA policy to control high rate peers

only for his own needs (downloads). Under the presence of the proposed reputation system in the network, though, the reputation of this peer will significantly fall, since he does not contribute any resources to the network, and this will affect his performance. The existence of the reputation system forces each peer to trade off his capacity among his upload streams, in order to increase his reputation and thus his revenue, and his download streams.

We model rational peers who try to maximize their utility with the least possible contributions, by progressively increasing their upload capacity till they obtain the quality of service they desire. Therefore, if the bandwidth a given peer i receives during a period p (B_i^p) is smaller than the capacity that he dedicates for downloading in the given period, it means that he needs to increase his cooperation level in order to improve his reputation and be able to receive more resources from the network. So, in this case peer i will increase his upload capacity by a constant b_i as soon as it does not exceed his total capacity and decrease the download capacity by the same constant b_i. On the other hand, if the peer receives as much bandwidth as his download capacity can afford, he will try to maximize his utility by further increasing his download capacity by the constant b_i for the next p+1 period, in cost of decreasing his upload capacity by the same constant b_i. From our simulations, we saw that a good trade-off between convergence rate and performance is succeeded in the system when each peer i uses a b_i in the order of $C_i/10$. Aforementioned algorithm is presented below.

Algorithm 1 Capacity Allocation Strategy for Peer i

if $(B_i^p < {}^dC_i^p)$ AND $({}^uC_i^p < C_i - b_i)$ **then**

$\quad {}^uC_i^{p+1} = {}^uC_i^p + b_i$

else if $(B_i^p == {}^dC_i^p)$ AND $({}^uC_i^p \geq b_i)$ **then**

$\quad {}^uC_i^{p+1} = {}^uC_i^p - b_i$

end if

${}^dC_i^{p+1} = C_i - {}^uC_i^{p+1}$

Someone could argue here that a peer i may receive less bandwidth than his download capacity in a period p, not because he is not cooperative enough but because of the existence of misbehaving (non contributive) peers in the system. Therefore, he should not further increase his cooperation level, i.e., his upload capacity. However, our reputation-based server selection policy refrains peers from requesting service from misbehaving peers. Moreover, the latter cannot take advantage of the increased cooperation level of high capacity peers, because of their low reputation, as we will see in the sequel.

As soon as peers decide about their download capacity for a given period p, they determine their demands from their servers according to one of the following strategies.

Basic Strategy (BS)

Under the basic strategy each peer i demands from each one of his servers in a given period p bandwidth equal to ${}^dC_i^p/g_i$. Since peers split their demands uniformly between servers and are unaware of the other peers' load, they may face the situation under which one of their servers cannot serve them and another one could give them more than what they actually asked. Since servers do not allocate more resources than the reported demands of the peers, peers may receive much less bandwidth than what they could in a given period. In order to confront this problematic case, we propose the greedy strategy described below.

Greedy Strategy (GS)

In the greedy strategy, each peer i requests from each one of his servers bandwidth equal to his total current download capacity. In this way he increases the possibility of receiving more bandwidth from an available and capable peer. If the upload capacity of a peer i is not fully exploited (because the demands of the competing peers are less than his available upload capacity), he uses the residual capacity for downloading. However, even then, the peer's available capacity for downloading may not be high enough to accommodate incoming flows, e.g., when more than one servers offer bandwidth equal to the requested one. We consider that as soon as the servers decide the portions of upload capacity they will give to each one of their requesters at a given period p, they will reserve these portions for their requesters' needs for the specific period. If their requesters cannot absorb these resources, they remain unexploited. Simulation results have shown that although the average bandwidth received by each peer is increased compared to the basic strategy, a lot of bandwidth which has been reserved for uploading is not utilized because it exceeds the download capacity of the requesters. If, however, peers release the upload capacity committed for the certain needs of some requesters when the latters' download capacity cannot sustain it and use it for other requesters or their own needs, they will make better exploitation of their resources. This is proposed in the adaptive strategy described below.

Adaptive Strategy (AS)

In the adaptive strategy as in the greedy strategy each peer i requests from each one of his servers bandwidth equal to his total current download capacity; however we consider that a server can be aware of the rate with which his requesters download data. Actually he can infer this information by the rate with which the acknowledgments of the TCP flows arrive to him. So, if a server sends with a rate of 6Mb/sec but the download capacity of peer i is only 4Mb/sec, then the server will receive the acknowledgments of the sent packets with a rate of 4Mb/sec. Therefore, he decreases his rate to peer i accordingly, and releases and spends the rest of his upload capacity to serve other peers by using the proposed allocation policies considering the actual needs of the competing peers, or if no other requesters exist he uses it for his own downloads. Ideally, we consider that one period is sufficiently long for all peers to adapt their download and upload rates according to the current supply and demand on all peers. This is an idealistic consideration; however we investigate this strategy in order to compare it with the others.

We also note here that in case of $g_i = 1$, all aforementioned strategies for peer i are equivalent.

5.4 Performance evaluation

5.4.1 Simulation Model

In order to evaluate the performance of our system, we simulate a p2p network of 100 peers. To bootstrap the system, all peers start with an initial small reputation of 0.07. 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 50 periods. Simulation time was set to 1000 periods. We consider that each peer i uses a constant b_i equal to 0.5 Mb/s, when performing the strategy of Algorithm 1.

We 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. Similarly, we investigated the system for different initial conditions for the upload and download capacities of all peers and saw that each peer's performance always reaches the same steady state which depends on the capacities of all peers in the system. The performance of the system scales even to larger overlay networks; however in larger networks the convergence time of the reputations and performance metrics increases.

We further note that for system of peers with homogeneous request generation profiles, RA policy performs almost identically with ERA. We say "almost" because when the reputation-based server selection policy is used, ERA slightly favors the most reputed peers, since most of the requests are directed there. For this reason, in sections 5.4.4-5.4.6 where we study peers with the same request generation profiles, we omit the performance of ERA policy.

In the following section we present some alternative schemes in order to highlight the benefits of our policies in comparison to them.

5.4.2 Alternative Schemes

No Strategy (NS)

Under this scheme peers do not follow any particular strategy in determining their request demands from the other peers. We consider that each peer i's demands vary randomly between [1Mb/s, C_i Mb/s]. Each peer uses his capacity to satisfy his own needs and offer the residual one (if any) to his requesters.

Simple Allocation Policy (SA)

In order to evaluate the performance of our reputation-based allocation policies we compare them with the simple policy, according to which a peer s allocates his upload capacity ${}^{u}C_s^p$ to his n requesters at a given period p, solely by considering their request demands.

According to the simple allocation policy peer s allocates his upload capacity by using a progressive filling algorithm and increasing all competing peers' bandwidth at the same rate of $1/n$ until one or several competing peers hit their limits (demands). Then he continues to increase the bandwidth of the remaining peers at the same rate as soon as all peers hit their limits or his upload capacity is fully utilized.

5.4.3 Performance Metrics

The average contribution level of a peer is reflected on his global reputation in the network as given by eq.(3.5). We further calculate the average received and offered bandwidth per request and per period for each peer, in order to evaluate the fairness and efficiency of our system. We consider that fairness is accomplished in the system when peers consume resources in proportion to their contributions, and efficiency is realized when all available resources are exploited in a loaded network.

5.4.4 Study of a Homogeneous System

In this section we consider a homogeneous system of peers who have the same capacity C and request generation rate g and they send their requests to each one of the other peers of the system with equal probability. In Fig.5.2 we can see the performance (average bandwidth per request) of different combination of strategies and policies for different capacity peers. Each peer has a request generation rate of 2 requests per period ($g = 2$). The BS/RA scenario stands for the combination of the basic request strategy (BS) and the reputation based allocation policy (RA). In the same way all possible scenarios are named.

As we can see from the results of Fig.5.2, the higher the capacity of all peers in the system, the higher the average bandwidth they receive per request. It is obvious that our proposed policies lead to the cooperation among peers and significantly improve the performance (average received

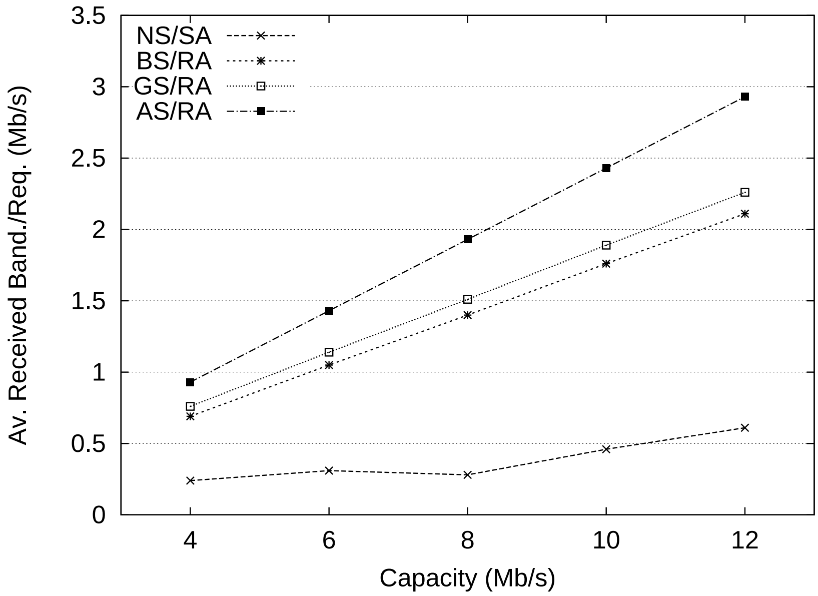


Figure 5.2: *Average received bandwidth per request for different capacity peers*

bandwidth per request) of the system compared to a system of peers who do not follow any strategies and reputation-based policies (e.g., NS/SA scheme). AS/RA scheme provides the best results in terms of average received bandwidth per request, followed by GS/RA and BS/RA.

Next, we would like to delve into the fairness issues of the different aforementioned schemes. We, thus, examine the average received and offered resources from peers and would like to see whether they are correlated. In Fig.5.3 we exhibit the average bandwidth received and provided per period for all schemes when the capacities of all peers are equal to 4 Mb/s and the request generation rate of each peer is 2 requests per period. What we see is that by using BS/RA or AS/RA schemes, the average bandwidth that peers receive and provide on average is almost the same, indicating fairness in a homogeneous system. However, with BS/RA scheme not all of the resources are exploited, while with AS/RA scheme, which adapts the upload and download rates according to the current supply and demand on peers, we see that almost all resources are exploited. The ideal case would be a centralized unit which would coordinate peers' requests among them in a way to balance the load in the network (each peer would have to serve the same number of requests every period). In our system, although

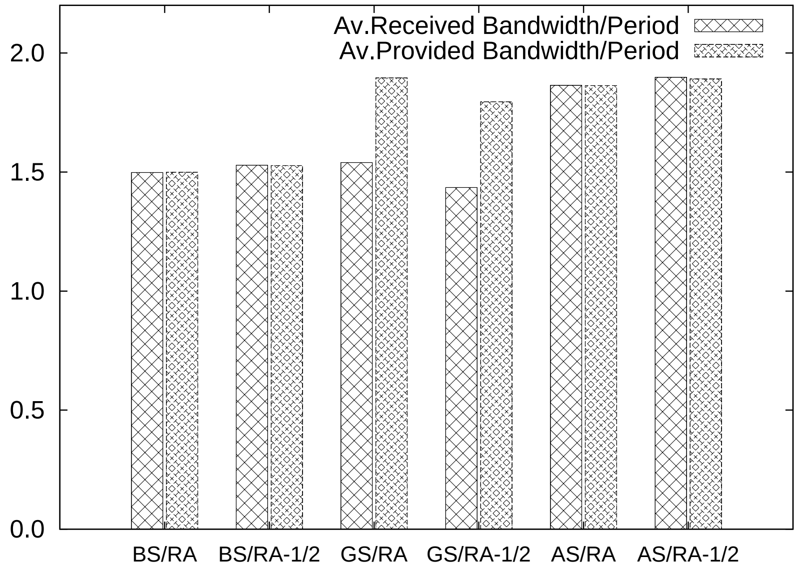


Figure 5.3: *Average received and provided bandwidth per period.*

all policies are distributed and peers are unaware of the load conditions on the others, peers succeed in reaching a cooperating point under which they provide as much resources as they offer and their performance is very close to the optimal one (exploitation of all resources) even with the BS/RA scheme. GS/RA scheme improves the quality of service received by peers compared to the BS/RA scheme, but many offered resources cannot be exploited, since they are dedicated for peers whose total capacity is currently occupied by other flows.

In Fig.5.3 we further see the performance of the schemes when peers, instead of using the rational dynamic capacity allocation scheme of Algorithm 1, divide their capacity into two halves; one for uploading and one for downloading (we use $^1\!/_2$ next to the name of the scheme to indicate these cases). The results reveal that the peers' performance is almost the same either for BS/RA or BS/RA-1/2 scheme. This indicates that rational users, under the presence of the reputation scheme, tend to equally share their capacity among their upload and download streams as in case they were imposed to do that (e.g., imposed by the software to use half of their capacity for uploading and half for downloading.), implying that rational users tend to cooperate. In Fig.5.4 we see the evolution of the

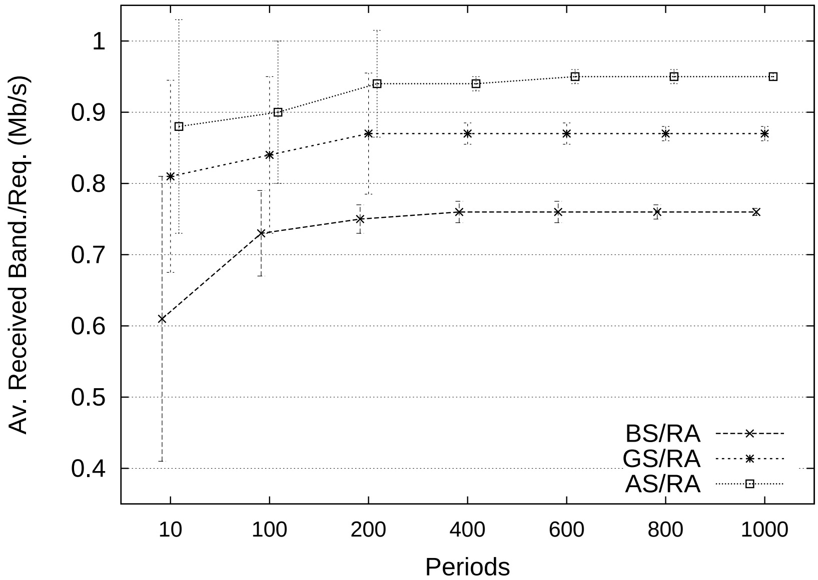


Figure 5.4: *Average received bandwidth per request through time.*

average received bandwidth per request through time for the aforementioned p2p system. In the early steps of the system, the average received bandwidth differentiates from peer to peer but as the system evolves, the average rate with which peers receive bandwidth converges to the same value for all the peers of the homogeneous system.

In Fig.5.5 we examine a smaller community of 10 peers and we see the average received bandwidth per request for different values of the request generation rate of all peers. For each generation rate that we study we consider such capacities of peers as to satisfy $C = 2g$ (Mb/s). In this way, each peer should receive 1Mb/sec on average per request in the optimal steady state. The average received bandwidth for the NS/SA scheme converges to zero. This was expected, as the higher the g, the bigger the sum of peer's demands and hence the less the capacity that is left for the needs of the other peers. All reputation based schemes' performance is slightly below the optimal one. This may be due to the heterogeneity of the load on servers. Some peers may have too many requests to serve in a given period and some may have none. The performance of peers is almost the optimal one in the case of $g = 9$ and $C = 18$Mb/s, under which each peer has to serve 9 other peers every period and thus the

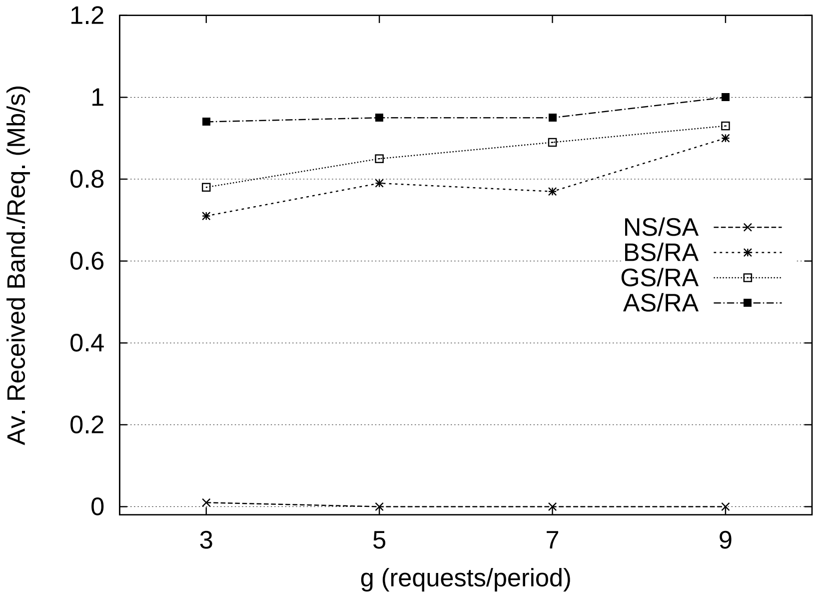


Figure 5.5: *Average received bandwidth per request for different couples of C and g.*

load is stable and the same for all peers.

We note that the same conclusions apply for any homogeneous system (e.g., other cases of capacities, request generation profiles and populations of peers) but here we just exhibit some indicative results.

5.4.5 Heterogeneous System

Here, we investigate a network of cooperative peers with different capabilities/link capacities. Some of the peers are more powerful than others. In particular we study a network of 100 peers, where 20% of them have a total capacity of 8Mb/s, 20% have one of 7Mb/s, 20% have one of 6Mb/s, 20% have one of 5Mb/s and 20% have one of 4Mb/s. All peers have a request generation profile of 2 requests per period. In Fig.5.6 and Fig.5.7 we can see the average received bandwidth per request and global reputations of each category peers for the various schemes, respectively.

It is remarkable that the global reputations of peers are proportional to their capacities. Moreover, the average received bandwidth per request by each category of peers is analogous to the global rep-

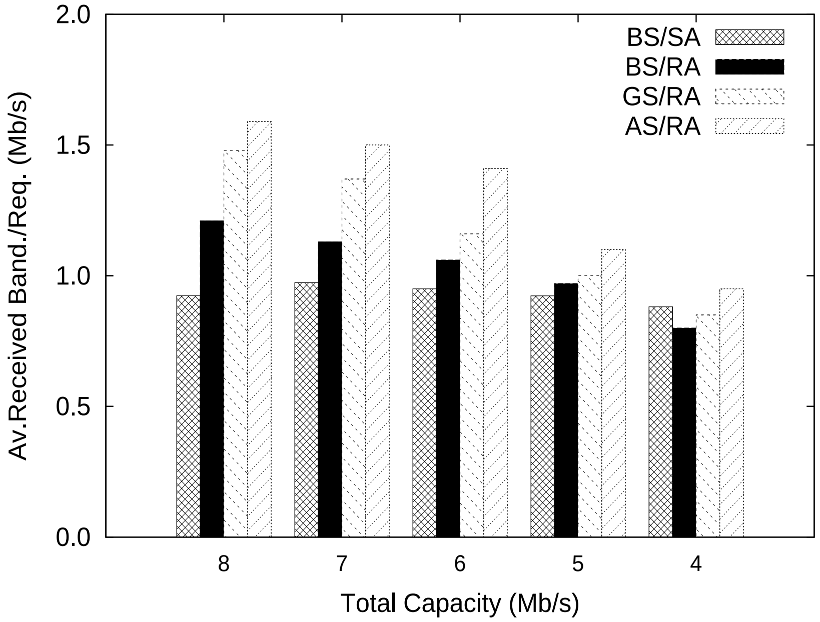


Figure 5.6: *Performance of different capacity peers in the network.*

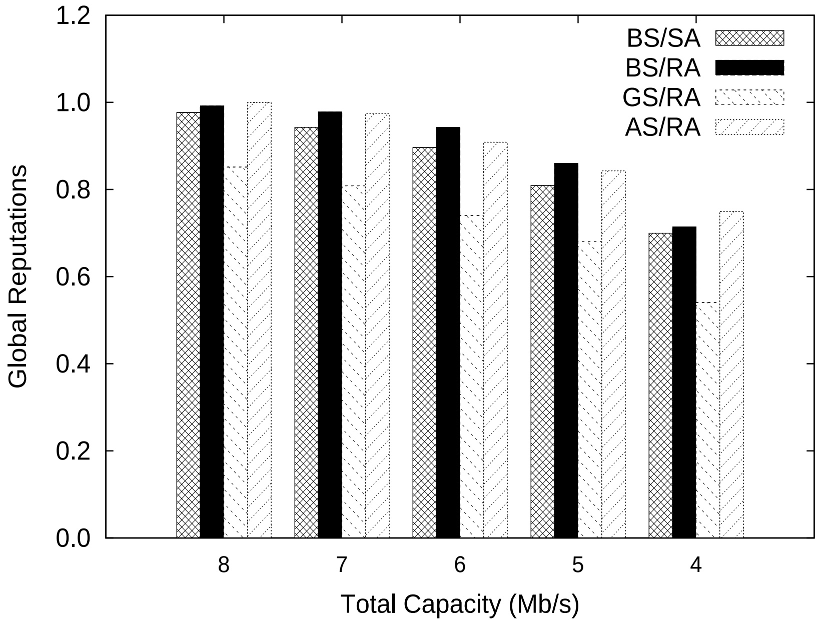


Figure 5.7: *Global reputation of different capacity peers in the network.*

utation of the category. This indicates that reputation on one hand correctly captures the capabilities (capacities) of the different peers and on the other hand the reputation system performs satisfactorily, as each peer receives bandwidth in proportion to his reputation.

From Fig.5.6 we see that by using the reputation-based schemes the higher capacity peers succeed in receiving more resources from the network as they are the ones who contribute more resources in it. Actually for BS/RA and AS/RA schemes, peers' average received and provided bandwidth is the same in the steady state (this is not exhibited in Fig.5.6). Higher capacity peers' resources are not exploited in this case study because of the inability of the other peers to offer and receive resources at the rate that high capacity ones can accept and provide respectively. If more high capacity peers existed in the heterogeneous network, their performance could be improved and the unexploited capacity would be decreased (see in the sequel the discussion on Fig.5.8 and Fig.5.9). Under BS/SA scheme all peers receive almost the same bandwidth irrespective of their contributions, as no reputation system is regarded.

Finally, we see that the performance of our reputation-based schemes outperforms the one of [34] according to which no peer can receive more than $\min_{i \in N}\{C_i\}/2$ bandwidth under a given period, and so the slowest link becomes the bottleneck of the system. This is the equilibrium reached by work in [34] for heterogeneous systems of peers who can perform at most one download and one upload at a given period. The authors claim that this inefficiency could be eliminated by multiple uploads/downloads but there is no obvious way how to extend their scheme for multiple uploads/downloads and this is part of their ongoing work. As we see from Fig.5.6 the reputation-based schemes succeed in overcoming this limit $\min_{i \in N}\{C_i\}/2 = 4/2 = 2$ Mb/s per period or 1 Mb/s per request and the performance of all capacity peers depends on their contributions. We also experimented with different peer populations, capacities and request generation profiles and saw that the reputation-based schemes always overcome this limit.

The main conclusions of our experiments are that under the reputation-based schemes peers provide and receive resources in proportion to their capabilities. So, higher capacity peers offer and receive resources in higher rates than lower capacity ones. According to the proportion of high and low capacity peers in the system, their performance improves or deteriorates and affects the capacity that remains unexploited. As an illustration example we exhibit Fig.5.8 and Fig.5.9. We consider different portions of the 8Mb/s capacity peers in the aforementioned peer system. In each scenario that we examine, lower capacity peers are progressively replaced by 8Mb/s capacity peers. The case 20% corresponds to the scenario that we examined in this section; the case 40% indicates that the 20% 4Mb/s capacity peers are replaced by 8Mb/s capacity peers; the case 60% corresponds to the case that 5Mb/s capacity peers are also replaced by 8Mb/s peers and so on. What we observe is that as the percentage of the 8Mb/s capacity peers increase in the system, their performance is improved (Fig.5.9) and their capacity that remains unexploited decreases (Fig.5.8).

If we further use the RS server selection policy described in section 3.3.4, the exhibited results of this section are slightly better. The higher the difference between the capacities of the peers in the heterogeneous system, the more effective the RS policy is for all capacity peers, as it helps them choose among the more appropriate traders. One such example is demonstrated in Fig.5.10 where we consider various scenarios of different capacity peers in the system. The case 8-0 corresponds to a system consisting of 50% peers of 8Mb/s capacity and 50% of misbehaving peers (contributing 0 resources); the case 8-2 corresponds to a system of 50% peers with 8Mb/s capacity and 50% peers with 2Mb/s capacity and so on. What we can see from this figure is that the higher the capacity difference between the peers of the system (e.g., case 8-0), the greater the improvement of the 8Mb/s peers' performance, when the RS policy is used.

5.4.6 New-comers

In this section we exhibit the behavior of the system under the case of new peers periodically entering and leaving the network. As in the previous section, we consider a network of 100 peers where peers are categorized according to their total physical capacity; there are 5 equal-sized categories of peers with 8,7,6,5 and 4Mb/s capacities. However, we consider that 50% of each category peers are

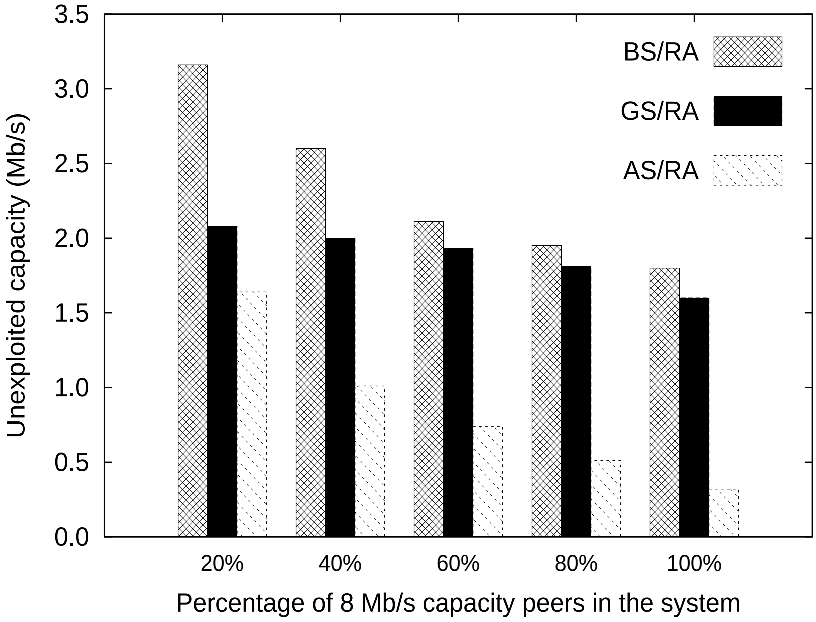


Figure 5.8: *Unexploited capacity of 8Mb/s capacity peers for different portions of capacity peers.*

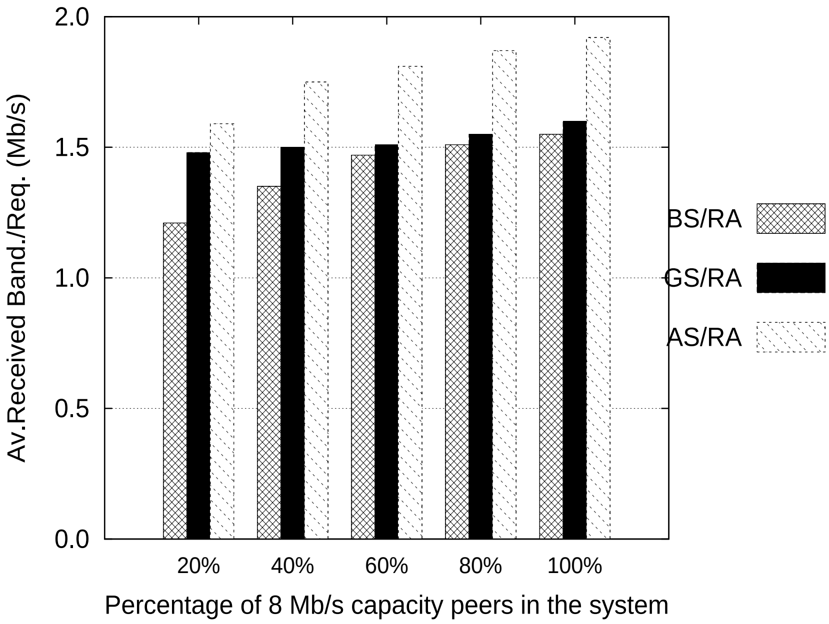


Figure 5.9: *Performance of 8Mb/s capacity peers for different portions of capacity peers.*

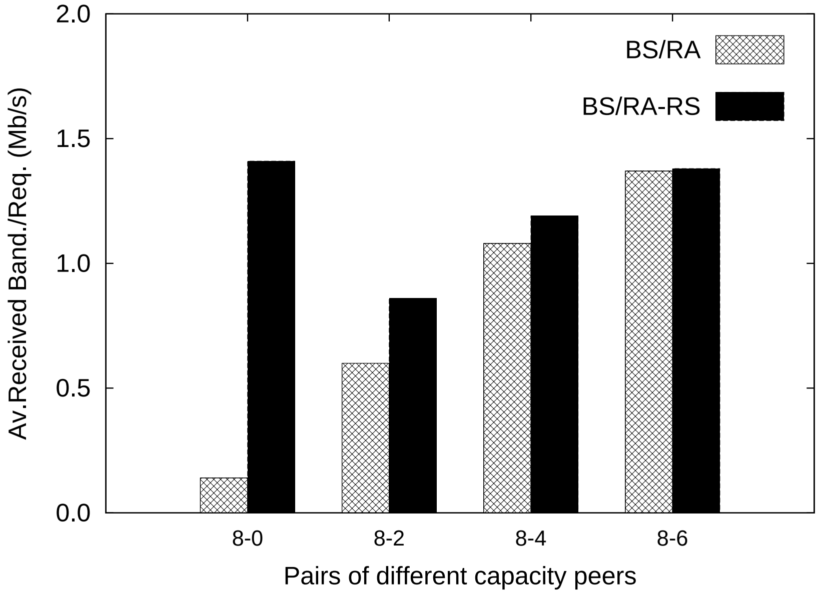


Figure 5.10: *Performance of 8Mb/s capacity peers coexisted with different capacity peers.*

permanent in the system (stable), while the remaining 50% of each category (capacity) peers stay in the network only for 100 periods and are replaced by new identity peers of the same capacity with those who left every 100 periods. We use this case in order to keep 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.

In Fig.5.11 we can see the performance of the stable and new-comers of 8Mb/s and 4Mb/s capacity peers, when the BS/RA-RS scenario is used. The conclusions from the performance of the other capacity peers and when the GS/RA-RS or AS/RA-RS strategies are used, remain the same. What we see from Fig.5.11 is that the performance of the stable peers is not much influenced by the presence of the new-comers, and on the other hand, new-comers' behavior is very near to the one of the stable peers of the same capacity, despite their short life in the network. This conclusion, which is also supported by many different scenarios that we run, shows that new-comers adapt quickly to the network and do not really affect the performance of the long-lived peers in the system.

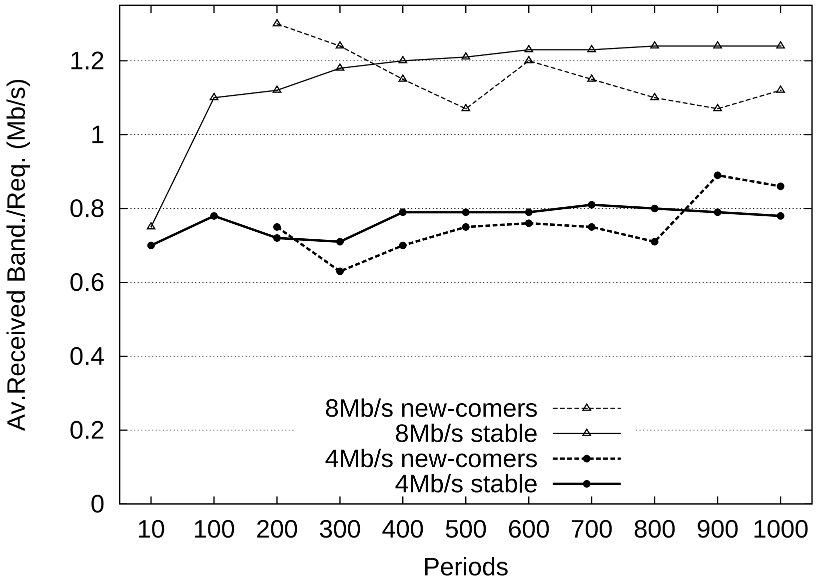


Figure 5.11: *Performance of stable peers and new-comers.*

5.4.7 Peers with different request generation profiles

In this section, we study a network of 4Mb/s capacity peers who generate requests with different rates. In particular we consider a network of 100 peers, 50% of which are high rate (HR) peers generating 4 requests per period and 50 % of which are lower rate peers (LR) generating 2 requests per period. In Table 5.2 we demonstrate the average received bandwidth per request and per period when different strategies and allocation policies are used.

We can see that in general the average received bandwidth per request is lower for the high rate peers than for the lower rate peers for all the schemes. This is due to the fact that high rate peers reserve less capacity than the lower capacity ones for uploading in order to free more space for their high number of download requests. In this way, their reputations and thus their performance fall. However, due to their large number of requests they succeed in receiving more bandwidth per period on average than the lower rate ones when BS/RA and AS/RA schemes are used, although all peers' capabilities/capacities are the same. In this way some peers may wish to exploit the network by generating much more requests than others without being able to provide the analogous resources to

Table 5.2: *Performance of HR and LR peers*

HR Capacity : $g = 4$		***Av. Band. / Req.***		***Av. Band. / Period***	
LR Capacity : $g = 2$		***HR***	***LR***	***HR***	***LR***
BS	***RA***	0.49	0.55	1.95	1.10
	ERA ($a = 1$)	0.39	0.74	1.58	1.47
AS	***RA***	0.61	0.78	2.44	1.56
	ERA ($a = 1$)	0.52	0.97	2.08	1.94

the network. In order to protect the system from such a selfish behavior, we use the ERA policy, which restricts the excess requests of high rate peers. As we can see from Table 5.2, when the ERA policy is used for $a = 1$, both high and low rate peers receive on average almost the same amount of resources per period. Since they all have the same capabilities/capacities of 4Mb/s, they should all be able to offer and receive resources in the same average rate during a period, irrespective of the number of requests they generate; this is accomplished by ERA ($a = 1$) policy which balances the performance of high and low rate peers, when combined with any of the BS or AS strategies.

5.5 Summary

In this chapter we presented a complete methodology for motivating peers to contribute their resources in a loaded network where the capacity-limited access link of each peer is shared among his download and upload streams. Our proposed allocation scheme can be implemented in a distributed manner at each peer, without the necessity of any global information. We considered rational peers who seek to maximize their utility with the least possible contributions and showed that under the presence of our proposed reputation system they are inclined to cooperation. In the homogeneous scenario where all peers have the same capacity and request generation rate, rational peers fast reach a cooperative operating point under which they offer half of their capacity for their download and half for their upload streams. In heterogeneous systems, utilization of all resources is not always possible since some of the higher capacity (powerful) peers' resources may not be exploited because of the inability of other peers to offer and receive resources at the rate that powerful peers can accept and provide respectively. However, fairness is still guaranteed as peers receive resources in proportion to their contributions. Proposed reputation-based allocation protocol significantly improves the performance of peers under the related work in [34] over heterogeneous systems and even in more general systems where parallel downloads and uploads take place and peers with different request generation profiles exist.

Part of the work described in this chapter has been published in the following journal:

* A.Satsiou, L.Tassiulas, Reputation-based Resource Allocation in P2P Systems of Rational Users, IEEE Transactions on Parallel and Distributed Systems, 21 (4), pp 466-479, April 2010.

Chapter 6

Reputation-based Framework over *BitTorrent-like (BT)* systems

In this chapter we apply our reputation-based framework in a BitTorrent-like file sharing system and we highlight the potential performance gains both for ADSL-type and single capacity-limited link systems.

The main contributions of this chapter can be summarized as follows:

a) we present BitTorrent and its operation in detail,

b) we delve into BitTorrent's limitations and explain how they could be transcended by our reputation-based scheme,

c) we propose two extension algorithms for BitTorrent, in order to improve its performance in single capacity-limited link systems and,

b) we simulate BitTorrent-like setups and present how our reputation-based framework outperform BT's reciprocation mechanism both in p2p systems with ADSL-type connections and in single capacity-limited link systems. More specifically, we present how our policies can shield

BT's performance from free riders, preserve fairness and provide incentives to seeds to remain and contribute in the system

The remainder of this chapter is organized as follows. Section 6.1 presents BitTorrent and its operation while section highlights the limitations of BT that can be transcended by our reputation-based scheme. In section 6.3 we present the performance evaluation and performance gains over BitTorrent both for the ADSL-type systems and the single capacity-limited link systems. Finally, section 6.4 concludes the chapter.

6.1 Overview of BitTorrent

BitTorrent (BT) is a popular peer-to-peer content distribution protocol, designed for bulk data transfer. It enables scalable and efficient content replication by exploiting the upload capacity of the downloading peers. The basic idea in BitTorrent is to divide a single large file into equal-sized pieces and deliver them in a non-linear way. The set of peers attempting to download the file do so by connecting to several other peers simultaneously and download in parallel different pieces of the file from multiple peers.

6.1.1 Terminology

Although there is no standardized terminology for BitTorrent, we provide in this section the terms which are widely used by the majority of the BT community and we further use in this dissertation.

- **Torrent**: A small file which contains metadata about one or more files, including their names, sizes and SHA-1 hash values of all pieces in the files to verify their integrity. It also contains the IP address and port number of a tracker that coordinates communication between the peers that download the specified files. The *torrent* is downloaded from a web site.

We also note that the term *torrent* is often used to refer to all files it describes and the set of peers that cooperate to download them.

- **Tracker**: The *tracker* is a centralized entity that maintains a list of all peers currently participating in the download of the same content and maintains statistics. Peers periodically (typically every 30 minutes) report their state to the *tracker* as well as the amount of uploaded and downloaded bytes. The tracker is responsible to return upon request a list of typically 50 peers to whom the requester could connect to download the desired file.

- **Leecher and Seed**: A BT peer can be either a *leecher*, i.e., a peer who does not have the entire content and still downloads pieces of it, or a *seed*, i.e., a peer who has the entire content and may upload it to others.

- **Swarm**: A *swarm* is the set of all peers (tracker, leechers and seeds) who participate in the download of the same content using the BitTorrent protocol.

- **Choked/Unchoked**: An *Unchoked* peer refers to a peer who was selected for uploading content, while a *choked* peer refers to one who was denied service.

- **Peer Set**: The peer set of a given peer contains all the peers he knows, also known as his *neighbors*.

- **Active Set**: The active set of a given peer contains all his *unchoked neighbors.*

6.1.2 BitTorrent Operation

A file in BitTorrent (BT) is divided in equal-sized pieces, and each piece is further divided in multiple blocks (typically 32-256 KB) to enable pipelining of requests and mask the request-response latency. A torrent is created by the content provider to include all the necessary information for the file. Each peer *P* that wants to download a file, accesses well-known websites which act as global directories of available files and downloads the torrent of interest. He then contacts the associated tracker, reported in the torrent, who is responsible to return the IP addresses/Ports of a random set of typically 50 peers currently transferring pieces of the file(s) specified in the torrent. This set forms the initial peer set of *P*, which can be later enhanced by new peers connecting directly to him. Peer *P* then attempts to establish connections to 20-40 of his neighbors. Whenever the number of his neighbors fall below 20, e.g., due to departure of peers, the node contacts the tracker to obtain a list of additional peers.

As soon as peer *P* connects to his neighbors, he exchanges *bitfield* messages with them, which contain information about the pieces each peer has. In this way, peers can see whether they are interested in pieces of each other. Moreover, peers send *have* messages to all of their neighbors in order to inform them about a new piece that they downloaded. Based on this information peers use a local rarest first (LFR) policy to decide which block to download. This policy is also known as local rarest-first policy since the block decision is based on local information at each peer. According to this policy, a peer seeks to download a block that is least replicated among his neighbors in order to maximize the diversity of content in the system. Further more, once a single block has been requested, the remaining blocks from that particular piece are also requested before blocks from any other piece. In this way, complete pieces are obtained as quickly as possible.

An exception to the local rarest first policy is in the beginning of the download. At that time, the peer has no piece to upload, so it's important to get a complete piece as quickly as possible in order to bootstrap himself. Rare pieces are hard to find, so they would be downloaded slower than pieces which are present on multiple peers for which it's possible to download blocks from different sources. For this reason, new peers who start downloading, select a block at random until the first complete piece is assembled, and then the strategy switches to the local rarest first.

Another exception to the local rarest first policy is in the end of the download and is called *end game mode*, a policy to make the end of the down load faster. Once a peer has requested all blocks, either received already or not, he starts the *end game mode* by requesting all not yet received blocks from all the peers in his peer set. Each time he receives a block, he cancels all the remaining requests for the specific block. End game mode has little impact on the overall performance of BitTorrent, since it is only used at the very end of the download.

In order to confront free-riding effect, BitTorrent uses a so called choking algorithm, which is actually a *tit-for-tat (TFT) strategy*, to determine which neighbors to upload data. Each peer executes the choking algorithm every ten seconds, according to which all its neighbors are ranked based on their upload rate during the last 20s, and only the first a top peers are unchoked. BitTorrent further uses an optimistic unchoke policy every thirty seconds, wherein each peer unchokes r randomly chosen peers, regardless of their rank and keep them unchoked for thirty seconds. The set of the unchoked peers is called the active set and its size is always limited to $a+r$. Therefore, each time a peer optimistically unchokes r peers, he chokes r other peers from his active set with the worst upload rates, in order to keep the size of the active set constant. A peer uses optimistic unchokes, in order to discover neighbors that might offer higher download rates than the peers it is currently downloading from, and give the chance to new peers to download their first block and prove their collaboration by reciprocating. In the basic version of BitTorrent and the most popular BitTorrent client, Azureus, the default active set size a is equal to four and r is equal to one. The upload bandwidth of a peer is allocated equally among all unchoked peers and is referred to as *equal-split*. The choking policy is not only performed every 10 seconds, but also each time an unchoked and interested peer leaves the peer set, or whenever an unchoked peer switches its interest state.

As far as seeds are concerned, since they do not download any pieces, they follow a different choking strategy. According to most implementations, seeds unchoke the fastest downloaders irrespective of their contributions, in order to utilize the seed capacity in the best possible way and maximize the efficiency of the content dissemination. BT version 4.0.0 introduces a modified seed choking policy which favors those peers who were most recently unchoked in order to serve as many peers as possible.

6.2 Limitations of BitTorrent transcended by our reputation-based scheme

Apart from work in [24] and [42] which present several shortcomings, as we already described in section 2.4, there is no other work to propose a reputation-based allocation policy for BitTorrent. The main strength of an appropriate reputation system over BitTorrent would first of all give the appropriate incentives to peers who already downloaded their files (seeds) to remain in the system and contribute their resources to others, in order to gain reputation for their future requests for other files. The presence of seeds in the system have proved to be a crucial factor in the success of BT [40] and so far no previous work has successfully provided incentives for seeds. Furthermore, the gains of such a reputation system should not only affect the long term interactions among peers but also the short term. Peers should rather select whom and how to serve based on competing peers' reputations, instead of performing optimistic unchokes, which are considered to be one of the main reasons of unfairness in BT [36, 41].

Seeds should also consider the reputation of competing peers in their allocation decisions, shielding BT from strategic peers. Currently, seeds upload to the fastest downloaders, leaving much open space for various exploits [36, 62]. With a reputation system applied, seeds can provide service differentiation to their requesters according to their reputations and probabilities of reciprocation in future requests; if, for example, they both participate in related torrents (e.g., Disney comics). Under this way, seeds are strongly motivated to contribute their resources.

Our local reputation system would be very effective in BT communities of users who share common interests and have to subscribe in order to participate, like TvTorrents.COM which is all about TV shows. BT communities have been studied by [11]. It is shown that although a sharing ratio enforcement policy is used in many BT communities, free riding is still present. Such communities prevent peers to gain access to new content if their sharing ratio falls below a certain threshold. However, the sharing ratio of a peer is evaluated based on reports from the respective peer; thus related software could be hacked to report fake uploading and downloading ratios. With our proposed local reputation mechanism, on the other hand, BT communities could guarantee that all participating peers cooperate and provide the right content without the need of a centralized control. Furthermore, peers of the same community will more likely transact between them quite often, creating a sufficient opinion for one another (local reputations) in order to make their allocation decisions.

What's more, our reputation system can well apply in multi torrent collaborations [35] which have been proved very promising in the overall BT performance. Peers can participate in multiple file downloads and share their content based on the reputation metrics of the participants, which actually reflect their general contribution levels and not just per torrent contributions. In this way, peers could exchange blocks from different files and exchange decisions could be based on reputation metrics.

Apart from introducing an appropriate reputation system for BitTorrent, our work also notes that all current BT implementations and proposed improvements over it assume a clear distinction between the upload and the download capacity of the peers. More specifically, the download capacity is much bigger than the upload capacity, as in ADSL connections, while the upload capacity is considered the bottleneck in the connection.

If BitTorrent peers had to share a limited cumulative capacity both for uploading and downloading data, as is the case in access technologies like WiFi, WiMAX and Ethernet LANS, they should decide how to allocate their available physical capacity between their uplink and downlink connections. Current implementations of BitTorrent do not support a capacity adaptation algorithm to help peers dynamically decide about their download and upload capacity. The problem with that is that if all peers acted rationally, they would rather use their whole capacity solely for downloading and the system would collapse. Our capacity adaptation algorithm along with our reputation-based framework can significantly improve the performance of peers in such BT setups as we will see in the sequel.

6.3 Performance Evaluation

6.3.1 Simulating the BitTorrent-setting

We simulate the already described basic version of BitTorrent [60] and examine the system as soon as a neighborhood of connected peers using the same torrent is set. It is reported from measurements of BitTorrent systems that although there are several large swarms, the majority of the swarms are very small and the average population of the swarms is only about 102 peers [35]. Here, we consider a neighborhood of 50 connected peers, in compliance with the typical set of peers returned by the tracker. All connected peers send requests for blocks of the file to one another. Since our focus is in the evaluation of the allocation policy, we omit the block selection algorithm used in BitTorrent and consider that the connected peers are always interested in blocks of each other. The system runs in periods, where each period lasts for ten seconds. Each peer performs the choked algorithm every 10 seconds, and every 30 seconds it performs the optimistic unchoke algorithm, just as in the basic version of BitTorrent.

In the following sections we provide comparison results of BitTorrent reciprocation strategy with our proposed policies for systems of peers with ADSL-like connections, described in chapter 3 and for single capacity-limited link systems, described in chapter 5.

6.3.2 Apllying RA and ERA to BT peers with ADSL-type connections

In this section we compare the performance of our policies, RA and ERA, with BitTorrent reciprocation mechanism for systems of peers with ADSL-type connections. The default values used for the initial reputations and misbehavior threshold are as defined in chapter 3. 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 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, $D^t_{ij}, \forall i, j, t$ are set equal to 1 in the reputation calculation formula (3.1), 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 a server 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 [63]. Optimistic unchoke policy is considered to be one of the reasons for that [36]. To see how our policies can overcome this deficiency of BT, we consider a scenario of 40% misbehaving peers, 20% peers who dedicate 7Mb/s for uploading, 20% 6Mb/s and 20% 5Mb/s. In Fig.6.1 we can see that when our policies are used (RA and ERA 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, peers contribute on average as much resources as they obtain per period. Please note that the performance of ERA is close to RA, 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 reputation-based server selection scheme. However, under ERA, peers do not have any motivation to hamper the software and generate more requests than the others, since they know that their performance will be restricted.

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 [35, 56]. Here, we show how our proposed scheme can motivate peers who have already downloaded a file to remain and contribute in the system. We study 50 connected peers with physical capacities of 7Mb/s who download blocks of files. One of these

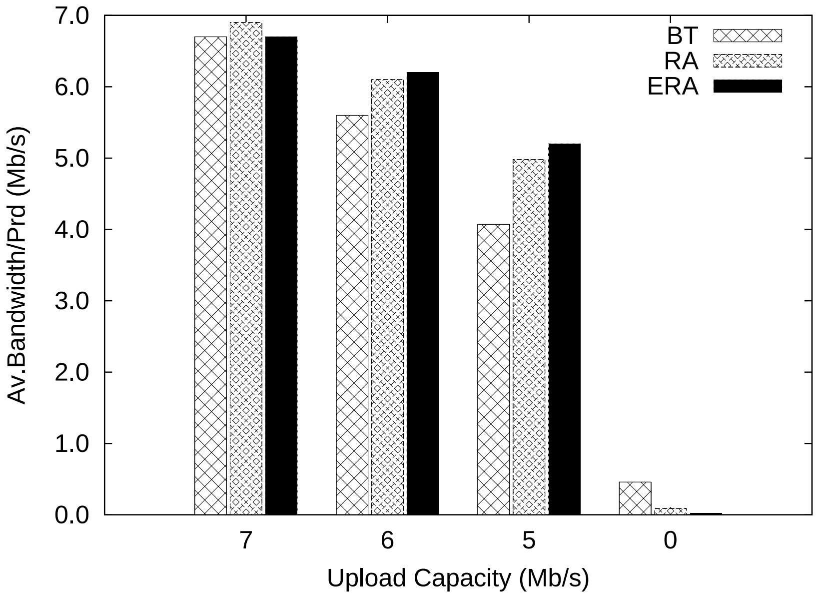


Figure 6.1: *Performance of misbehaving peers in a BT scenario.*

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.6.2 we see the performance of the seed before and after his uploading 100 periods. With BT and RA 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 ERA 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 to other peers, the number of requests directed to him compared to the ones he has sent to others significantly increases; thus from eq.(3.3) it is obvious that he is going to be given high priority in his future requests.

We note, however, that Fig.6.2 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

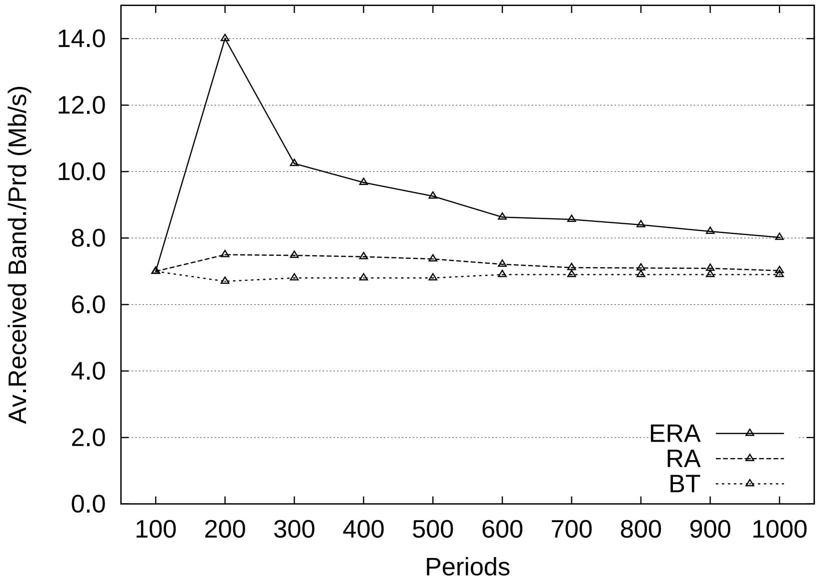


Figure 6.2: *Performance of seed.*

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.

6.3.3 Applying RA and ERA to BT peers with a single capacity-limited access link

Limitations of BitTorrent-like systems and proposed extensions for SC systems

The work related to BitTorrent and several improvements over it [25, 31, 64] assumes a clear distinction between the upload and the download capacity of the peers. The download capacity is much bigger than the upload capacity (as in ADSL connections), which is considered to be the bottleneck in the connections.

If BT peers had to share a limited single-link capacity both for uploading and downloading data, they should decide how to allocate it between their uplink and downlink connections. Current implementations of BT do not support a capacity adaptation algorithm to help peers dynamically alter their

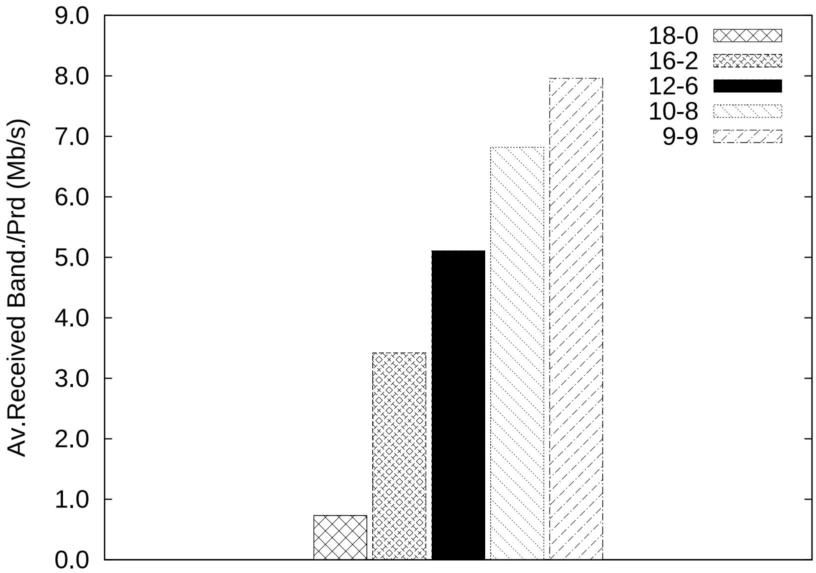


Figure 6.3: *Performance of BitTorrent peers with 18Mb/s physical capacity.*

download and upload capacities in order to improve their performance. The problem with that is that if all peers acted rationally, they would rather use their whole capacity solely for downloading and the system would collapse.

In order to make the above limitations more clear, we study a neighborhood of 50 connected BT peers with the same link capacity of 18Mb/s. We consider that 20% of these peers decide to use their whole capacity for downloading and zero for uploading (pair 18-0 in Fig.6.3), other 20% use 16Mb/s as their download capacity and 2Mb/s as their upload capacity (pair 16-2), other 20% use the pair 12-6, 20% use the pair 10-8 and final 20% are fair by giving equal portions to their upload and download capacities (pair 9-9). The upload and download limits can be set by a user through an application program like the "speed scheduler" [78], a plugin for popular BT client, Azureus.

In Fig.6.3 we can see the average received bandwidth per period for each category of BT peers. The results are disappointing, as for all category peers the average bandwidth they receive per period is less than the half of their total physical capacity. Fig.6.3 is also representative of the well-known unfairness issues in BitTorrent [25, 31, 35, 36]. Some peers contribute more resources than what they receive (e.g., peers with pair 10-8 contribute 8Mb/s each period and receive on average 6.8Mb/s per period) and some others contribute less than what they receive (e.g., peers with pair 16-2). Even worst, there are peers that although they do not contribute any resources (pair 18-0), succeed in receiving on average 0,73Mb/s per period, which is certainly unfair.

Next, we propose two extensions over BitTorrent in order to improve its performance in single capacity-limited link systems.

BT-fair From Fig.6.4 we see that peers are better off by offering half of their capacity for uploading and half for downloading; thus we propose that peers are imposed by the software to use half of their capacity for uploading and half for downloading. We call this scheme BT-fair. However, peers who use this BT extension may wish to hacker the software, hoping to improve their performance.

BT-alg We further propose BT-alg policy, which combines the current implementation of BT with our capacity allocation strategy of Algorithm 1 described in chapter 5. Each BT client will dynamically decide his upload and download capacities, by performing our capacity allocation strategy every period, and he will then determine how to allocate his upload capacity among his requesters, based on the BitTorrent's reciprocation mechanism [60]. Our algorithm works rationally, seeking to maximize the peer's benefits and therefore BT-alg is less probable to be hackered than BT-fair. In cases of excess resources available in the system, BT-alg in contrast to BT-fair can allow peers to set their download rates higher than their half link capacity.

In Fig.6.4 we can see how the performance of the 18Mb/s capacity BT peers improves when any of our proposed BT policies is used. While, in Fig.6.3 the average performance of all 18Mb/s peers is only 4,8Mb/s due to their different spontaneous decisions on capacity allocation, this is improved to 7.94Mb/s if they all use BT-alg to dynamically allocate their capacity between uplink and downlink flows, and 8.17Mb/s if they all use BT-fair. Their performance can be even more improved to 8.75Mb/s with any of our BS/RA-g49 and GS/RA-g49 policies or 8.9Mb/s with either AS/RA-g49

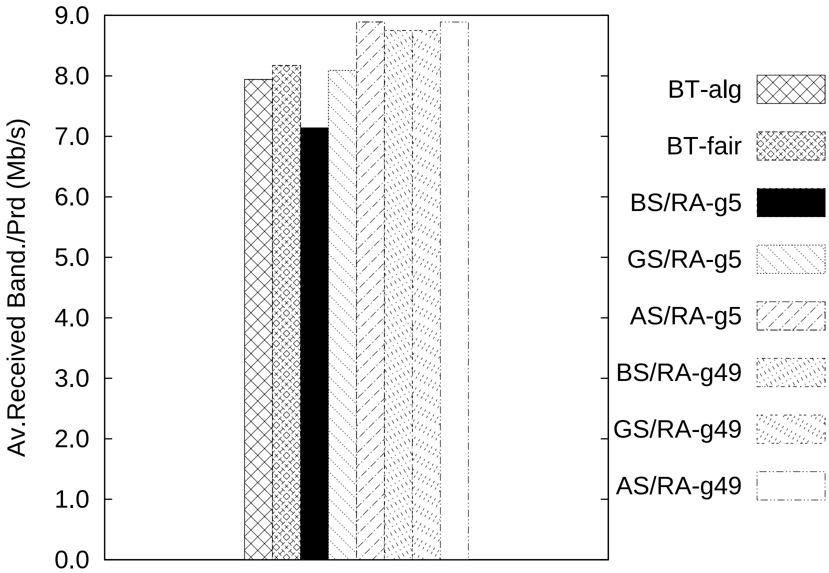


Figure 6.4: *Performance of 18Mb/s capacity peers for various proposed policies.*

or AS/RA-g5 policy. The "g" indication next to our policies declares the request generation profile of the peers, which can be determined by the protocol. In case of g49, a peer sends requests to all of his neighbors and as a server he may serve during a period the maximum 49 peers, based on our policies. The case of g5 better complies with the basic version of BitTorrent, where a server serves five peers per period; thus, in the rest of our analysis we will consider that peers who follow our policies have a request generation profile of $g = 5$.

Motivating seeders

As for the ADSL type systems we study the performance of the seeders for single capacity-limited link systems, as well. We study 50 connected peers with physical capacities of 18Mb/s who download blocks of files. However, we consider that one of these peers has finished downloading a file (seed) in period 100 but remains in the system and uploads with his total physical capacity for a duration of 500 periods. After this duration, i.e., in the 600th period, he starts downloading once again (e.g., other files).

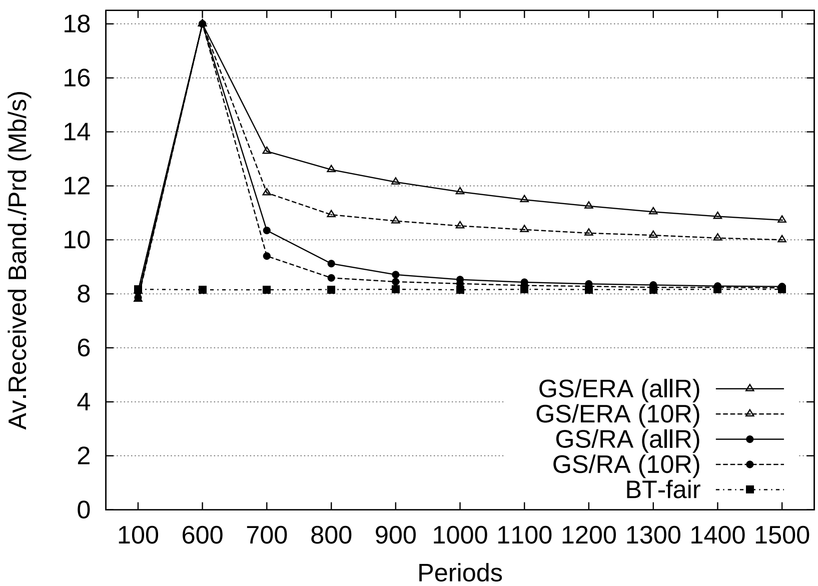


Figure 6.5: *Performance of seed.*

In Fig.6.5 we can see the performance of the seed before and after his uploading 500 periods. We exhibit the performance of our GS/RA policy along with BT-fair policy, as these two have similar performance in case of no seeds in the system (see Fig.6.3). ERA is the policy described in eq.(3.3) with parameter a equal to 1. Definition (allR) stands for peers who calculate one's reputation based on the whole history of transactions with him, while definition (10R) stands for our default case, according to which peers use the history of the 10 most recent transactions to calculate one's reputation. We can see that when BT-fair is used, the seed's performance is 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 any of our exhibited policies is used, the seed's performance is improved when he starts downloading again and this improvement remains for many more periods when any one of the GS/ERA schemes is used (either allR or 10R). This is due to the fact that during the periods that the seed uploads to other peers, the number of requests directed to him compared to the ones he has sent to others significantly increases; thus from eq.(3.3) it is obvious that he is going to be given high priority in his future requests. Which of the aforementioned policies

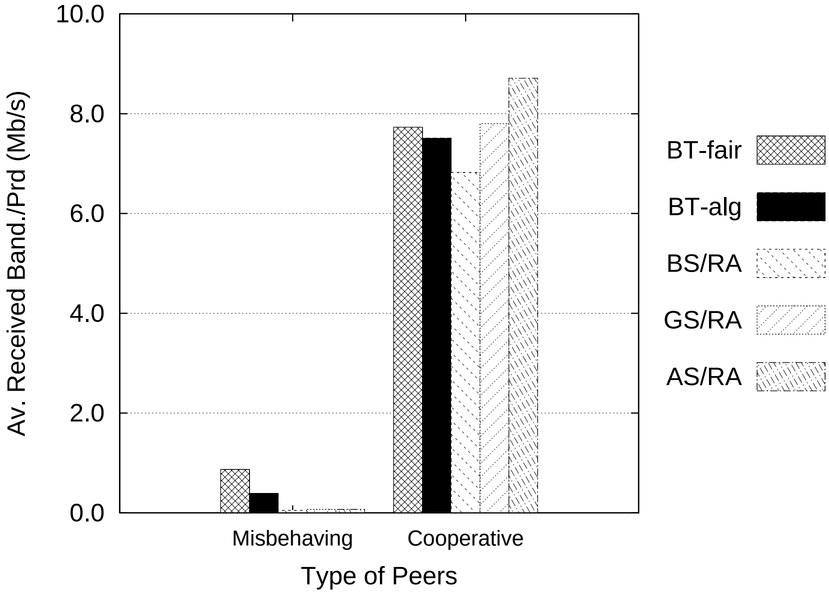


Figure 6.6: *Performance of misbehaving peers.*

is preferable, depends on the network designer's perspectives (e.g., how much he wants to privilege contributive seeds).

Deter Misbehavior

Here we consider that 50% of the 18Mb/s connected peers are misbehaving (do not contribute any resources and do not follow any of ours or BT policies). What we see from Fig.6.6 is that when our policies are used by the other peers, misbehaving peers cannot receive almost any resources from the network in contrast to BT policies which cannot deter misbehavior and thus are more susceptible to strategic clients, as also reported in [36, 63].

6.4 Summary

In this chapter, we adapted and evaluated our proposed reputation framework over BitTorent, both for ADSL-type and single capacity-limited link systems. Our reputation-based policies exhibit significant advantages over BitTorrent reciprocation mechanism which fails to deter misbehaviour and motivate seeds to remain and contribute in the system. Moreover, BitTorrent lacks of a dynamic capacity allocation algorithm for single capacity-limited link systems to help them improve their performance. We have shown that our capacity adaptation algorithm can improve the performance of BitTorrent under such systems, especially when it is combined with our reputation-based framework.

Part of the work described in this chapter has been published in the following journals:

* A.Satsiou, L.Tassiulas, Trust-Based Exchange of Services to Motivate Cooperation in P2P Networks, accepted for publication in Peer-to-Peer Networking and Applications Journal, Springer.

* A.Satsiou, L.Tassiulas ”Reputation-based Resource Allocation in P2P Systems of Rational Users”, IEEE Transactions on Parallel and Distributed Systems, 21 (4), pp 466-479, April 2010.

Chapter 7

Reputation-based Internet Sharing in Wireless Neighborhood Community Networks

7.1 Introduction

In contrast to chapters 3-4 where users decide how to offer their available service capacity among the competing peers, under the scenario considered in this chapter, each user is requested to share his internet capacity among *his own* and other users' traffic. Although at a first glance it looks like the problem we faced in chapter 5 where peers take decisions on how to share their capacity among their own and other peers' needs, the latter one refers to file sharing systems like BitTorrent, where peers act as servers and clients at the very same time. The internet sharing scenario introduces a different problem which motivated us to propose a different reputation system than the one proposed in previous chapters. Moreover, the implementation of a reputation-based allocation policy in existing 802.11 APs and clients, necessitate a careful and appropriate design.

The main contributions of this chapter can be summarized as follows:

a) we propose a reputation metric to reflect the cooperation level of each user in the community, in terms of the quality of service he provides. Reputation metrics dynamically adapt to possible users' cooperation variations,

b) we propose a reputation-based allocation framework according to which each user independently decides how to allocate the available resources of his Internet connection among his visiting requesters based on their reputations and consumption profiles, and how much he prioritizes his own traffic over his visitors',

c) we propose a reputation-based AP association/selection policy to help users associate with APs belonging to the best community contributors, in order to improve their QoS,

d) we describe how we implement aforementioned allocation policy in IEEE 801.11 APs and clients, with no modifications at existing standards at the MAC or network layer, introducing R-HTB application,

e) we present indicative experimental results of the performance of R-HTB in UTH testbed and individual ORBIT nodes,

f) we evaluate our scheme via extensive simulations in a community of users with different cooperation and consumption profiles, exploring the fairness properties of our policies and overcoming limitations of previous work in this area [69, 70].

The remainder of this chapter is organized as follows: Section 7.2 describes the system model. Our reputation based policies are described in 7.3, while our experimental and simulation results are presented in sections 7.4 and 7.5, respectively. Finally section 7.6 concludes the chapter.

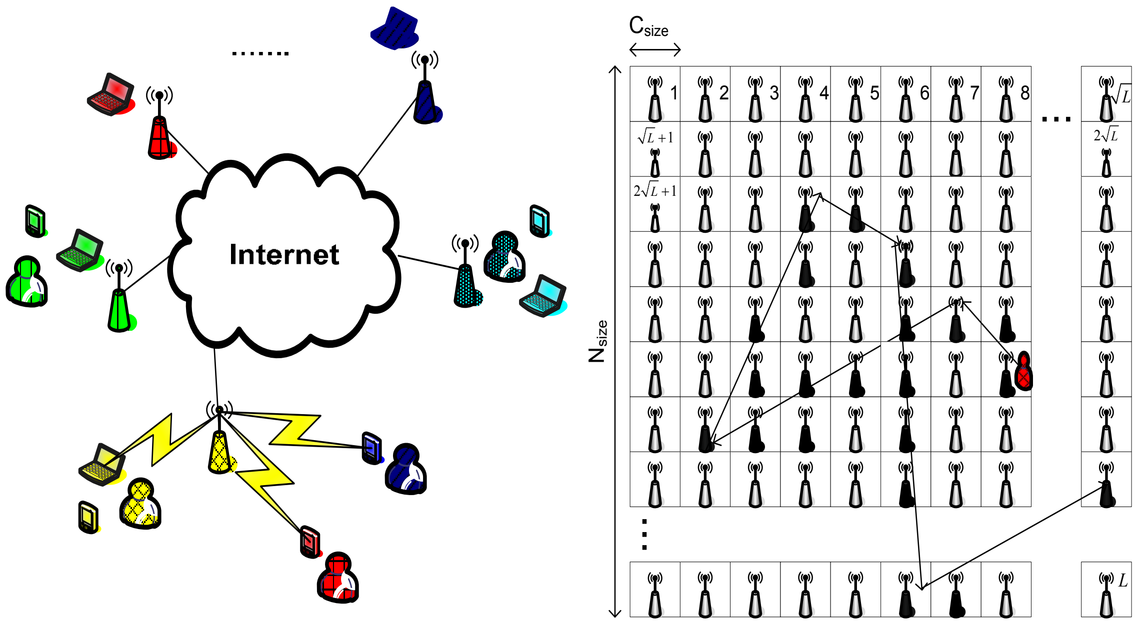


Figure 7.1: *(a) Internet Sharing Community (b) Neighborhood Community Topology.*

7.2 Model Description

We consider a one-hop Internet sharing (IS) community of L users who own an AP and a wired connection to the Internet and share their available wired and wireless bandwidth among their community members, when the later are mobile in their vicinity (please refer to Fig.7.1(a)). We denote the wired capacity of user u as $Cwired_u$ and the wireless capacity of his connection at a given time t as $Cwireless_u^t$. The available wireless capacity of user u varies with time and depends on several factors, like the channel conditions, the number of associated users and their link rate, MAC layer impairments such as hidden terminals, etc. As we will see in our experiments in the testbed, the wireless capacity of a user's AP at a given time can be estimated by summing the average throughputs of the clients for a small duration of time, when communicating with the AP. Thereafter, the maximum internet bandwidth each user u can provide to himself and the connected clients at a given time t is $Cmax_u^t = min(Cwired_u, Cwireless_u^t)$ and is considered to be the bottleneck in the connection.

We model IS users with different cooperation levels, determined either from the priority P_u they give to their own traffic or the probability $Popen_u$ they have their AP open and available to IS users. Priority P_u determines the amount of bandwidth user u will reserve for his home users (e.g., himself, family, etc) and his community requesters, as we will see in section 7.3.2. We consider that the APs of the IS users are arranged in a square area (see Fig.7.1(b)), which is divided in L subsquares (cells) with a side of C_{side}, at the center of which each AP is placed and covers the whole cell. The length and width of the studied neighborhood is $N_{size} = C_{size} * \sqrt{L}$. In Fig.7.1(b) we further see how the community identities (IDs) of the users are determined according to their location, starting from ID 1 to ID L. Each user has at least one WiFi-enabled device with which he can use the internet connection of other community members as he passes by through his neighborhood. All his devices along with his own AP are distinguished with the same community ID.

The consumption profile of each user u is reflected by the probability with which he requires Internet service at any minute from his community, $Preq_u$. A user with higher $Preq_u$ tends to occupy the resources of the community APs for more time. Throughout the user's walk toward his destination and during his stay there, he requests Internet service with probability $Preq_u$ from the nearest to his position at any time AP, in compliance to 802.11 standard (see the black APs in Fig.7.1(b)), or according to the policy proposed in 7.3.3.

Each time t, an AP owner receives Internet connection requests from IS users he allocates his available capacity to them, as described in section 7.3.2. Each IS user then calculates the reputation of the owner of the serving AP based on his perceived quality of service (see section 7.3.1) and sends this information to his home AP. In case IS users would like to secure their traffic against eavesdropping attacks by possible untrusted visited APs we consider that they could open a VPN tunnel to their home APs and use it to tunnel their Internet traffic. This functionality could be incorporated in IS APs.

7.3 Reputation-based Internet Sharing

7.3.1 Reputation System

Each IS user keeps a local reputation value for each one of the members of his community according to the quality of Internet connection he is offered by them. The physical proximity of the users in a wireless neighbourhood community and the fixed location of their home APs lead to long-lived relationships and foster frequent transactions between them. Consequently, each user can acquire his own personal opinion about the cooperation of any member of his community. On the other hand, when users use recommendations to determine one's reputation, they can speed up their perception of their community. However, there should be mechanisms to guarantee the trustworthiness of the users who vote which induce complexity to the system [23].

Unknown or new IS members are given an initial small reputation R_{init} in order to have the chance to receive some resources. If they prove to be collaborative enough, their reputation will quickly increase, permitting them to receive more resources, as we will see in the sequel. Each user u further keeps a variable named $Rtop_u^t$, which is initially set equal to R_{init} and is updated to hold the maximum perceived average QoS from the community, as described in algorithm 2.

Each time t a user u requests service from user i, he assesses the total amount of bytes he sends and receives through i's connection, b_{ui}^t, during the service time, s_{ui}^t. Then, he can calculate the average bandwidth he is offered by user i by:

$$B_{ui}^t = \frac{\sum\limits_{\tau \in \mathrm{T}_{ui}^t} b_{ui}^\tau}{\sum\limits_{\tau \in \mathrm{T}_{ui}^t} s_{ui}^\tau}, \tag{7.1}$$

where T_{ui}^t denotes the set of the specific time instances in which user u requested service from AP owner i, till time instance t, including time instance t. Please note that B_{ui}^t expresses the average bandwidth (or elsewhere QoS) that user u received from user i, calculated from his admission to the community till the end of the service that started at time t.

We have seen from our simulations that a history of 10 past and most recent connections with the AP of a given user is enough to evaluate his reputation. So, each time a user u completes 10 transactions with any user i, he starts replacing the oldest values of b_{ui} and s_{ui} with newer ones, always keeping a history of, at the most, 10 past values, maintaining the overhead low. Therefore, B^t_{ui} is calculated based on the 10 most recent transactions log in our simulation studies. After calculating B^t_{ui}, user u computes the local reputation of i to him, R^t_{ui}, as described in algorithm 2.

Algorithm 2 Reputation calculation

if $B^t_{ui} > Rtop^t_u$ **then**

  $Rtop^t_u = B^t_{ui}$

  for $j = 1$ TO L **do**

    $R^t_{uj} = \frac{B^t_{uj}}{Rtop^t_u}$

  end for

else

  $R^t_{ui} = \frac{B^t_{ui}}{Rtop^t_u}$

end if

It is obvious that reputations take values between [0,1]. The most reputed users to u have reputation equal to 1, while all others' local reputations are proportional to the ratio of the QoS they offer over the one offered by the most local reputed users. User u will update the local reputations of all community members in case of ($B^t_{ui} > Rtop^t_u$), i.e., when the average QoS he receives from user i exceeds the maximum perceived average QoS from the community, so far.

7.3.2 Reputation-based Internet Sharing Policy

Suppose that at a given time t, an IS user u receives internet connection requests from community members in his vicinity. We denote by V_u^t the set of his visiting requesters at time t. Then, we propose that u will allocate his maximum available capacity $Cmax_u^t$ among his competing visitors according to:

$$W_{ui}^t = \frac{(M_{ui}^t)^a R_{ui}^t}{\sum\limits_{j \in V_u^t} ((M_{uj}^t)^a R_{uj}^t) + P_u} Cmax_u^t, \tag{7.2}$$

where W_{ui}^t represents the allocated bandwidth from user u to i at time t and $M_{ui}^t = \frac{MinCounter_{ui}^t}{MinCounter_{iu}^t}$. $MinCounter_{ui}^t$ represents the total amount of time in minutes that user u received service from user i in past requests till time t; therefore, M_{ui}^t represents how much user u occupied the connection of user i compared to how much user i occupied the connection of user u in previous requests till time t. These variables are initialized to 1 and change through time. Parameter a can be adjusted by the software designer to balance the fairness in the IS system. The bigger a is, the more restrictive the policy is for the greedier users. We note that user u keeps the remaining bandwidth of his connection for his own use and is given by $W_{uu}^t = \frac{P_u}{\sum\limits_{j \in V_u^t} ((M_{uj}^t)^a * R_{uj}^t) + P_u} Cmax_u^t$ and is proportional to his own priority P_u. The bigger priority a user puts to himself, the less resources will be donated to his visitors. In case of $a = 0$, i.e., when the users' consumption profiles are not considered in the allocation decision, or in case of $M_{uj}^t = 1 \; \forall j \in V_u^t$, a priority $P_u = 1$ means that u will equally share his resources among himself and another user $i \in V_u^t$ with reputation $R_{ui} = 1$, which is the maximum attainable reputation in the community.

Each time a client completes service, or a new visitor requests service, user u dynamically reallocates his resources based on (7.2). By using this policy, u prioritizes most cooperative community members who further tend to occupy his Internet connection less than he occupies theirs. In this way, he somehow repays them for the excess service he has received from them in order to hope for more future collaboration with them. We see that user u takes his allocation decisions based not only on the average QoS that he has received from his requesters but also on the amount of time he has reserved their resources. This information is very important, since there may be more than one competitors who offer similar QoS Internet connections but for much different session times; the one who has offered his connection for longer time should be prioritized.

7.3.3 Reputation-based AP Selection Policy

In this section, we propose a reputation-based AP selection policy, refereed to as R-AP, in order to help users avoid requesting service from less contributive users and improve their received QoS. Under this scenario, we consider that a user u in a given cell can also hear the active APs in the neighboring cells and can select with which one to associate. Optimally, we suppose that he can attain equally good channel conditions with all of his neighboring APs consisting the set A_u^t at time t and selects to associate with AP $i \in A_u^t$ with probability p_{ui}^t equal to:

$$p_{ui}^t = R_{ui}^t / \sum_{j \in A_u^t} R_{uj}^t \tag{7.3}$$

A more sophisticated association policy would select the AP that has the best pair of reputation and channel conditions at the time of request; however, in our simulation study we focus on the cooperation levels of the AP owners, assuming optimal channel conditions.

7.4 Implementation of Proposed Scheme

In order to realize our reputation-based internet sharing scheme over existing 802.11 APs and clients, we designed a reputation-based application based on HTB tool [48], which we call R-HTB and describe in detail in following sections.

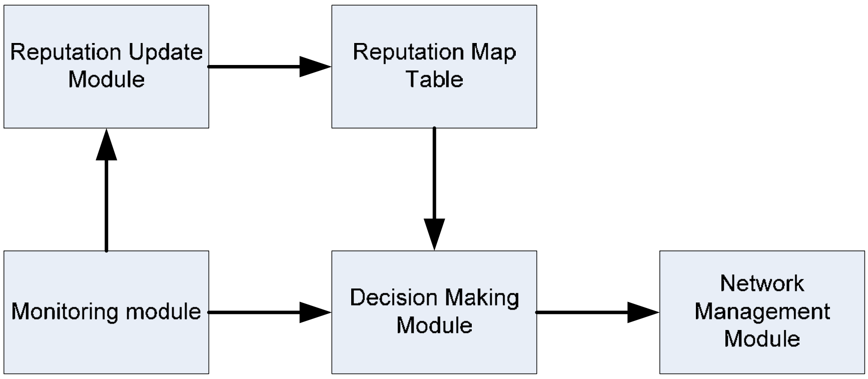


Figure 7.2: *Reputation-based application in the AP side.*

7.4.1 The reputation-based application in the AP side

The application that runs in the AP of an IS user u consists of the modules presented in Fig.7.2.

The monitoring module is responsible to listen whether data traverse the link between the AP and its associated clients. As soon as more than one client occupies the available link with the AP, the monitoring module estimates the average wireless throughput for duration of 5 seconds by summing the average bit rates of the clients. It then informs the decision making module about the estimated capacity C_u^t of the link and the IP addresses of the associated clients who transmit or receive data.

The decision making module gathers the information about the reputations of the associated clients from the Reputation Map table along with the MinCounter variables and then performs the allocation policy described in section 7.3.2 to determine the offered bandwidth W_{ui}^t to each client. Finally, the decision making module informs the network management module about the offered bandwidth to each associated client.

The network management module uses the hierarchical token bucket filter tool (HTB) [48], located in the IP layer, to control how the packets are delivered to MAC layer. HTB is a class based queue discipline which can be seen as a black box able to queue and dequeue packets in order and at times determined by the algorithm hidden in it. The decision making module sets the HTB Limits for each client i equal to W_{ui}^t apart from the most reputed user. The HTB Limit determines the maximum rate with which a client can transmit or receive data. The main advantage of this tool is that it is easily applicable to 802.11 APs and clients with no modifications at existing standards at the MAC or network layers.

When the clients only download data, the HTB Limits are only set at the AP and are equal to W^t_{ui} for each active client i. However, when the clients both upload and download data, the HTB Limits must be set both at the AP and the client side. Thus, the network management module sets the HTB limit equal to $W^t_{ui}/2$ and inform the associated clients about the HTB Limits that they should set, equal also to $W^t_{ui}/2$ for each client i. In that case, the maximum available bandwidth for a client i both for uploading and downloading data would be equal to W^t_{ui}.

The monitoring module keeps informing the decision making module about the estimated capacity of the link and the active associated clients each time a new event occurs (an active client becomes inactive or disassociated, or a new client becomes active), or in case of no new events, it informs the decision making module every 5 seconds. The decision making module updates the resource allocation decisions accordingly, and HTB adaptively conforms to these decisions.

The monitoring module of an AP should also be responsible to receive reports from its owners about the quality of service they receive from other community members in order to inform the reputation update module to calculate the new reputations according to algorithm 2 and update the reputation map table. We haven't implemented this final stage, since our focus is on service differentiation among users based on their reputations, which are considered known.

7.4.2 The reputation-based application in the client side

The application that runs in the client side consists of the modules presented in Fig.7.3.

The monitoring module is responsible to collect the information about the HTB Limits. It then informs the network management module, which is responsible to adapt the HTB Limits. The network management module should also inform the user's home AP through a VPN tunnel about the estimated received quality of service from different community members. However, for reasons already

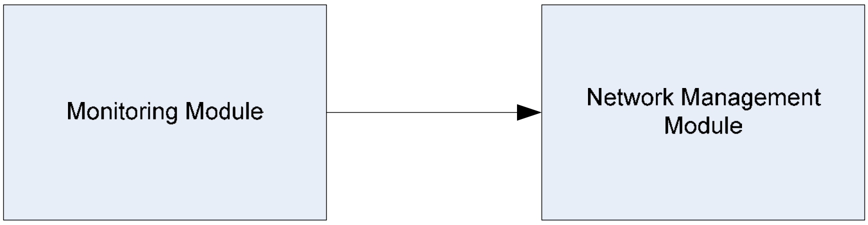


Figure 7.3: *Reputation-based application in the client side.*

explained, we have't implemented this stage.

7.4.3 Experimental results

The experiments took place both in the UTH Wireless Testbed [79] and individual nodes. ORBIT nodes have been used with MADWIFI driver and Linux kernel 2.6.16.19. In order to evaluate our policies, we use Iperf tool [80] to generate UDP and TCP traffic and measure the throughputs of the clients when they send and receive traffic from the AP. Wireless nodes are based on commercial WiFi wireless cards as per the the IEEE 802.11 standard. In the following experiments we use the 802.11b/g mode of the cards.

First, we consider the implementation scenario of Fig.7.4. We have four clients associating with the AP of community member A. The reputations and IP addresses of the clients are as shown in Table 7.1. We assume that the owner of AP A sets his priority at 0 and the variables M are incorporated in the reputation values. In the sequel, we will show via several indicative scenarios how R-HTB can dynamically adapt to possible changes in the medium and traffic and realize our proposed allocation scheme.

Scenario A: Clients transmitting at different rates

An intrinsic functionality of 802.11b progressively degrades the bit rate of a host when repeated unsuccessful frame transmissions are detected. However, when one host captures the channel for a long time because its bit rate is low, it penalizes other hosts that use a higher rate, inducing a

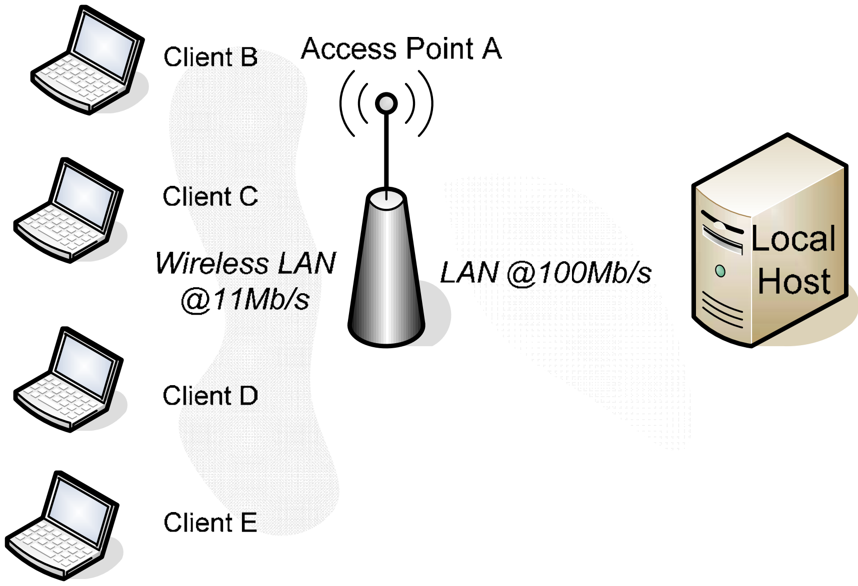


Figure 7.4: *Implementation Scenario.*

Table 7.1: *Reputation Map Table*

Client ID	IP address	Reputation
B	11.11.11.2	0.2
C	11.11.11.3	0.4
D	11.11.11.4	0.6
E	11.11.11.5	0.8

performance anomaly. Here we examine such a scenario and see how our application can jointly cope with performance anomaly and service differentiation. Suppose that at time 0s only client C is active in the hotspot transmitting UDP traffic at 11Mb/s with a useful throughput of approximately 6.5Mb/s. After 10s client B also starts transmitting UDP traffic with a rate of 11Mb/s. However, we suppose that client B experiences a lower quality channel and its transmission rate progressively decreases according to 802.11 standard. We model this by decreasing his transmission rate every 20 s from 11Mb/s to 5.5, 2, 1 and then increase it again at 11Mb/s. In Fig.7.5 we see the throughputs of the two clients when either a) FIFO scheme or b) our R-HTB scheme is used.

When the simple FIFO scheme is employed, the throughput of client C is severely degraded when the transmission rate of client B decreases. More specifically, the throughput of client C falls down

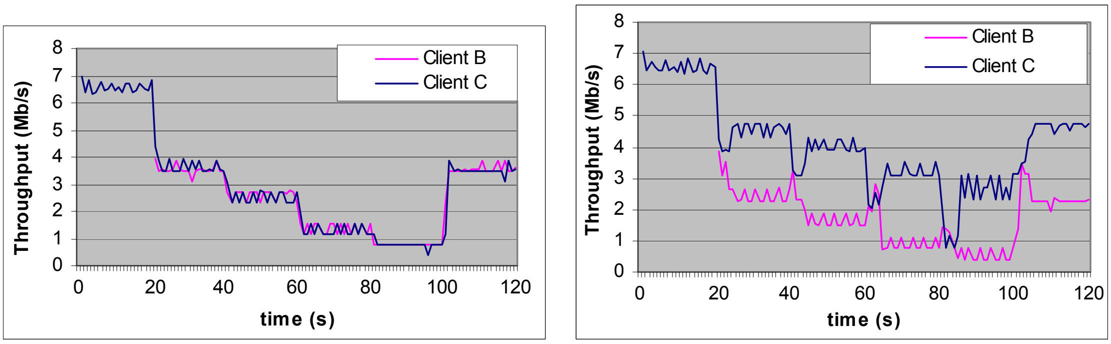


Figure 7.5: *Throughputs of clients for (a) FIFO scheme and (b) R-HTB scheme under scenario A.*

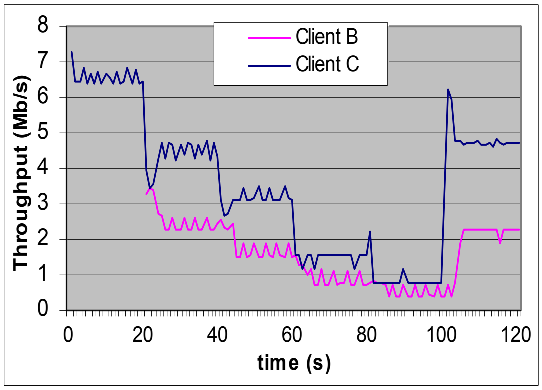
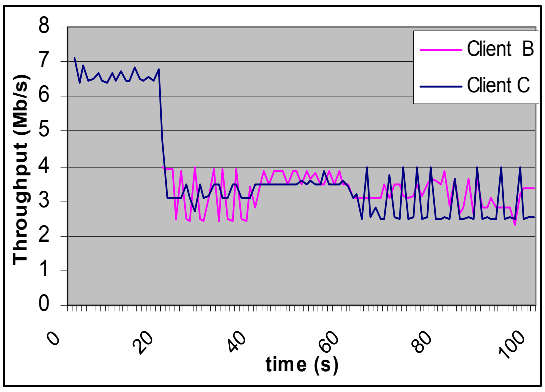


Figure 7.6: *Client C experiences a poor quality channel*

the performance of the low rate client B. In order to alleviate this so called "performance anomaly" and boost the performance of the most reputed client C we use our R-HTB application. What we see from Fig.7.5(b) is that the throughputs of the two clients are proportional to their reputations, despite their different transmission rates and the performance of the most reputed client cannot be severely affected by the performance of the low rate and reputation client.

Next, we consider that the client who experiences the poor quality channel at t=40s is the most reputed one, i.e., client C, and we control his transmission rate in exactly the same way as we did for client B in previous experiment. With R-HTB scheme, when Client C's transmission rate becomes very low (60-100s), its performance cannot significantly improve despite the substantially restricted rate of client B (please refer to Fig.7.6). However, the ratio of the throughputs of the two clients is



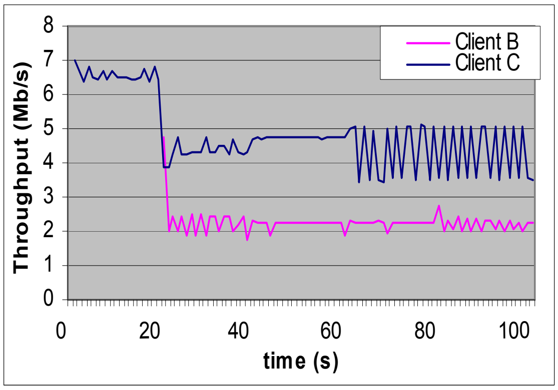


Figure 7.7: *Throughputs of clients for (a) FIFO and (b) R-HTB scheme under scenario B*

almost equal to the ratio of their reputations, despite their different transmission rates.

Scenario B: Clients uploading different kind of traffic

In this section, we present the performance of R-HTB for different kind of transmitted traffic. We consider that at time 0s only client C is active, sending UDP traffic, while at time t=20s client B becomes also active sending TCP traffic for 20s and then UDP for another 20s. Finally, at t=60s client C stops sending UDP traffic and starts sending TCP traffic and at t=80s both B and C send TCP traffic. In Fig.7.7 (a) and (b) we see the throughputs of clients in case of 802.11 with FIFO and our R-HTB scheme, respectively. With R-HTB scheme, the ratio of the average throughputs of clients is approximately equal to the ratio of their reputations for any kind of traffic.

Scenario C: Clients both uploading and downloading data

In this scenario we present the performance of R-HTB for concurrent uploads and downloads. We consider that at t=0s both clients B and C upload UDP data, while at t=50s they also start downloading UDP data and at t=100s they stop uploading and only download data for another 50s. In Fig.7.8 (a) and (b) we see the throughputs of clients in case of 802.11 with FIFO and our R-HTB scheme respectively. While with FIFO, each stream is allocated an approximately equal share of bandwidth, with R-HTB scheme, the ratio of the total throughputs (sum of upload and download throughput)

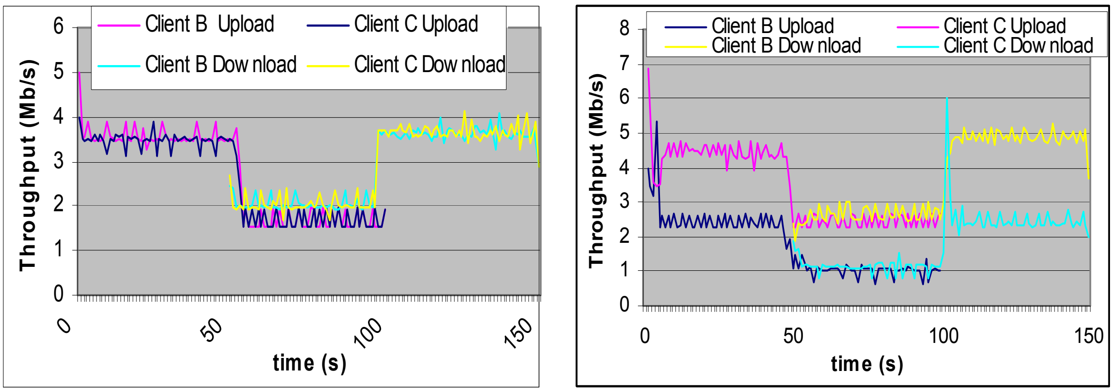


Figure 7.8: *Throughputs of clients for (a) FIFO scheme and (b) R-HTB scheme under scenario C.*

of clients is approximately equal to the ratio of their reputations. All aforementioned case studies indicate the ability of R-HTB in succeeding clients' throughputs proportional to their reputations.

Scenario D: Multiple clients downloading data

Here, we consider that progressively all four clients become active in the system downloading data. At t=0, only client B is active, downloading UDP data from the AP, at t=20s client C becomes also active downloading UDP data, and at t=40s and t=60s clients D and E become active, respectively downloading UDP data. The traffic duration of all clients is considered to be 80s and their transmission rates are set to auto (i.e., they are adapted according to the medium). In Fig.7.9 (a) and (b) we see the throughput of each client through time, for FIFO and R-HTB, respectively. We can see that with R-HTB the ratio of the users' throughputs is almost equal to the ratio of their reputations at any time they are active, as desired.

Scenario E: Mobile clients in UTH testbed

Last, we experimented with a mobility scenario in UTH testbed. As it can be shown from Fig.7.10, the AP is node 11, client A is in fixed position next to node 05 and mobile client B moves from position 13 (i.e., next to node 13) to position 07 and finally position 12. Via iperf tool, and sending traffic between clients in the different positions and the AP, we investigated that clients in position

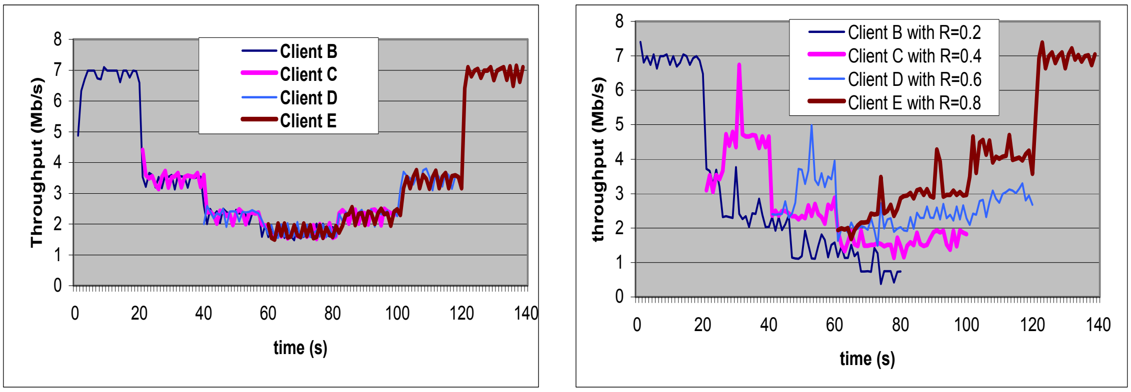


Figure 7.9: *Throughputs of clients for (a) FIFO scheme and (b) R-HTB scheme under scenario D.*

05 and 13 have similar performance when associating alone with the AP (node11), while client in position 07 has worse and client in position 12 has better performance than in position 13 and 05. For example, when AP 11 sends UDP traffic of 20Mb/s for 20s to node 05, the average throughput of node 05 is 7,5 Mb/s, when sending to node 13, the average throughput of node 13 is 7,32 Mb/s, when sending to node 07, the average throughput of node 07 is 3,85 Mb/s and when sending to node 12, the average throughput of node 07 is 18Mb/s.

Client A is the most reputed client among the two and has a reputation equal to 0.8, while client B has a reputation equal to 0.4. Initially, at t=0 only client A is associated to the AP and receives service, while at t=20s client B also connects to the AP from position 13 and receives service. At t=40s, we consider that client B moves to position 07 and at t=60s he is in position 12, always receiving service from AP 11. The goal of our proposed R-HTB is to provide to each of the competing clients an average throughput in proportion to his reputation so that more reputed users can have a better performance in the community and be given motivation to improve their contributions to it. Ideally, we would like the ratio between the average throughputs of the two clients to be equal to the ratio of the reputations, irrespective of the different channel conditions sensed by the users. Therefore, under this experiment we would like to see that the throughput of client A is approximately twice the one of client B in whichever position client B is (and different channel quality he perceives).

In Fig.7.11(a) we can see the performance of the two clients when the FIFO scheme is employed.

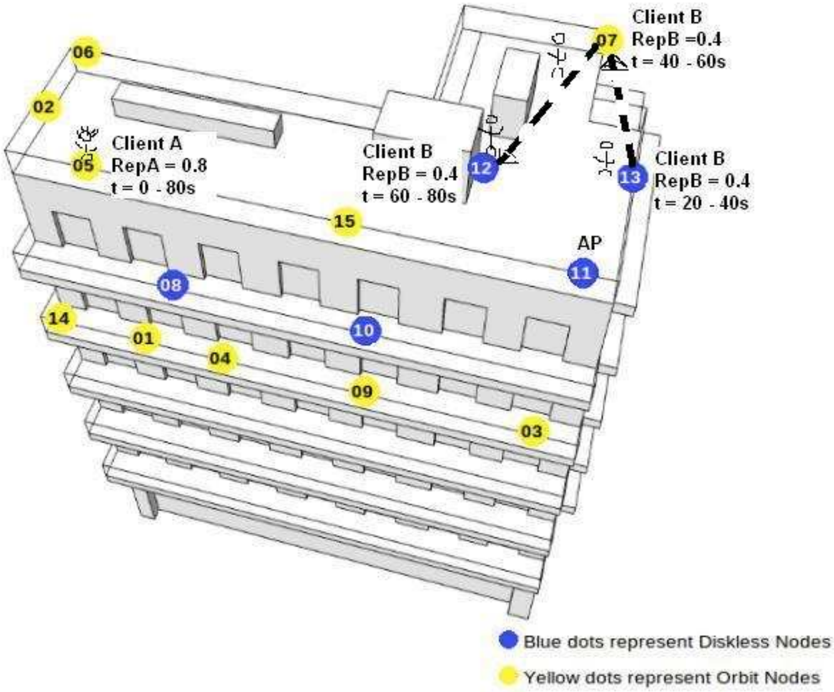


Figure 7.10: *Scenario E in UTH testbed*

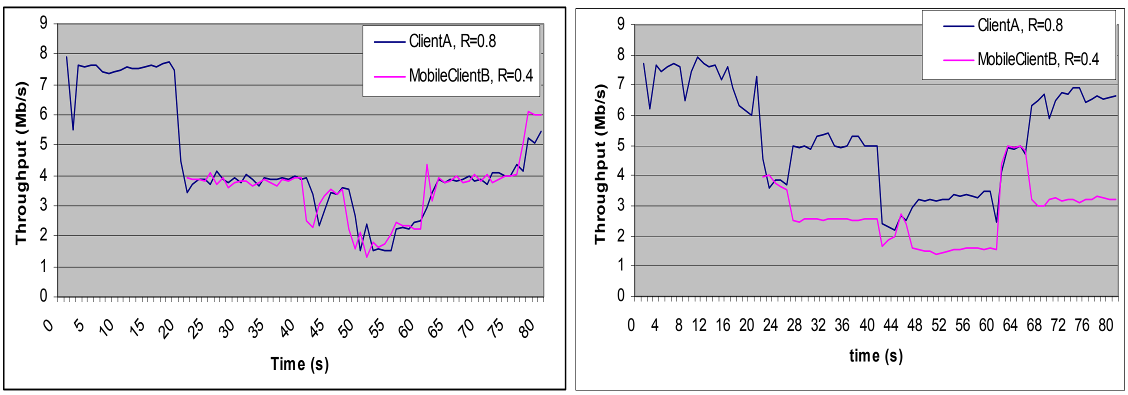


Figure 7.11: *Throughputs of clients for (a) FIFO scheme and (b) R-HTB scheme under scenario E.*

It is notable that the performance of client B deteriorates as he moves to position 07 after 40s. Due to his performance deterioration, the performance of client A also degrades, as was expected from the known performance anomaly of 802.11b/g. In the 60th second, when client B reaches position 07, his performance improves, however it is constrained by the worse channel conditions of client A.

In Fig.7.11(b) we can see the throughputs of the two clients in the different stages of the experiment when our R-HTB scheme is used. As we can observe, with R-HTB scheme the throughputs of the two clients are in proportion to their reputations, and more specifically, the ratio of their throughputs is almost equal to the ratio of their reputations during all stages of the experiment.

7.5 Performance Evaluation

7.5.1 Simulation Model

In previous section we presented that under R-HTB, we can achieve throughputs of the competing users proportional to their reputations for any channel conditions, type of traffic and number of clients. In this section we will present that by succeeding this analogy in reputations and throughputs, our reputation-based system lead members of a community to cooperation by guarantying that the more cooperative a member is by offering a good quality of internet connection through his home AP to community passersby, the better quality of service he can receive from his community, when being mobile inside it. Towards this goal, we performed simulations on a custom-made simulator in C++. Since the channel quality of each competing user does not affect the performance of our allocation policy, as we have shown in section 7.4, in this section we see how users of different cooperation levels perform in a dynamic large-scale setup, ignoring the lower layer conditions. Therefore, we consider that the maximum capacity of the Internet connection of each IS member is the same for all users at any given time and equal to $Cmax = 11Mb/s$.

We simulate the neighborhood community of Fig.7.1(b) with $L = 100$ and $Csize = 200m$. We consider that IS users move in their neighborhood according to the random waypoint mobility model [81], widely used in the ad hoc network community. In the beginning of the system, users are randomly placed anywhere inside their community. Each user then randomly selects a destination point inside the square area and moves to his destination on a straight line (see Fig.7.1(b)) with a randomly chosen speed S_u^t, uniformly distributed between $[1m/s, 20m/s]$. As soon as he reaches his destination he stays

Table 7.2: *Default Values of Variables*

Variable	Notation	Default Value
L	Number of community members	100
Cmax	Maximum capacity of community members	11Mb/s
Csize	Cell size	200m
Smin	Minimum speed value	1m/s
Smax	Maximum speed value	20m/s
Dmin	Minimum requested session time value	1min
Dmax	Maximum requested session time value	15min

there for a randomly chosen time duration D_u^t, uniformly distributed between $[1min, 15min]$ and then picks another random destination, speed and duration, time, repeating the process. Initial reputation *Rinit* for all users is set to 0.1 and parameter a is set equal to 1, unless otherwise stated. The default values of their consumption and mobility profiles can also be seen in table 7.2.

IS users differentiate in terms of their cooperation levels as explained in following scenarios and related results are presented to show that the more cooperative they are the better QoS they receive. We define the average QoS offered to user u from his community till time t, as the average offered bandwidth from all the members of his community till that time, which is expressed by the ratio of *Total Number of Bytes he sent and received through IS APs other than his home AP, till time t* over *Total Duration he was connected to IS APs other than his own, till time t*. We further calculate the global reputation of user u, given by the average of his local reputations to all other community users. Following results are the average outcome of 10 simulation studies with the same inputs.

7.5.2 Users of different priorities

In this scenario, each IS user u gives a different priority P_u to himself, when he is requested to share his connection with other community members. More specifically, we consider that the population of the community splits in 5 equally-sized groups of randomly chosen members with the following priorities

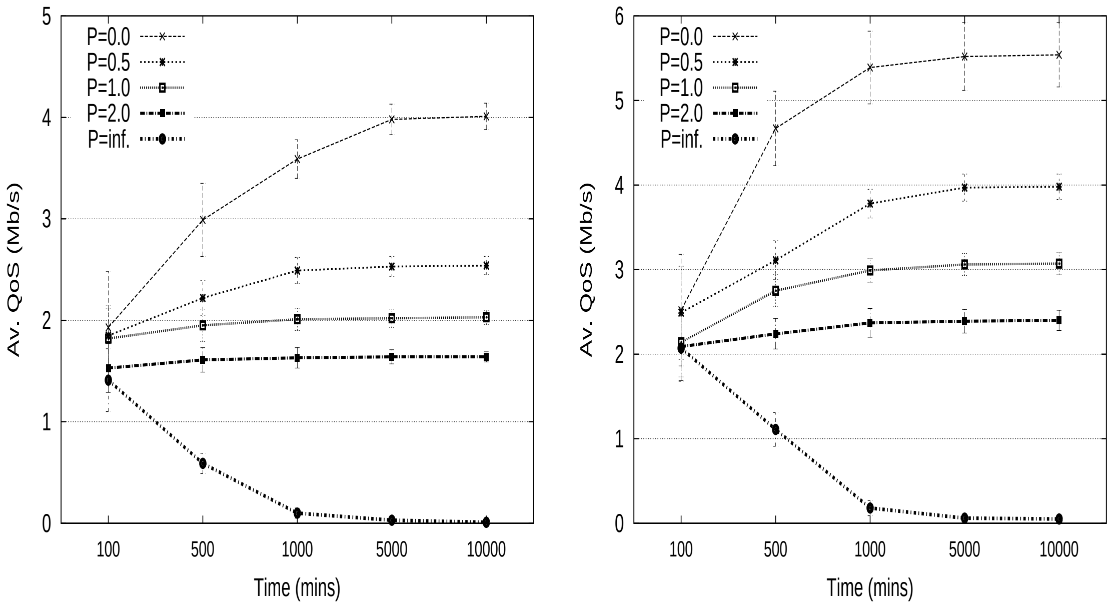


Figure 7.12: *Average QoS through time when (a) no R-AP and (b) R-AP policy is applied for different priority users in the same community.*

(0,0.5,1,2,inf). As it can be seen from eq.(7.2), AP owners with priority 0 are the most altruistic ones, giving 0 priority to themselves and dedicating their whole capacity to community members whenever the latter request service. This group, for example, could involve users who rarely use their home connection but often roam throughout their neighborhood. Members with infinite (inf) priority, on the other hand, represent misbehaving users who do not offer any resources to the community. In the following results, we consider that all users have $Preq_u = 1$, i.e., they are all greedy users seeking to be connected all the time, and $Popen_u = 1$, i.e., they always have their AP open.

In Fig.7.12(a) we see the average received QoS of different priorities' users through time. The vertical bars show the standard deviation between users of the same priority. Initially, the performance of the same priority group users differentiate, but as time progresses they converge to the same value. Users receive a QoS in proportion to their contribution levels, i.e., in reverse proportion to their priorities. In the beginning of time, users do not know each other and thus misbehaving ones succeed in receiving some resources; however as time passes, misbehaving users are revealed and

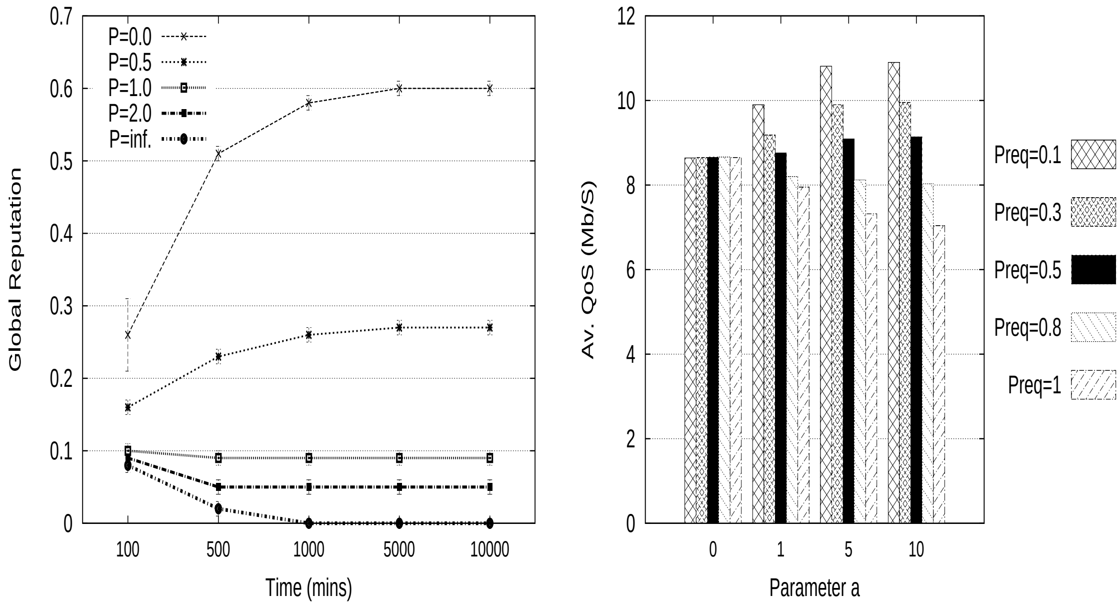


Figure 7.13: *(a) Global Reputations of different priority users in the same community (b) Steady State Performance of different consumption profile users in the same community.*

proposed policies block them from receiving any service. In Fig.7.12(b) we see that the performance of all categories peers (apart from misbehaving ones) significantly improve when they further use the proposed R-AP policy, indicating the potential benefits of such an association policy and the fact that reputations correctly capture the contribution levels of the IS users. This can also be seen from Fig.7.13(a). The global reputations of users at any time are in reverse proportion to their priorities and in proportion to their received QoS (jointly see with Fig.7.12(a)).

7.5.3 Users of different consumption profiles

In this section, we study users of different consumption profiles. More specifically, the population of the community splits in 5 equally-sized groups of randomly chosen members with $Preq_u$ taken from the set (0.1,0.3,0.5,0.8,1). The smaller $Preq_u$ is, the less time user u requires Internet service from his community. For simplicity, we consider that all users have the same priority of $P_u = 0$. What we see from Fig.7.13(b) is that for $a = 0$, i.e., when consumption profiles are not captured in the allocation

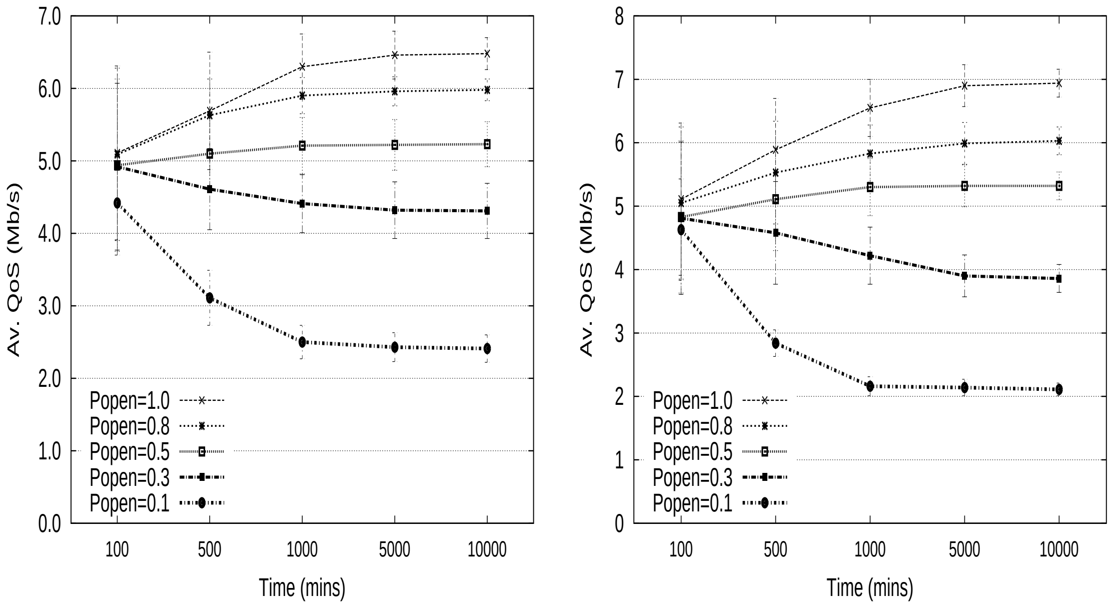


Figure 7.14: *Average QoS through time for (a) $a = 1$ and (b) $a = 5$ for users with different Popen in the same community.*

decisions, all users enjoy a similar average QoS, irrespective of their consumption profile; however since greedy users tend to occupy AP's resources for more time, they consume much more resources in total, than other equally contributive users. For $a > 1$, on the other hand, the fairness is balanced is the system by restricting the performance of greedier users; in this way greedy users cannot exploit the community resources in cost of other equally contributive ones. The higher the value of parameter a is selected by the software designer, the higher the restriction of greedy users. The same conclusions hold for different priorities users.

7.5.4 Users of different AP opening times

In this section, the cooperation level of IS users is determined by the percentage of time they leave their AP open and available to community members. Here, we consider that community population splits in 5 equally-sized groups of randomly chosen members with the same probability of leaving their AP open, $Popen_u$, taken from the set (1,0.8,0.5,0.3,0.1). Users with $Popen_u = 1$ have their

AP open all the time. For simplicity, we assume that under this scenario all users have $P_u = 1$ and $Preq_u = 1$. Users who roam throughout their neighborhood connect to the nearest AP that they can hear (i.e., it is open). From Fig.7.14(a) it is obvious that users who are more cooperative by leaving their AP open for more time, enjoy a better QoS from the Internet sharing community. A user u with higher $Popen_u$, tends to serve community members more time than others and variables M_{iu} for his clients i increase. Thus, the performance restriction of less cooperative users in favor of more contributive ones can be emphasized by using a higher value of parameter a, as can be seen from Fig.7.14(b); the amount of desired performance restriction is due to the software designers' perspectives.

7.6 Summary

In this chapter, we proposed a practical distributed reputation-based scheme to foster cooperation in a wireless neighborhood community, inside where users can enjoy free Internet access through the private contributions of the community APs owners. Our scheme ensures that mobile users enjoy a QoS internet connection in their community proportional to their contribution level; thus, users are encouraged to cooperate and increase their contributions. Proposed allocation scheme is implemented in 802.11 APs and clients with no modifications at existing standards at the MAC or network layers and exhibited via various experimental case studies that users' throughput can be in proportion to their reputation, irrespective of the number of competing clients, the kind of traffic sent or received, or the perceived channel conditions.

Part of the work described in this chapter has been published in the following magazine and conference proceedings:

* A.Satsiou, L.Tassiulas, Reputation-based Internet Sharing in Wireless Neighborhood Community Networks, in Proc. of the IEEE International Conference on Communications, Cape Town, South Africa, 23-27 May 2010.

* P.Antoniadis, B.Le Grand, A.Satsiou, L.Tassiulas, R.Aguiar, J.Barraca, S.Sargento, Community Building over Neighbourhood Wireless Mesh Networks, IEEE Technology and Society, special issue on Potential and Limits of Cooperation in Wireless Communications, 27 (1), pp 48-56, 2008.

Chapter 8

Conclusions

8.1 Overview

P2P overlay systems' rapid and promising evolution is based on their numerous attractive features, providing the efficient means for resource exchange in the Internet. However, their performance is critically depended on the cooperation between the peers. This dissertation presented a reputation-based framework to promote cooperation and fairly regulate the transactions between peers in various different setups, including single and multiple services environments, single capacity-limited link systems, the popular file sharing system, BitTorrent, and wireless internet sharing communities.

Proposed reputation-based framework and system design considers many important aspects of real systems, like peers with heterogeneous consumption (requests and demands), capability and service evaluation profiles, newcomers, misbehaviour and strategic behaviour. As far as we know, no work has attempted to encompass all the above under a common design framework.

In the following we delve into the main points, contributions and conclusions from our investigation and research, as these were evolving during this dissertation, starting from general systems to specific case studies, like BitTorrent and Internet sharing communities.

Single Service Systems

Reputation systems have extensively been studied in the literature to promote cooperation in p2p systems, as being much simpler and practical than game theoretical or pricing-based schemes. However, previous reputation systems were focused on distinguishing peers in two categories, declared either as altruistic and egotistic, or reciprocative and selfish, or contributors and freeriders, and provided binary service differentiation. On the contrary, our focus was on distinguishing peers in terms of different contribution levels. By distinguishing users in more layers (in terms of different QoS offered), we can motivate them to improve their contributions and thus their reputation and perceived QoS. What's more, we were interested in differentiating peers not only based on their contributions levels but also on their consumption profiles, so that greedy users can no longer exploit the network by absorbing much more resources than other similar contributive ones.

Towards this goal we proposed (i) a novel reputation metric that reflects the satisfaction in terms of QoS offered by the given peer (ii) a reputation update mechanism to track the contributive behavior of the peer as the system evolves, (iii) a reputation based allocation policy (RA) which decides the amount of resources offered to each competing peer according to his reputation and demands (iv) an enhanced version (ERA) that further considers the competing peers' consumption profiles in the allocation decision, and (iv) a reputation-based server selection policy (RS) to help peers choose among most contributive servers and avoid misbehaving ones. Our reputation-based framework can be 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.

Proposed reputation-based framework with its desirable properties was firstly examined, via simulation studies, on single service environments where only one resource (e.g., bandwidth) is being exchanged. We presented how our policies greatly outperform previous representative reputation scheme which solely binary differentiate peers and observed how our policies shield the system from (a) misbehaving peers and whitewashers, (b) strategic peers who seek to maximize their satisfaction 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.

Peers benefit by using our proposed 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 further robust to network changes (e.g., peer arrivals and departures).

Multiple Services Systems

As a next step, our interest was directed to multiple services environments. P2P grids are an example of such systems where more than one resources are being exchanged between peers. Previous market-based models that have been proposed in such environments to control the transactions are subject to several complexities, while some reciprocity based mechanism face several limitations, as we point out in the introduction and related work of this thesis. This motivated us to extend our practical reputation-based framework to control the exchange between different services and promote cooperation in such systems. Towards this goal, we defined (i) the reputation vector of each peer consisting of his reputations in providing each service of the system and (ii) each peer's service evaluation profile to determine the value of each service according to his criteria. Thereafter, our allocation policy was extended , forming RA-MS policy, to consider the reputation vectors of the competing peers and the service evaluation of the server. Under RA-MS 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 provides more resources to those peers who have offered him more satisfaction (captured by their reputations) in the services he values more, in order to repay them and foster their bond.

Our simulation studies in multiple services environments demonstrated that proposed policies outperform previous work, leading to the dynamic formation of coalitions (cooperation) among peers, who mutually profit by the exchange of their services, without pre-existing knowledge of one another's capability and service evaluation profiles. In this way, the utilities of all peers progressively increase. Only misbehaving (non contributive) peers do not benefit, as proposed policies efficiently recognize and block misbehavior. Peers' coalitions are further adaptive to network changes and self-organized accordingly.

Single Capacity-Limited Link Systems

Next, we extended our framework to promote cooperation in systems of rational peers who share their capacity-limited access link among their download and upload streams. These systems impose a different problem to solve, since the contributions of one peer would directly and negatively affect his performance, if no incentive mechanisms incurred. Towards solving this problem, we proposed a novel capacity adaptation algorithm based on which peers, being aware of the reputation system's existence, dynamically adapt the capacity that they dedicate for uploading and downloading in order to improve their utility. Our simulation studies have shown that rational peers are inclined to cooperation, while fairness is preserved, as peers receive resources in proportion to their contributions even in heterogeneous setups. Proposed framework outperforms existing work in this area in which the slowest link becomes the bottleneck of a heterogeneous system of different link capacity peers. In the contrary, no such bottleneck appears when our policies are used, improving the performance of the system even in more general systems where parallel downloads and uploads take place and peers with different request generation profiles exist.

BitTorrent

Following research targeted at popular file sharing system BitTorrent. An extensive investigation has indicated several deficiencies of the protocol. First of all, it lacks of an incentive mechanism for seeders to motivate them to remain and contribute in the system, although seeders' presence has been proved crucial for the performance of BT. On the other hand, BT's reciprocation policy fails to prevent unfairness across peers, as well as misbehavior, and, therefore, is vulnerable to strategic clients. We adapted our reputation-based framework over BT-like systems and exhibited via simulations that it can outperform the reciprocation mechanism of BT and preserve fairness among peers by guarantying that they can receive a proportional to their contributions amount of resources, while blocking misbehaviour. Furthermore, we presented how our policies can motivate seeders to remain in the system by gaining reputation during their contributions and exploiting it in their future requests.

It is also remarkable that the work related to BitTorrent and several improvements over it, assumes a clear distinction between the upload and the download capacity of the peers. The download capacity is much bigger than the upload capacity (as in ADSL connections), which is considered to be the bottleneck in the connections. If BT peers had to share a limited single-link capacity both for uploading and downloading data, they should decide how to allocate it between their uplink and downlink connections. Current implementations of BT do not support a capacity adaptation algorithm to help peers dynamically alter their download and upload capacities in order to improve their performance. The problem with that is that if all peers acted rationally, they would rather use their whole capacity solely for downloading and the system would collapse. In case of arbitrarily determining their contributions, our simulations have shown that in a loaded network, the average bandwidth peers receive per period is less than the half of their total physical capacity, while some peers receive more resources than their contributions and some less.

In order to improve BT performance under single capacity-limited link systems, we proposed two extensions over it. The one that cannot be hacked by a software is based on our capacity adaptation algorithm. We have seen that indeed the performance of the peers improve when our capacity adaptation algorithm is used in combination with the reciprocation mechanism of BT; however the performance gains are even more, if the capacity adaptation algorithm is combined with our reputation-based framework, instead.

Wireless Internet Sharing Communities

Finally, being attracted by the latest public interest on wireless communities and internet sharing, we decided to propose an incentive mechanism for such environments in order to establish cooperation among the users and motivate them to contribute their internet connection to mobile community members. The principles of the reputation-based framework that we have already designed guided our efforts, although we propose a different reputation metric, update mechanism and allocation policy to better comply with such systems.

Our goal was to design simply implementable reputation-based allocation policies for current wireless infrastructure, without the need of modifications to existing standards at the MAC or the network layer. This was a challenging task since the unreliable nature of the wireless medium complicates the QoS provision in wireless networks. Towards this goal, we proposed a reputation-based application, which we call R-HTB because it makes use of HTB tool, in order to control the amount of bandwidth offered to each competing client according to his reputation and estimated channel capacity. Our experimental results both in UTH Wireless Testbed and individual ORBIT nodes have shown that the throughput each user can achieve by connecting to community APs is proportional to his reputation for any channel conditions, type of traffic sent or received and number of competing clients.

Thereafter, we simulated a wireless community of users with different contribution and consumption profiles who roam inside it and connect to nearby community APs, while offering their own internet connection at home to community passersby through their own AP. Our simulations demonstrated that by succeeding throughputs in proportion to reputations for competing users, our reputation-based system lead users to cooperation. More specifically, it ensures that the more cooperative a user is by offering a good quality of internet connection through his home AP to community passersby, the better quality of service he can receive from his community, when being mobile inside it. Whats' more, the more greedy for community resources users are, the more contributive they should be to satisfy their needs under our reputation-based framework.

8.2 Future Directions

As we have already pointed out in this dissertation, there have been several recent research and commercial efforts towards wireless neighborhood communities which can provide free internet access service to its members. These communities can provide even other kind of services to its members, 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). It would be very challenging to fit in such kind of communities a reputation-based scheme 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.

In this thesis we adopted a local reputation system according to which each user keeps a local reputation for each other user of the system according to his past transactions with him. As we have already discussed and presented in this thesis, by keeping local reputations, users can distinguish and form coalitions with the most suitable for them partners, improving their performance. Moreover, load from communication exchanges due to recommendations and other possible problems like misreporting, reputation pollution and collusion, appearing in global reputation systems are avoided. On the other hand, when peers use recommendations/votes to determine one's reputation, they can speed up their perception of the p2p system. It would be very interesting to investigate an appropriate mechanism to differentiate users according to a combination of a local and a global reputation, in order for peers to maintain the ability of selecting and prioritizing suitable providers for their individual needs, while speeding up the cooperation in the community. However, it is critical to provide suitable mechanisms to guarantee the trustworthiness/sincerity of the peers who vote, which is a very difficult and demanding task, as we highlighted in the introduction.

Considering the internet sharing scenario studied in this thesis, clients in their association/ server selection decisions were ignorant of the channel conditions perceived by each sensed AP. We assumed that a mobile user can perceive the same channel conditions with each neighboring AP and either was associated to the nearest one (similar to 801.11 association policy) or, according to our reputation-based association policy, to one among the most reputed in a given distance range. A more sophisticated association policy could select the AP that has the best pair of reputation and channel conditions at the time of request, or according to a metric reflecting both the channel quality at a given time with a specific AP and the cooperation behavior of his owner. It would be very interesting to observe the performance gains of such a policy for wireless internet sharing users.

Bibliography

[1] Th. G. Papaioannou and G. D. Stamoulis. Reputation-based Policies that Provide the Right Incentives in Peer-to-Peer Environments. *Computer Networks (Special Issue on Management in Peer-to-Peer Systems: Trust, Reputation and Security), Elsevier*, 50(4):563–578, 2006.

[2] M. Mowbray *et al.* A Reciprocation-Based Economy for Multiple Services in Peer-to-Peer Grids. In *Proc. of the IEEE International Conference on Peer-to-Peer Computing*, pages 193–202, 2006.

[3] G. Hardin. The tragedy of the commons. *Science 162,*, pages 1243–1248, 1968.

[4] E. Adar and B.A. Huberman. Free Riding on Gnutella. *First Monday*, 5(10), 2000.

[5] D. Hughes, G. Coulson, and J. Walkerdine. Freeriding on Gnutella Revisited: the Bell Tolls. *IEEE Distributed Systems Online*, 6(6), June 2005.

[6] V. Vishnumurthy, S. Chandrakumar, and E. G. Sirer. 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*, 2003.

[7] R. T. B. Ma, S. C. M. Lee, J. C. S. Lui, and D. K. Y.Yau. Incentive and Service Differentiation in P2P Networks: A Game Theoretic Approach. *IEEE/ACM Transactions on Networking*, 5(14), 2006.

[8] M. Feldman, K. Lai, I. Stoica, and J. Chuang. Robust incentive techniques for peer-to-peer networks. In *Proc. of the 5th ACM Conference on Electronic commerce, ACM Press*, pages 102–111, 2004.

[9] H. Ma and H. Leung. 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 (AINA06), Vienna, Austria*, pages 1005–1010, April 2006.

[10] H. T. Kung and C. H. Wu. 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*, June 2003.

[11] N. Andrade, M. Mowbray, A. Lima, G. Wagner, and M. Ripeanu. Influences on Cooperation in BitTorrent Communities. *Proc. ACM SIGCOMM*, August 2005.

[12] L. Mekouar, Y.Iraqi, and R.Boutaba. A contribution-based service differentiation scheme for peer-to-peer systems. *Peer-to-Peer Networks and Applications journal, Springer*, 2(2), June 2009.

[13] A. Odlyzko. The case against micropayments. In *Proc. International Conference in Financial Cryptography*, 2003.

[14] C. Buragohain, D. Agrawal, and S. Suri. A Game Theoretic Framework for Incentives in P2P Systems. In *Proc. of the 3rd IEEE International Conference on Peer-to-Peer Computing*, 2003.

[15] S. D. Kamvar, M. T. Schlosser, and H. Garcia Molina. The EigenTrust algorithm for reputation management in P2P networks. *Proc. 12th ACM International Conference World Wide Web (WWW)*, pages 640–651, May 2003.

[16] S. Marti and H. Garcia-Molina. Limited Reputation Sharing in P2P Systems. In *Proc. of ACM Conference on Electronic Commerce*, pages 91–101, 2004.

[17] PeerTrust: Supporting Reputation-Based Trust for Peer-to-Peer Electronic Communities. *IEEE Transactions on Knowledge and Data Engineering*, 16(7):843–857, 2004.

[18] V. Pai and A. E. Mohr. Improving Robustness of Peer-to-Peer Streaming with Incentives. In *Proc. of the First Workshop on the Economics of Networked Systems*, June 2006.

[19] Y. Yan, A. El-Atawy, and E. Al-Shaer. Ranking-based Optimal Resource Allocation in Peer-to-Peer Networks. In *Proc. of IEEE INFOCOM*, 2007.

[20] D. Banerjee, S. Saha, S. Sen, and P. Dasgupta. Reciprocal Resource Sharing in P2P Environments. In *Proc. of the 4th International Conference on Autonomous Agents and Multi-Agent Systems*, pages 853–869, 2005.

[21] R. Mahajan, M. Rodrig, D. Wetherall, and J. Zahorjan. Sustaining Cooperation in Multi-hop Wireless Networks. In *Proc. of the Networked Systems Design and Implementation*, May 2005.

[22] Q. He, D. Wu, and P. Khosla. SORI: a secure and objective reputation-based incentive scheme for ad hoc networks. In *Proc. of the IEEE Wireless Communications and Neworking Conference*, March 2004.

[23] T.G. Papaioannou and G.D. Stamoulis. 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*, May 2005.

[24] M. Meulpolder, J. A. Pouwelse, D. H. J. Epema, and H. J. Sips. Bartercast: Fully distributed sharing-ratio enforcement in BitTorrent. *Technical report, Delft*, 2008.

[25] A.R. Bharambe, C. Herley, , and V.N. Padmanbhan. Analyzing and improving BitTorrent performance. *Technical Report MSR-TR-2005-03, Microsoft Research*, 2005.

[26] The Economist. Who wants to buy a computon?, March 2005.

[27] A. AuYoung, B. N. Chun, A. C. Snoeren, and A. Vahdat. Resource Allocation in Federated Distributed Computing Infrastructures. In *Proc. of OASIS '04*, October 2004.

[28] M. Feldman, K. Lai, , and L. Zhang. A Price-Anticipating Resource Allocation Mechanism for Distributed Shared Clusters. In *Proc. of the 6th ACM Conference on Electronic Commerce*, June 2005.

[29] http://www.planet-lab.org/.

[30] I. Foster, C. Kesselman, and S. Tuecke. The anatomy of the grid: Enabling scalable virtual organizations. *International Journal of SuperComputer Applications*, 15(3), 2001.

[31] K. Eger and U. Killat. Bandwidth Trading in Unstructured P2P Content Distribution Networks. In *Proc. of the 6th IEEE International Conference on Peer-to-Peer Computing*, 2006.

[32] A.Satsiou and L.Tassiulas. Dynamic Cooperation Enforcement through Trust-based Allocation Policies. In *Proc. of the 4th EURO-NGI International Conference on Next Generation Internet Networks*, April 2008.

[33] H. Ma and H. Leung. A Demand and Contribution Based Bandwidth Allocation Mechanism in P2P Networks: A Game-Theoretic Analysis. *Proc. 20th Interantional Conference Advanced Information Networking and Applications (AINA)*, pages 1005–1010, April 2006.

[34] M. Meo and F. Milan. A Rational Model for Service Rate Allocation in Peer-to-Peer Networks. In *Proc. of 8th. IEEE Global Internet Symposium*, March 2005.

[35] L. Guo, S. Chen, Z. Xiao, E. Tan, X. Ding, and X. Zhang. Measurements, analysis, and modeling of BitTorrent-like systems. *Proc. 5th ACM SIGCOMM*, October 2005.

[36] M. Piatek, T. Isdal, T. Anderson, A. Krishnamurthy, and A. Venkataramani. Do incentives build robustness in BitTorrent. *Proc. 4th USENIX Symposium Networked Systems Design and Implementation (NSDI)*, pages 1–14, 2007.

[37] S. Jun and M. Ahamad. Incentives in BitTorrent Induce FreeRiding. *Proc. ACM SIGCOMM Workshop on Economics of Peer-to-Peer Systems (P2PECON)*, August 2005.

[38] A. Sherman, J. Nieh, and C. Stein. FairTorrent:Bringing Fairness to Peer-to-Peer Systems. *Proc. ACM CoNEXT*, December 2009.

[39] M. Li, J. Yu, and J. Wu. Free-Riding on BitTorrent-like Peer-to-Peer File Sharing Systems: Modeling Analysis and Improvement. *Proc. IEEE Transactions on Parallel and Distributes Systems*, 19(7):954–966, July 2008.

[40] A. Legout, N. Liogkas, E.Kohler, and L.Zhang. Clustering and Sharing Incentives in BitTorrent Systems. *Proc. ACM SIGMETRICS*, 2007.

[41] J. H. Park, R. Chen, and X. Yang. Free-riding in BitTorrent Networks with the Large View Exploit. *Proc. Interntional Workshop on P2P systems (IPTPS)*, 2007.

[42] P. Shah and J-F. Paris. Incorporating Trust in the BitTorrent Protocol. *Proc. Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS)*, July 2007.

[43] http://en.fon.com/.

[44] NetShare, http://www.speakeasy.net/netshare/.

[45] Skyrove, http://www.skyrove.com/.

[46] P.Antoniadis, B.Le Grand, A.Satsiou, L.Tassiulas, R.Aguiar, J.Barraca, and S.Sargento. Community Building over Neighbourhood Wireless Mesh Networks. *IEEE Technology and Society, special issue on Potential and Limits of Cooperation in Wireless Communications*, 27(1):48–56, 2008.

[47] K. Choumas, T. Korakis, and Tassiulas L. New Prioritization Schemes for QoS Provisioning in 802.11 Wireless Networks. In *Proc. of the 16th IEEE Workshop on Local and Metropolitan Area Networks*, pages 25–30, September 2008.

[48] B. Hubert, T. Graf, G. Maxwell, R. van Mook, M. van Oosterhout, P. Schroeder, J. Spaans, and P. Larroy. Linux Advanced Routing Traffic Control HOWTO.

[49] A. Satsiou and L.Tassiulas. Trust-Based Exchange of Services to Motivate Cooperation in P2P Networks. *Accepted for publication in Peer-to-Peer Networking and Applications Journal, Springer.*

[50] A. Satsiou and L. Tassiulas. A Trust-based Exhange Framework for Multiples Services in P2P Systems. In *Proc. of the 7th IEEE Internatianal Conference on P2P Computing*, pages 45–52, September 2007.

[51] A.Satsiou and L.Tassiulas. Reputation-based Resource Allocation in P2P Systems of Rational Users. *IEEE Transactions on Parallel and Distributed Systems*, April 2010.

[52] A.Satsiou and L.Tassiulas. Reputation-based Internet Sharing in Wireless Neighborhood Community Networks. In *Proc. of the IEEE International Conference on Communications*, May 2010.

[53] G. Iosifidis and I. Koutsopoulos. Reputation-assisted utility maximization algorithms for peer-to-peer networks. In *Proc. of the 16th IEEE International Workshop on Quality of Service*, June 2008.

[54] G. Iosifidis and I. Koutsopoulos. Distributed resource allocation algorithms for peer-to-peer networks. In *Proc. of the 3rd International Conference on Performance Evaluation Methodologies and Tools*, October 2008.

[55] D. Qiu and R. Shrikant. Modeling and performance analysis of BitTorrent-like peer-to-peer networks. *Proc. ACM SIGCOMM*, September 2004.

[56] J. A. Pouwelse, P. Garbacki, D. H. J. Epema, and H. J. Sips. The BitTorrent P2P File-Sharing System:Measurement and Analysis. *Proc. International Workshop on P2P Systems (IPTPS)*, February 2005.

[57] B. Fan, D.-M. Chiu, and J. C. Lui. The Delicate Tradeoffs in BitTorrent-like File Sharing Protocol Design. *Proc. IEEE Interantional Conference on Network Protocols)*, November 2006.

[58] M . Izal, G. Urvoy-Keller, E.W. Biershak, P.A. Felber, A. Al Hamra, and L. Garces-Erice. Dissecting BitTorrent:Five Months in a Torrent's Lifetime. *Proc. Passive and Active Measurements*, 2004.

[59] B. Cohen. Incentives Build Robustness in Bittorrent. http://bittorrent.com/bittorrentecon.pdf 2003.

[60] A. Legout, G. Urvoy-Keller, and P. Michiardi. Rarest First and Choke Algorithms Are Enough. *Proc. Internet Measuremet Conference (IMC)*, 2006.

[61] N. Liogkas, R. Nelson, E. kohler, and L. Zhang. Exploiting BitTorrent For Fun (But Not Profit). *Proc. International Workshop on P2P systems (IPTPS)*, 2006.

[62] T. Locher, P. Moore, S. Schmidt, and R. Wattenhofer. Free Riding in BitTorrent is Cheap. *Proc. 5th Workshop Hot Topics in Networks (HotNets)*, November 2006.

[63] D. Carra, G. Neglia, and P. Michiardi. On the Impact of Greedy Strategies in BitTorrent Networks: The Case of BitTyrant. *Proc. 8th IEEE Internationla Conference Peer-to-Peer Computing (P2P)*, September 2008.

[64] W. Yang and N. Abu-Ghazaleh. GPS:a general Peer-to-Peer simulator and its use for modeling BT. *Proc. 13th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems*, 2005.

[65] K. Eger and U. Killat. Fair resource allocation in peer-to-peer networks. *Computer Communcications, Elsevier*, 30:3046–3054, 2007.

[66] N. Laoutaris, D. Carra, and P. Michiardi. Uplink Allocation Beyond Choke/Unchoke. *Proc. ACM CoNEXT*, December 2008.

[67] N. Ben Salem, J. P. Hubaux, and M. Jakobsson. Reputation-based Wi-Fi deployment. *Mobile Computing and Communications Review (MC2R)*, July 2005.

[68] E.C. Efstathiou, P.A. Frangoudis, and G.C. Polyzos. Stimulating Participation in Wireless Community Networks. *Proc. IEEE INFOCOM*, April 2006.

[69] E.C. Efstathiou. Public infrastructures for internet access in metropolitan areas. *Proc. 1st International Conference on Access networks*, 2006.

[70] B. Cohen. Incentives Build Robustness in Bittorrent. *http://bittorrent.com/bittorrentecon.pdf*, 2003.

[71] W. Cirne, F. V. Brasileiro abd J. Sauve, N. Andrade, D. Paranhos, E. L. Santos-Neto, R. Medeiros, and F. Silva. Grid computing for Bag-of-Tasks applications. In *Proc. 3rd IFIP Conference on E-Commerce, E-Business and E-Covernment*, 2003.

[72] S. Basu, S. Banerjee, P. Sharma, and S. J. Lee. NodeWiz: peer-to-peer resource discovery for grids. In *Proc. 5th IEEE International Symposium on Cluster Computing and the Grid*, 2005.

[73] *OurGrid*. http://www.ourgrid.org/.

[74] E. C. Efstathiou, P. A. Frangoudis, and G. C. Polyzos. Stimulating Participation in Wireless Community Networks. In *Proc. of IEEE INFOCOM*, 2006.

[75] B. Hubert *et al.* Linux Advanced Routing & Traffic Control HOWTO, Documentation from http://lartc.org/.

[76] D. Smith. Azureus Speed Scheduler. http://thesmithfam.org/blog/azureus-speed-scheduler/.

[77] UTH Wireless Testbed, http://inf-server.inf.uth.gr/ testbed/doku.php.

[78] A. Tirumala, F. Qin, J. Dugan, J. Ferguson, and K. Gibbs. Iperf, http://dast.nlanr.net/Projects/Iperf/.

[79] C. Bettstetter. Mobility modeling in wireless networks: categorization, smooth movement, and border effects. *ACM SIGMOBILE Mobile Computing Communication Review*, 5(3):55–66, 2001.