Rapport de Thèse Professionnelle

László Toka

09/2007

Partage de disque dur dans les réseaux pair-à-pair: étude de mécanismes d'incitation

Société: ENST Bretagne

Encadrant dans l'entreprise: Patrick Maillé

Encadrant académique: Pietro Michiardi

La thèse n'est pas confidentielle

Networking
Institut Eurécom

Abstract

Peer-to-peer sharing of hard disks: study of incentive mechanisms

During the spring semester 2007, I participated in a large scale project, called "Dispairse", in which I was assigned to work on a specific task. The project aimed to study the possibility of creating a backup service in a peer-to-peer network way: a ciphered copy of each user's data is stored on the hard drives of other participants in the network. Each participant is at the same time a service user and a service provider, she must therefore determine the proportion of her hard drive dedicated to store data from other users.

Such a service presents a lot of advantages: each user can access her data from any computer connected to the network, and the replication of data is a protection against failures. To study that kind of service, it is reasonable to consider that each user is selfish, i.e. she is only sensitive to the quality of service she experiences, regardless of the effects of her actions on the other users. The situation can be therefore studied as a non-cooperative game played among users, where a user strategy is choosing the amount of backup capacity used at other users and offered for them. Notice that as in peer-to-peer file sharing systems, there is no direct incentive for a user to provide a service to the others, which motivates the use of incentive mechanisms for the service to actually exist. Establishing such a mechanism was actually my job. I studied and compared several incentive mechanisms that had been suggested in the literature in other - but linked - contexts, like file sharing systems, connection sharing systems, and ad hoc networks. Some of those schemes rely on monetary incentives, and some others are based on service degradation for users who do not contribute enough to the service. Both types of schemes were considered, and I weighted their pros and cons for the particular context of the service in question. The criteria included economic measures (incentive properties, social welfare, cost of the mechanism) as well as practical issues (technical feasibility, computational complexity) in regard to the given technical constraints that needed to be satisfied to make the service feasible. I defined an economic model for the incentive game and then compared the selected schemes from an economic point of view analytically and through simulations. Based on the results, I drew interesting and important conclusions.

Partage de disque dur dans les réseaux pair-à-pair: étude de mécanismes d'incitation

Durant mon stage de fin d'études en 2007, j'ai participé dans un projet á grande échelle, sous le nom de "Dispairse", dans lequel j'étais affecté à un travail sur un détail spécifique. Le projet visait à étudier la possibilité de créer un service de protection des données d'une manière de réseau à pair-à-pair: une copie chiffrée des données de chaque utilisateur est stockée sur les disques dures d'autres participants dans le réseau. Chaque participant est un utilisateur et un fournisseur de service en même temps, elle doit donc déterminer la proportion de sa disque dure consacrée aux données d'autres utilisateurs.

Un tel service présente beaucoup d'avantages : chaque utilisateur peut accéder à ses données à partir de n'importe quel ordinateur relié au réseau, et les copies des données garantissent une protection contre des échecs. Pour étudier ce genre de service, il est raisonnable de considérer que chaque utilisateur est égoïste, c.-à-d. elle est seulement sensible à la qualité du service qu'elle recoit, indépendamment des effets de ses actions sur les autres utilisateurs. La situation peut être donc étudiée comme un jeu non-coopératif joué parmi des utilisateurs, où une stratégie d'utilisateur est de choisir la quantité de capacité de secours allouée chez les autres utilisateurs et celle qu'elle offre pour eux. Notez que comme dans le cas des systèmes pair-à-pair de partage de fichiers, il n'y a aucune incitation directe pour qu'un utilisateur fournisse un service aux autres, qui motive l'introduction des mécanismes d'incitations pour que le service existe réellement. L'établissement d'un tel mécanisme était ma tâche. J'ai étudié et ai comparé plusieurs mécanismes d'incitation qui avaient été suggérés dans la littérature dans d'autres - mais lié - contextes, comme le systèmes de partage de fichiers, de connexion, et les réseaux ad hoc. Certains de ces mécanismes se fondent sur des incitations monétaires, et quelques autres sont basés sur la dégradation de service pour les utilisateurs qui ne contribuent pas assez au service. Les deux types de mécanismes ont été considérés, et je les ai évalué dans le contexte particulier du service donné. Les critères ont inclus les mesures économiques (propriétés d'incitation, bien-être social, coût du mécanisme) aussi bien que les aspects pratiques (faisabilité technique, complexité informatique) en vue de les contraintes techniques données qui ont dû être satisfaites pour rendre le service faisable. J'ai défini un modèle pour le jeu d'incitation et puis ai comparé les mécanismes choisis d'un point de vue économique analytiquement et par des simulations. Basé sur les résultats, j'en suis arrivé à des conclusions intéressantes et importantes.

Contents

Introduction

With the convergence of fixed and mobile telecommunication systems, all kinds of digital documents are likely to be accessed by different types of devices. Therefore storing of all those documents raises several questions: should there be only one storing location? If so, what happens in case of a crash? If not, how to update documents between several locations? Will it be simple to transfer a document from a storing location to a given device?

The best way to address those problems is to implement a distributed storing system working in a peer-to-peer (p2p) way: using a p2p network infrastructure, a (ciphered) copy of each user's data is stored into the hard drives of other participants in the network. As in p2p file sharing networks, each participant is consequently at the same time a service user and a service provider. Such a service presents a lot of advantages in terms of reliability (data replication within the system provides a protection against failures) and ease of access (each user can access her data from any device connected to the network). Nevertheless, interesting studies on how users would behave in a p2p backup system should be carried out. We intent to investigate that particular issue.

Indeed, it seems reasonable to assume that each user is selfish, i.e. she is sensitive only to the quality of service she experiences, regardless of the effects of her actions on the other users. The framework of Game Theory [19] is therefore particularly well-suited to study that kind of situation: it can be studied as a non-cooperative game played among users, where a user strategy is the amount of memory capacity offered to provide service to the other users and the amount of data she stores into the system. Notice that as for other p2p applications, a user valuation for the service depends on the "generosity" of other users: each user benefits from others' shared capacity. However, there is no direct incentive to offer one's own capacity to the others, and users then intend to free-ride [1]. If the sharing efforts do not get some kind of proof of appreciation, nobody has interest to cooperate and the service cannot exist. Therefore it is necessary that some incentive mechanisms be properly designed for the service to actually exist and be valuable for users.

The existing literature on p2p backup systems mainly focuses on security, reliability and technical feasibility issues [13, 24], and the incentive aspect received little attention. The economic models developed for p2p file sharing systems do not apply to p2p backup services: in file sharing systems, when a peer provides some files to the community, it contributes to the whole system in terms of accessibility. This means that the resource is not dedicated to a certain number of users, but it is offered to all the rest of the peers, and in that sense the information stored in the p2p network is a public good. On the contrary, a p2p backup system operates on non-divisible resources, i.e. a certain disk space belongs to one given user (for the time being) and no other peer can access it. The stored resource available on the network is then a private good, and it cannot be managed the same way as a public good from an economic point of view.

The existing p2p backup service models focus on solutions that do not require many financial transactions. Therefore the counter payment for a given service is usually the service in question as well. This approach finally leads to a symmetric scheme where every peer should contribute to the system in terms of service at least as much as she benefits from others [10, 11].

We investigated more flexible solutions, that could still provide peers with incentives to contribute to the system and we focused their performance. We studied and compared two types of incentive mechanisms that have been suggested in the literature in other contexts. One of the schemes relies on monetary incentives, the other one is an exchange-based approach. Both types of schemes are considered, and their effects on the overall social welfare are weighted for the particular context of the p2p backup service.

I would like to thank Patrick Maillé for his extremely useful suggestions and help.

The next parts are organized as follows. A short, general description of the project which gives framework for this study is provided in Chapter 1. In Chapter 2 we summarize the main problems in p2p systems and several counter-measures addressing them. Chapter 3 presents the existing incentive solutions in the p2p domain. The model we consider for user preferences is depicted in Chapter 4, where we also study the maximum reachable value of social welfare yielded by the service. Also a strict symmetry-based scheme and schemes implementing pricing are investigated, moreover we present our test results. Chapter 5 presents our conclusions and directions for future work.

Chapter 1

The DisPairSe project

1.1 The project's goal

The goal of the DisPairSe project (French acronym for peer-to-peer safe disk) is to study a distributed file system service at the system level and in an economic approach while passing by the network and security aspects.

1.2 Storage space types

The backup application, of which feasibility the project aims to study, allows a user to have some storage space on the hard disks of other users belonging to the same community. This service can be stand-alone, independent of any operator or may be offered by an operator who would put the infrastructure of its own to offer the service. The backup of the file savings made by a user on her hard disk are also sent toward other disks on-the-fly. If facing a crash, the user will have the possibility of restoring her data. The storage capacity of a host can be broken up into four pieces:

- private capacity: only the user has access to the data and can modify it. The access can be carried out either from the local area network or from the outside (e.g. a mobile phone). This private disk part may contain confidential data.
- community capacity: the data is stored and shared between a defined community (group)
- public capacity: the data of this storage space is freely accessible
- backup capacity: this part makes backup possible for other users.

The storage capacities can be physically distributed on a pool of hard disks at the user. The implementation of the service must be simple to make it scalable, and has to provide data confidentiality at the same time. The users should have the possibility of defining a more or less large backup zone.

1.3 Overlay applications

Several possible application can be provided by this given service

- backup storage: A user can backup her data on other hard disks distributed on the network. In the case of disk breakdown, the data can be easily recovered.
- access to the contents of the hard disk from a mobile phone: A mobile network subscriber connects to her hard disk to gain access to data (email, photos, music, etc). The data can be adapted to allow updates on the mobile terminal.
- backing up photos: A mobile network subscriber uploads her photographs, taken by her mobile phone, to her hard disk.

1.4 Architecture issues

The users put several files "for backup" into the system to consider. The system will be charged to manage the access control of them according to attributes given by the user. As the number of users can be very large, the necessary backup capacity for these files seems to be too heavy for a centralized architecture: the files will be duplicated at storage capacities of users willing to contribute.

However, users can connect and disconnect at every moment. This dynamicity complicates the implementation of a centralized management of storage spaces, which is thus resource-demanding at each operation (e.g. connection, disconnection, file deposit, file removal, file modification, access to read). Consequently a completely decentralized architecture is considered, therefore autonomous and potentially accessible to an unlimited number of participants.

Among the existing approaches dealing with cooperative file storage in p2p networks ([6], [10], [22]), there is no real consensus on the most effective implementation of the system. As in p2p networks usually, a distributed solution (Gnutella-alike) can be placed in face of a "structured" solution (DHT-alike), each proposal showing defects and advantages.

In addition, network coding principles have a large interest in the project. They provide, in particular, low-cost guarantees of safety. Moreover, they make possible to decrease the number of copies required by the system to guarantee a certain level of quality of service. For these two reasons, the network coding was subject of some related work in the field of p2p networks [20]. Also in the DisPairSe project, this theory appears relevant.

1.5 Network issues

A simple functioning of the service can be achieved only by using the IPv6 protocol, thus making possible the deployment of a native or a tunneling infrastructure toward the end hosts.

Currently the networks, by the fact of the massive use of private IPv4 addressing, are built around the model client-server; the customers are placed inside of a private zone, generally a domestic or enterprise network and the servers are in the public space of the Internet. The current peer-to-peer (p2p) applications like Skype, break this model by integrating relatively complex mechanisms which allow outsiders to contact an internal equipment on a private local network. They can be regarded as precursors of services until IPv6 becomes more widespread in the networks. So that the contents of a hard disk would be accessible from the outside in IPv4, it is necessary that either the disk is located in the access box, or the NAT is configured to translate the requests going to a number of particular ports toward the user's private address. In the DisPairSe project, the users have the possibility of building hard disk farms to generate an income from this activity. It is very difficult to use a private addressing since the user would have to configure her NAT for each disk in her private network associated a number of different ports, which would make the localization of the equipment difficult.

IPv6 will simplify this model because in a quasi-unlimited address space all equipment will have a public address, making them directly reachable. Stability of the addresses in time will facilitate the localization.

The weak penetration level of IPv6 could be a problem, but solutions exist. The IETF developed a whole set of transition mechanisms allowing IPv6 usage on the top of IPv4 networks. The automatic mechanisms like 6to4 or TEREDO suffer from bad performances mainly because of the excessive tunnel lengths. Another approach, by the Softwires group, is under IETF standardization process. It expands the work of the RSM department at ENST-Bretagne (Point6box). The performance is nearly as good as direct access and it makes possible to start developing services using IPv6 (stable and known addresses, end-to-end, etc) without waiting for its wide spread implementation.

1.6 Security issues

When storing data in a distributed way, some confidential data (family photo album, payrolls, banking documents, etc) requires guarantees as for the respect of user's private life. Therefore a straightforward user identification and authentication mechanism is needed to control user accesses to the various storage capacities (private, group, public, archive) on their own or on others' hard disks. Even if the fixed network is relatively well studied, its integration with a 3rd generation mobile network has never been seen in details. Authentication issues arise when accessing the hard disks from mobile phones.

When dealing with the restoration of her hard disk, a user has to be able to locate all fragments of her data disseminated on distant hard disks, meanwhile preserving her anonymity, i.e. a particular fragment located on a given hard disk cannot be associated with her. On the other hand it is necessary to take care of the confidentiality of the fragments' content during their storage, as well as during up- and downloading through the network.

Again, recent techniques of network coding could be adapted to this type of application. The goal is to provide a solution that satisfies the anonymous localization of a user's fragments, the anonymity of a fragment's owner and the confidentiality of its contents. To achieve these requirements the system is based on the operator's or ISP's control.

Chapter 2

Economic frameworks

2.1 Economic issues in peer-to-peer systems

Users who join a p2p network have, in general, suboptimal incentives to contribute to the network. The result is an inefficient system where the overall levels of contribution are less than would be the case if each peer acted in the interests of the entire network of peers [4]. First, it is essential that peers be compliant with the protocol specification. Second, it is important that peers be available to the system. [17].

2.1.1 Free-riding

One phenomenon that should not be ignored is free riding. Some peers simply consume system resources but contribute nothing to the system. In particular, there is a *free-rider* problem in traditional p2p networks such as Napster: individual users are provided with no incentive for sharing their own files and thereby adding value to the network. Instead, such services currently rely upon altruistic behavior from their users. Studies of the Gnutella network have found that a very large proportion of its users contribute nothing to the system [1].

Uncontrolled or excessive free riding in a p2p network leads to network congestion at some hotspot peers and the degradation of system performance. It is thus important to design some mechanisms that encourage peers to contribute and reduce free riding behavior in the p2p networks. In [16] Feldman et al. develop a model to study the phenomenon of free-riding in p2p systems. At the heart of their model is a user of a certain type, an intrinsic and private parameter that reflects the user's willingness to contribute resources to the system. A user decides whether to contribute or free-ride based on how the current contribution cost in the system compares to her type. Asymmetric transactions create the possibility for asymmetry of interest.

Each peer anticipates that it has little effect on the total number of files shared when the network is large. Consequently the incentive to contribute nothing and rely only on the contributions of others grows with network size.

2.1.2 Tragedy of commons

Many p2p systems rely on cooperation among self-interested users. For example, users of file sharing systems who do not share their own resources cause long delays or download failures. When non-cooperative users benefit from free-riding on others' resources, the "tragedy of the commons" is inevitable. Avoiding this problem requires incentives for cooperation.

2.1.3 White-washers, large populations, high turnover, churn

A file sharing system can exceed a million simultaneous users, and nodes can have an average life-time of the order of minutes. Zero-cost identity in many p2p systems allows peers to continuously switch identities, so they are able to avoid possible penalties for not contributing to the system: the phenomena is called whitewashing and false trading. However, the enforcement of an incentive scheme is not straightforward in a realistic p2p system, where no trusted software or central entity accounting for peers' transactions can be assumed and peers are free to change their identity with no cost [5].

A possible proposed solution is imposing costs to newcomers in the memoryless system in order to discourage whitewhasing. [16] found that when penalty is imposed on all newcomers, system performance degrades significantly only when the turnover rate among users is high. [23] proposes mechanisms based on reputation to address the issue of false trading and other related attacks.

2.2 Proposed solutions

There exist proposed solutions for the mentioned issues in p2p networks. In fact, it is considered to be a hot topic in the research community. In this section we present the main approaches.

2.2.1 Distributed Algorithmic Mechanism Design

The Distributed Algorithmic Mechanism Design (DAMD) approach focuses on incentive compatibility solutions to Internet problems in a decentralized manner. An ideal solution would be a system design where nodes, acting selfishly, behave collectively to maximize the common welfare. When such a system has no centralized authority with total knowledge of the system making decisions, this becomes a DAMD problem. DAMD is a current area of study which combines computer science with incentive compatible mechanism design in the economics literature. It provides a useful framework for considering p2p systems and many researchers are currently studying this approach.

[14] discusses the challenges of DAMD in p2p systems and overlay networks with techniques like redundancy and cryptography. It inspects the notions of rationality and self-interest in p2p systems as well. Researchers also highlight some open problems in computational complexity of mechanisms and effects of mechanism design on network topology formation.

2.2.2 Micro-payments with clearance infrastructure

A natural approach is to charge users for every download and to reward them for every upload in file sharing p2p systems. In [21] Golle et al. made a first effort to model the utilities and costs associated with the participation in such a system using game theoretic analysis. They analyze the free riding problem and the equilibrium of user strategies under several micro-payment mechanisms.

[27] examines the problem of free riding in agent-based p2p systems, especially referral systems. The authors introduce two classes of micro-payment protocols and analyze the strategies of agents under these protocols in order to control free riding. The paper only provides a preliminary study of mechanism design in referral systems.

[17] puts emphasis on the requirements a payment-based scheme must fulfill. The authors state that the payment mechanism must provide anonymity and achieve high availability rate at the peers.

A major finding in the p2p file sharing systems related work is that a simple fixed fee scheme can achieve a high level of efficiency in large networks. This suggests that a complicated mechanism that relies on information reporting and complicated payments to peers is not necessary to run an efficient p2p network. Instead, peers need only contribute a standard number of files to the network on joining. There are two reasons why these results are relevant: First, many p2p networks are large, for example, the number of peers on Gnutella and Kazaa runs into the millions. Secondly, the results mean that it is not necessary to collect large amounts of information, or to undertake complicated calculations, in order to implement the correct incentives in a p2p network [4]. Of course the utility of the contributed files should be examined to prevent rewarding peers that share useless, unknown files.

2.2.3 Exchange, *tit-for-tat*

Pricing or micro-payment is only one incentive mechanism. Other mechanisms are also proposed to reduce the problem of free riding in p2p systems: non-priced incentives to encourage efficient behavior of p2p users. Some examples include delay time (e.g. users who share more content with the system have higher priority), network membership (e.g. removing non-sharing members from the systems), or peer ratings of content providers. However, empirical analysis is needed to measure the impact of these mechanisms.

In [5] a robust memory-less enforcement mechanism needs to rely only on the time a peer is consuming resources in the p2p system (namely, downloading a file) and thus use the uploading peer (the uploader) as the "enforcing entity". So they propose a system where the uploader first checks whether a candidate downloader shares the amount of files required before providing a requested file. Moreover, he should ensure that these files are accessible to the rest of the group (e.g. by checking with the search mechanism for their availability). However newcomers may often be able to get away with advertising invalid content for long enough to successfully complete their downloads.

The problem has been extensively studied adopting a game theoretic approach. The prisoners' dilemma model provides a natural framework to study the effectiveness of different strategies in establishing cooperation among players. In a simulation environment with many repeated games, persistent identities, and no collusion it is shown that the *tit-for-tat* strategy dominates [15].

2.2.4 Reputation

In reputation-based systems peers punish or reward each other based on observed behavior. Reputation mechanisms require the capability to evaluate a peer's cooperativeness on the basis of observable behavior and to provide a punishment or reward with the desired incentive. The reputation schemes assume a relatively small and static population of nodes, and are thus poorly suited to p2p systems (except for mobile ad-hoc networks where the number of peers usually does not reach high values).

2.3 User behavior under incentives in game theory

Among many other environment, p2p systems can also be seen as strategic games, in which each agent has a utility function over the possible actions. A rational agent is assumed to play a strategy which maximizes its own expected utility. In this section we describe the most important elements of game theory suited to p2p systems.

2.3.1 Rational users

All peers are assumed to be strategic. They are rational users who will change their behavior (evolve) to maximize their own benefit. In reality, there may be other types of individuals (altruistic, malicious, etc), but we concentrate on the strategic majority.

2.3.2 Information

Designing an optimal incentive scheme requires complete knowledge of the types and preferences of the individual peers and their identities [4]. To attain full efficiency, a network manager must have complete information about the payoffs of all peers who join the network, be able to offer each peer a personalized scheme, and where necessary, enforce (or make payments to ensure) participation. None of these conditions are likely to be achievable in practice.

As far as economic efficiency is concerned (i.e. maximization of social welfare), besides the complicated modeling, an additional challenging issue is the lack of information concerning the types and preferences of the individual peers, which is required for the computation of the optimal allocation of resources and cost in a p2p system. The existence of externalities rules out as inefficient the adoption of a free-market approach using market-defined prices for controlling a p2p system [5].

In addition to this challenging economic resource allocation problem, a p2p system designer usually has to deal with the inability to rely on trusted software or on central entities that can monitor and account for peers' transactions to ensure that they contribute and consume the amount of resources dictated by an underlying economic model.

2.3.3 Pay-off function for contribution

Detailed modeling of the economic transactions carried out in a file sharing system is, in general, a very complex task. The main reason is that participating peers should contribute different types of resources (bandwidth, storage, CPU cycles, content) with differing characteristics, the provision of which generates complex costs such as legal risks and time spent using the system [5].

In [4], [5] the authors attempt to define a simple model. After establishing the pay-off functions at the peers, the authors consider several possible mechanisms: first best rule where complete information is at hand, fixed contribution mechanism when the number of users gets large and exclusions are possible, a fixed contribution mechanism (e.g. each peer must contribute a uniform minimum number of files to join the network) suffices to lead the system to the second-best maximum efficiency that could be achieved under the 'feasibility', 'incentive compatibility' and 'individual rationality' constraints, and a fixed fee scheme.

2.3.4 Pay-off function for availability

[17] studies the use of payments to implement an incentive mechanism that attaches a real monetary cost to low availability of peers in a system. Through a game theoretic analysis, the authors evaluate the effectiveness of such an incentive, finding that peer availability can be significantly increased through the introduction of payments under certain conditions. They also demonstrate how a payment-based incentive that preserves anonymity can be implemented and integrated with a popular class of p2p anonymity systems by forwarding messages for each other. Their scheme borrows from work in anonymous digital cash and micro-payments to embed small payments for service in the messages exchanged in path-based p2p anonymity systems.

They introduce a relatively simple utility function that considers users to be subject to the following costs and rewards: a cost per unit of time for being joined to the system, a cost of using external funds to send messages anonymously, and the reward of sending messages anonymously.

Chapter 3

Related work

P2p systems must be designed to take participant incentives and rationalities into consideration and provide appropriate mechanism to ensure participants are fairly sharing their resources. We consider four different domains where incentives should be introduced to provide a functional system.

3.1 P2p file sharing

Performance of p2p resource sharing networks depends upon the level of cooperation of the participants. While most existing p2p architectures have assumed that participants are generally cooperative, there is growing evidence from widely deployed systems suggesting the opposite. The result of this non-cooperation can vary between tolerable service degradation and complete system collapse depending on design goals and performance requirements [3].

3.1.1 Systems without incentives

Napster handles searches through a centralized index server, while Gnutella uses broadcast queries. Both systems focus more on information retrieval than on publishing. Adar and Huberman [1] found that 70% of the Gnutella users did not share any content files and 90% did not answer to any queries from other peers. Furthermore, the most generous 1% of hosts serviced nearly half of all requests. Later studies of Gnutella and Napster confirm this general trend, though with a less dramatic fraction of non-cooperative nodes. They also found that almost one third of all Napster users under-reported their available bandwidth to avoid being selected as a site from which others would download files.

3.1.2 Basic incentives, credit-based systems

Some commercial p2p file sharing clients provide incentive mechanisms by applying simple system-specific rules to peers. In later versions of Kazaa, for example, the contribution of each peer is computed and, according to its level, peers have a corresponding priority in case of congestion. These mechanisms can be circumvented by a skilled user or by a competing company releasing a compatible client without the incentive restrictions.

A lightweight, two-way credit scheme is implemented in the eMule system. The goal of the credit system is to reward users contributing to the network by reducing their waiting time in the upload queue. The main advantage of this scheme is simplicity: there is no communication overhead and a peer only needs to maintain the upload and download volumes for each peer it has communicated with.

3.1.3 Payment-based systems

Payment-based incentive mechanisms have been proposed for various specific p2p systems with the intent of promoting compliance with the system protocol.

MojoNation was a deployed p2p system for robust file storage and retrieval in which peers traded a form of private currency called *mojo* in exchange for both the storage and retrieval of data. MojoNation network was the first known use of payment-based incentive mechanisms in p2p file sharing. The main limitation of this approach is that all transactions had to be cleared in a centralized system, and users were burdened with managing their *mojo*.

[21] proposed a simple game theoretic model of agent behavior in centralized p2p systems and shown that their model predicts free riding in the original Napster mechanism. The paper analyzes several different payment mechanisms designed to encourage file sharing in p2p systems.

A distributed cash-based scheme for p2p systems is presented in [26]. The system uses a currency called *karma* which is maintained for each user by a collection of random participants called a bank-set that is located using a DHT lookup. Users need to negotiate the price of serving an object through an auction mechanism, and coordinate with the bank-set for transferring karma between accounts. The result is a fully-fledged economic system that is in principle more flexible than an exchange system. In an idealized setting, a cash-based scheme may be able to offer a stronger performance advantage to contributing peers, as it is not subject to the “double coincidence of wants” constraint that drives exchanges.

In practice however, the cash-based approach has three main limitations. First, it suffers from all the complexities of currency management. If the mechanisms used to negotiate prices, adjust accounts to inflation and deflation and manage a user's budget are not made completely transparent to the user, then such a system is likely to have a high cost in terms of user attention which is suggested as a major reason why micro-payment schemes are unlikely to get wider acceptance in general. Second, the need to provide start-up funds to new users creates a potential loophole in the economy. Third, empirical evidence suggests that users strongly dislike micro-payments: having to decide before each download if a file is worth a few cents imposes mental decision costs. Users often prefer flat pricing plans, even when such mechanisms may increase their expected costs.

3.1.4 Exchange-based systems

To date, cash-based systems have seemed too complex, while lighter-weight credit mechanisms have not provided strong incentives for cooperation. In many real cases stricter, reciprocity-based approach is implemented.

BitTorrent is an example of a real world application focusing on bandwidth provisioning for content distribution, which actually implements a reciprocative incentive scheme without relying on past transactions of peers but on a direct exchange of resources [8]. Because the incentive scheme does not rely on tracking the long term behavior of peers it is simple to implement and largely immune to problems of false trading and whitewashing. However, because it is designed to incite only the provision of bandwidth, it does not tackle the objective of improving content availability.

[3] proposes also an exchange-based mechanism. The authors generalize the BitTorrents's reciprocity approach to N-way exchanges among rings of peers and present a search algorithm for locating such rings.

3.1.5 Reputation-based systems

[23] presents incentive techniques relying on reputation, and history records kept at peers. Incentive techniques that adapt to the behavior of strangers can cause systems to converge to complete cooperation despite the existence of zero-cost identities and without centralized identity allocation.

3.2 Ad-hoc networks

Self-configured or p2p ad-hoc networks require collaboration between nodes for successful data delivery. Forwarding packets of other nodes is an example of activity that requires collaboration. Additionally when mobility-assisted routing and custody transfers are necessary, as in the case of Delay Tolerant Networks, long-term storage is also required by the “nodes in the middle”. Consequently, an important problem in such a scenario is to provide incentives for collaboration among the participating entities. The incentive mechanisms are usually based on

- price or credit-based systems, where a node has to spend some “currency” to forward its own packets which can be collected by forwarding 3rd party traffic, or
- reputation systems that make sure that non-collaborating nodes are discovered and reported throughout the network.

3.2.1 Payment-based models

Payment-based mechanisms that require tokens or money to be exchanged in return for service [7], [28]. In [7] the authors use payments to drive the system to an equilibrium point where nodes comply with the protocol. Their approach requires all participating peers to have access to tamper-proof hardware to enforce honest exchange of payments. In [28], a payment scheme called Sprite is proposed: it uses a centralized record keeping authority along with a cryptographic scheme for deferred payments, and does not require specialized hardware. Both of these payment mechanisms do require the knowledge of the identity of the payer and thus cannot be used in an anonymity system.

3.2.2 Reputation-based models

Usually the heterogeneous nodes have no a-priori knowledge about other nodes, and they may come and go frequently, i.e. no long term connectivity can be supposed. Selfish nodes can collude with each other in order to maximize their welfare. Moreover some of the current systems depend on the broadcast nature of wireless networks in order to monitor other nodes. Such monitoring, however, may not always be possible (e.g. subject goes off the radar). Overall we can state that monitoring mobile nodes in a delay tolerant ad hoc network is not likely to be functional.

3.3 P2p backup system

Each node stores its replicated data on an independent set of remote nodes in the collective providing strong fault-tolerance, and agrees to accept data from other hosts in return. Nodes may enter and leave the collective without administrative action. Several projects have suggested the use of p2p routing and object storage systems as a substrate for backup, including Chord, Freenet and Pastry. File systems built on them, such as pStore, CFS and PAST, provide protection against machine failure.

3.3.1 pStore, CFS, PAST

[6] presents a secure distributed backup system: pStore exploits unused personal hard drive space attached to the Internet to provide the distributed redundancy needed for reliable and effective data backup. On top of this reliability, pStore includes support for file encryption, versioning, and secure sharing thus provides a user with the ability to securely backup files in, and restore files from a distributed network of untrusted peers. A quota policy where the amount of space available to a user for pStore backups is proportional to the amount of personal storage space contributed to the system.

In CFS [12], each storage node limits any individual peer to a fixed fraction of its space. These quotas are independent of the peer's space contribution. CFS uses IP addresses to identify peers, and requires peers to respond to a nonce message to confirm a request for storage, preventing simple forgery.

Trusted third parties can enforce quotas and certify the rights to consume storage but require centralized administration and a common domain of control. The PAST system [13] suggests that the same smart cards used for authentication could be used to maintain storage quotas. The smartcard is issued by a trusted third party, and contains a certified copy of the node's public key along with a storage quota. The quota could be determined based on the amount of storage the peer contributes, as verified by the certification authority. The smartcard is involved in each storage and reclamation event, and so can track the peer's usage over time.

In [24] Lillibridge et al. present a novel p2p backup technique. Each computer has a set of partner computers, which collectively hold its backup data. In return, it holds a part of each partner's backup data. By adding redundancy and distributing the backup data across many partners, a highly-reliable backup can be obtained in spite of the low reliability of the average Internet machine.

Farsite [2] aims to provide a very high degree of reliability by replicating data across PCs sharing a local area network. Farsite is a secure, scalable file system that logically functions as a centralized file server but is physically distributed among a set of untrusted computers. However, the authors left two main issues open: first, to prevent a single user from consuming all available space in the system, a mechanism to enforce per-user space quotas is needed. Second, Farsite relocates file replicas among machines according to the measured availability of those machines, which requires a mechanism to measure machine availability.

In [25] Ngan proposed a distributed accounting infrastructure. Each peer maintains a signed record of every data object it stores, and every object stored on its behalf elsewhere. Each node periodically chooses another random node to audit. If the auditor finds a node claiming an object it does not actually have, that node can be ejected from the system. The existence of a public key infrastructure and strong node identities is assumed. The mechanisms are fully decentralized and require no consensus or synchronization among the participating nodes.

3.3.2 Pastiche & Samsara

[10] presents Pastiche that exploits excess disk capacity to perform p2p backup with no administrative costs. Pastiche builds on three major developments to accomplish its goals: Pastry for p2p routing and node location, content-based indexing which provides flexible discovery of redundant data within similar files, and convergent encryption that allows hosts to use the same encrypted representation for common data without sharing keys. The backup data can be greatly compressed by choosing partners with similar software installations, most files being backed up will be already present on the partner, thus requiring no extra storage.

Pastiche needs a distributed quota enforcement mechanism; a node should occupy only as much space as it contributes. The authors came up with Samsara [11] that enforces fairness in p2p storage systems without requiring trusted third parties, symmetric storage relationships, monetary payment, or certified identities. Each peer that requests storage of another must agree to hold a claim in return - a placeholder that accounts for available space. After an exchange, each partner checks the other to ensure faithfulness.

Chapter 4

Model and simulation

4.1 DisPairSe project and technical constraints

This section presents our model which describes incentives for users to cooperate in a p2p backup system. We have to face two problems. On the one hand the model needs to "promote" those who offer storage space for other peers to backup files and "punish" those latter peers; on the other hand to provide the necessary availability for a backup service, incentives are needed to honor peers that are online almost all the time.

As for other p2p applications, a user values the service depending on the "generosity" of other users: each user benefits of others' shared capacity, but there is no direct incentive to force him to offer its own to the others. If the sharing efforts do not get some kind of proof of appreciation, nobody has interest to cooperate. It is necessary to implement coercive regulation, control or more flexible incentive mechanisms. Like the recent FON [18] initiative for WIFI accesses, such a mechanism will be able for example to allow a user to use storage space on the network for free if he offers a piece of his own in return, or to pay a certain amount of money for using the service without offering capacity in return, or to earn money by proposing large size storage. The question of balance point in such a service rises.

4.1.1 Public vs. private

We study a p2p backup system, where users offer some of their storage space to provide service for the others. The economic model for such a system is different from the ones applicable to p2p file sharing systems, since the storage capacity is a private good here. In file sharing p2p systems when a peer provides some files to the community, it contributes to the whole system in terms of accessibility. This means that the resource is not dedicated to a certain number of users, but it is offered to all the rest of the peers, at least in most of the cases. In that sense the information stored in the p2p network is a public good. On the contrary, a p2p backup system operates on non-divisible resources, i.e. a certain disk space belongs to one given user (for the time being) and no other peer can access it. It is more likely to resemble to a network bandwidth access sharing system, just on a long term basis. The stored resource available on the network is then a private good, and it cannot be managed the same way as a public good form an economic point of view.

4.1.2 Symmetric and payment-based

The existing models and systems focus on solutions that do not require many financial transactions, therefore the counter payment for a given service is usually the service in question as well. This approach finally leads to a symmetric scheme where every peer should contribute to the system in terms of service at least as much as she benefits at others. In a more flexible system users should have the possibility to act as pure consumers, or pure service providers which necessarily requires the introduction of monetary means, as ultimate reward for offering service. We study both incentive mechanisms: the price-based (also as revenue-driven monopoly) and the classic symmetric schemes. We compare the outcomes of such mechanisms to the socially optimal situation that could be attained if users were not selfish, and show that depending on user heterogeneity, a revenue maximizing monopoly can be a better way for the service to exist than a symmetric scheme in terms of social welfare. As the model is going to give ground to a commercial application, a payment based model and the presence of a central authority or clearance service is welcome.

4.1.3 Availability

In a p2p backup system the availability of the stored data is considered as the most important factor in user's appreciation. If one backups her files at other peers and pays for this service, she requires to be able to fetch her files at any time. Therefore we need to provide extremely high availability in the system. As the backup disks are users' property, there is no direct means to guarantee that a given backup disk storing a specific file will be online at almost 100% of the time. To ensure availability for those who rely on the service, the model introduces several tools, such as data network coding and replication. From the economic point of view, our model has to deal not just with the need for incentives for offering storage space, but also with the lack of means to force a peer to be/stay online. In the model we suppose that if a user goes offline for a relatively long time, when she comes back online new data load will be transferred into her offered storage space independently what and whose data she was storing before.

4.1.4 Replication

Note that in the following, we say that the amount of storage space that is necessary to safely store some data in the system equals the size of those data. This is done without loss of generality: assume that the system introduces a redundancy factor r to improve the data availability on the system, then this is equivalent to replacing C^s, the amount of data to be backuped, by $C^s * r$ in the user benefit and cost functions. So we do not care about the number of copies the user wants to backup her files at to gain more security in terms of availability. On the contrary the system will maintain and advertise the minimum guarantee of a backup copy's availability. The user will decide what level of availability she wants and then she will determine the number of copies she should backup her files. The minimum guarantee comes from the central authority who keeps record of every user in the system, manages administration, authentication, accounting.

Other common points in the mentioned related works are security and anonymity. The implemented system must provide peer authentication, encryption, anonymity for the users.

4.2 User preferences, payoff function

In a game theoretical framework, which is ideal to model a selfish, non-cooperative environment, the preference of a user $i \in \mathcal{I}$ in the user set denoted by $\mathcal{I}$ can be described by a payoff function (which provides the perceived utility). In our specific case the function consists the benefit of using the backup service by storing an amount C_i^s of data in the system, the cost of offering storage space C_i^o for other users, and the monetary transactions (the total charge ϵ_i should be a decreasing function of C_i^o and an increasing function of C_i^s), if any. We suggest to use a separable additive function of the utility perceived by a user i, as described in the following definition.

Definition 1 *The utility U_i of a user $i \in \mathcal{I}$ is of the form*

$$U_i\left(C_i^s, C_i^o, \epsilon_i\right) = V_i(C_i^s) - P_i(C_i^o) - \epsilon_i, \tag{4.1}$$

where

- *$V_i(C_i^s)$ is user i's valuation of the service, i.e. the price she is willing to pay to store an amount C_i^s of data in the system. We will assume that $V_i(\cdot)$ is positive, continuously differentiable, increasing and concave in its argument, and that $V_i(0) = 0$ (no service yields no value).*
- *$P_i(C_i^o)$ is the overall non-monetary cost of user i for offering capacity C_i to the system with a probability π_i to be found online to provide service ($C_i^o = C_i\pi_i$). It consists of several elements:*
  - *opportunity cost of storage capacity when giving it away for other users (just for the time being online)*
  - *re-entry cost because of high traffic and hardware utilization when coming back online after some time and new data load arrives for storing*
  - *background cost, caused by the churn generated data traffic in which the specific user is involved.*
- *ϵ_i is the monetary payment for the taken service (the remuneration for offering decreases it). In case of symmetric scheme, this term is 0.*

Now we can define the terms of the overall cost function $P_i(C_i^o)$:

$$P_i(C_i^o) = O_i\left(C_i^o\right) + C_i\delta_i \frac{1}{\frac{1}{\mu_{off}^i} + \frac{1}{\mu_{on}^i}} + C_i^o k_i \bar{\mu} \sum_j C_j^o, \tag{4.2}$$

where

- O_i is the opportunity cost of storage capacity when giving it away for other users (just for the time being online). We assume that $O_i(\cdot)$ is positive, continuously differentiable, increasing and strictly convex, and that $O_i(0) = 0$ (no contribution brings no cost).
- μ_{on}^i and μ_{off}^i are the peer's characteristics which give her rate of going offline and coming back online respectively. We assume here that the spent time in each status is a stochastic variable and follows exponential distribution with these parameters.

- π_i is the average online status of the given peer i (the probability that she is available) in the storage offering point of view, moreover $\pi_i = \frac{\mu^i_{off}}{\mu^i_{off}+\mu^i_{on}}$.

- δ_i is a user constant in the re-entry unit cost term which represents non-monetary costs due to high traffic when a peer comes back to the system (CPU, bandwidth utilization, etc.). If a peer is offline for at least a given time period, after showing up again, the operator will arrange new data loads to her storage capacity (if offering any) no matter what she was storing before she left, so this term is inversely proportional to the average time period between two arrivals. δ_i should reflect of course, the user characteristics such as access bandwidth, hardware profile, etc.

- The background cost terms consists of k_i, a coefficient which describes the user's part of the common cost created by the community's churn (usual peer status changes generate large traffic of exchanging backup data); $\bar{\mu}$ that represents the weighted (by the offered capacity) average of overall rate of peer status changes (both peer departures and arrivals generate data traffic); and the sum of offered storage spaces which determines the amount of traffic. This term appears only at those peers who offer storage space (proportionally to their offered capacity), and only during the time they are online.

Considering that $C_i = \frac{C^o_i}{\pi_i} = C^o_i \frac{\mu^i_{off}+\mu^i_{on}}{\mu^i_{off}}$

$$P_i(C^o_i) = O_i\left(C^o_i\right) + C^o_i\left(\delta_i\mu^i_{on} + k_i\bar{\mu}\sum_j C^o_j\right). \tag{4.3}$$

4.3 User strategies

A user can choose her own strategy by varying the following parameters:

- The amount of data C^s_i to be stored in the system.

- The quantity of disk space to offer to the other system users C^o_i.

- The proportion of time that user i is online π_i. We may assume that users can choose when to be online, and when to disconnect by turning one's device off or by moving to an unconnected area.

A user will choose the strategy which maximizes her utility function. One can raise her own payoff by increasing C^s_i until the point where the unit price (in a payment based scheme) or the unit exchange cost (in a symmetric scheme) reaches the marginal benefit. Same for C_i^o, but in this case it is the marginal cost which approaches to the remuneration per bit. Making an effort to stay online gives the opportunity to earn money actually by offering storage space, so keeping π_i high is also a positive factor in the user's payoff.

Differentiating the payoff function in μ_{on}^i gives us negative value which means that with lower μ_{on}^i the valuation will be higher. But still there is a constraint that is not in the user's power: no matter how much a user wants to stay online (to decrease her μ_{on}^i), unexpected events (user habits, power black-outs, accidents, hardware failures, etc) may force the user off the network. So we introduce a user specific parameter $\mu_{on_{min}}^i$ to describe the exogenous constraint the user faces. Let us suppose therefore that the user, by trying to minimize her μ_{on}^i, chooses $\mu_{on_{min}}^i$.

Assumption A *For every user i there exist a low threshold $\mu_{on_{min}}^i$ and the user's μ_{on}^i can not fall under this level. To maximize her revenue, the user will choose this value as her μ_{on}^i, so it becomes a user parameter instead of a strategic variable.*

4.4 Assumptions

To carry out a deeper analysis in the next sections, we need to make some assumptions regarding the exact valuation and cost functions.

Assumption B *We assume quadratic functions for the valuation and cost functions:*

$$V_i(C_i^s) = \frac{1}{b_i}\left(-\frac{\left(C_i^s \wedge b_i p_{max}^i\right)^2}{2} + b_i p_{max}^i \left(C_i^s \wedge b_i p_{max}^i\right)\right) \tag{4.4}$$

$$P_i(C_i^o) = \frac{1}{a_i}\frac{C_i^{o2}}{2} + p_{min}^i C_i^o \tag{4.5}$$

where a_i, b_i, p_{max}^i, p_{min}^i are user specific constants and $\wedge$ denotes the min. *To be conform with the model's cost terms, in the assumed cost function $p_{min}^i = \delta_i \mu_{onmin}^i + k_i \bar{\mu} \sum_j C_j^o$.*

Notice that the same kind of assumption (i.e. same form of utility functions for all users) is made in [9] in the framework of a p2p file sharing system, for user valuation functions. From the valuation and cost functions, we can be derive (by differentiation and taking the inverse functions) two other functions.

Definition 2 *For a user $i \in \mathcal{I}$, we call* demand function *(resp.* supply function*) the function $d_i(\cdot)$ (resp. $s_i(\cdot)$) such that for all $p \in \mathbb{R}_+$,*

$$d_i(p) := \begin{cases} (V_i')^{-1}(p) & \text{if } p \leq V_i'(0), \\ 0 & \text{otherwise.} \end{cases} \tag{4.6}$$

$$s_i(p) := \begin{cases} (P_i')^{-1}(p) & \text{if } p < \lim_{q\to+\infty} P_i'(q), \\ +\infty & \text{otherwise,} \end{cases} \tag{4.7}$$

where f' stands for the derivative function of function f.

For a given $p \geq 0$, $d_i(p)$ (resp. $s_i(p)$) is the amount of storage capacity that user i would choose to use from (resp. to offer to) the others if she is charged (resp. paid) a unit price p for it.

Remark that, as intuitively expected, the demand (resp. supply) function is nonnegative and decreasing (resp. increasing) in the unit price. Under Assumption B and based on Definition 2 the nonnegative, increasing linear-type demand function d_i and the nonnegative, increasing linear-type supply function s_i for user $i \in \mathcal{I}$ satisfy the following properties.

- $d_i(0) > 0$, then d_i is strictly decreasing until p^i_{max} while it takes strictly positive values, then $d_i(p) = 0$ for $p \geq p^i_{max}$.
- $s_i(p) = 0$ if $p \in [0, p^i_{min}]$, then s_i is strictly increasing

Under Assumption B, Definition 2 gives the demand and supply functions of a user i:

$$d_i(p) = b_i[p^i_{max} - p]^+, \tag{4.8}$$

$$s_i(p) = a_i(p - p^i_{min}). \tag{4.9}$$

On Figure 4.1 we show the sums of the supply and demand functions for all users. The sum of the supply functions is not linear on the $[\min_i(p^i_{min}), \max_i(p^i_{min})]$ interval, same is true for the demand functions on $[\min_i(p^i_{max}), \max_i(p^i_{max})]$.

4.5 Optimal social welfare

In this section, we introduce the social welfare, which will be used later to compare the performance of different incentive schemes.

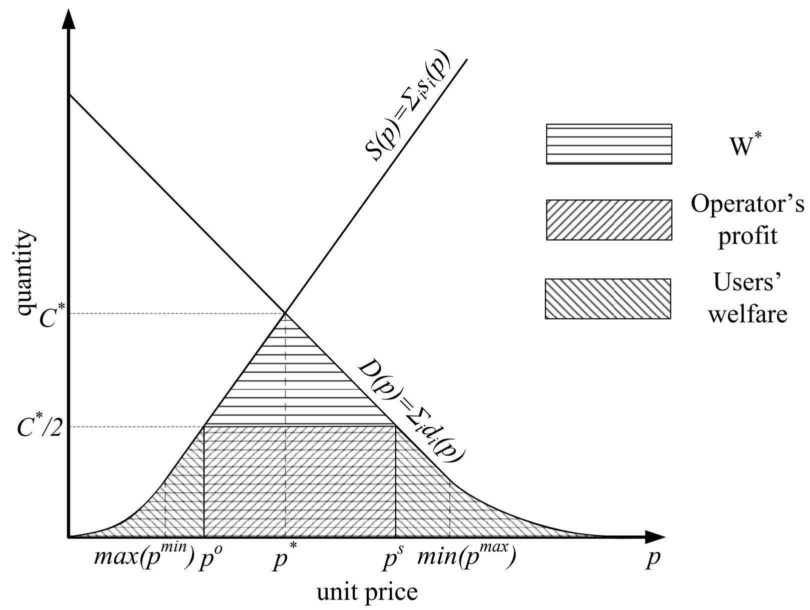

Figure 4.1: The sums of supply and demand functions

Definition 3 *We call* social welfare *(or welfare) and denote by W the sum of the utilities of all agents in the system:*

$$W := \sum_i V_i(C_i^s) - P_i(C_i^o). \tag{4.10}$$

Notice that no prices appear in (4.10). This is because even if we consider a payment-based incentive scheme, we choose to include in social welfare all system agents, eventually including the entity that receives (or gives) payments. The utility of this entity would be its revenue, and all money it exchanges with the users would stay within the system and therefore does not influence social welfare.

In this subsection we seek the "optimal" situation that the system can attain (in terms of social welfare maximization). The problem expresses

$$\max_{C_i^s, C_i^o}(W), \tag{4.11}$$

subject to $C_i^o \geq 0$, $C_i^s \geq 0$ for $\forall i$ and $\sum_i C_i^o \geq \sum_i C_i^s$ necessary feasibility conditions.

This is a classical convex optimization problem, that can be solved by the Lagrangian method.

$$\mathcal{L} = \sum_i V_i(C_i^s) - P_i(C_i^o) + \lambda \left[\sum_i (C_i^o - C_i^s) \right]$$

First order conditions for $\forall i$ imply that:

$$\frac{\partial \mathcal{L}}{\partial C_i^o} = 0 \Rightarrow P_i^{'}(C_i^o) = p$$

$$\frac{\partial \mathcal{L}}{\partial C_i^s} = 0 \Rightarrow V_i^{'}(C_i^s) = p,$$

where $p \geq 0$ is the Lagrange multiplier relative to the feasibility constraint. Complementary slackness condition writes:

$$min\left[p, \sum_i (C_i^o - C_i^s)\right] = 0 \Rightarrow \sum_i (C_i^o - C_i^s) = 0,$$

because p must be strictly positive, otherwise the first order conditions give $C_i^o = 0$ and $C_i^s > 0$ for all i, violating the feasibility constraint. We therefore obtain

$$C_i^s = d_i\left(p^*\right), \qquad C_i^o = s_i\left(p^*\right), \tag{4.12}$$

where p^* is the (unique) solution of

$$\sum_i \left(s_i\left(p^*\right) - d_i\left(p^*\right)\right) = 0, \tag{4.13}$$

and the optimal value of the social welfare is then

$$W^* = \sum_{i \in \mathcal{I}} V_i(d_i(p^*)) - P_i(s_i(p^*)). \tag{4.14}$$

Under Assumption B, by substituting to 4.8 and 4.9, the optimal social welfare is

$$W^* = \frac{1}{2}\sum_i b_i\left({p_{max}^i}^2 - p^{*2}\right) - a_i\left(p^{*2} - {p_{min}^i}^2\right). \tag{4.15}$$

Figure 4.1 gives a graphical interpretation of the maximum social welfare that can be attained by the system, where the optimal social welfare is graphically represented by the area bounded by the sums of demand and supply functions. Notice that the Lagrange multiplier can be interpreted as a unit price: if users buy the resource at unit price p^*, and sell their available disk capacity at the same unit price, then the selfish user decisions drive the system to the welfare maximizing solution. Note that these functions (i.e. $d_i\left(p\right), s_i\left(p\right)$) represent the maximum payoff providing storage space values corresponding to a given unit price for each user. If these functions were in the operator's possession in the real life, most of the transient convergence time leading to the equilibrium could be dropped.

We therefore have a characterization of the optimal solution. However, as pointed out in the introduction, user selfishness does not lead to this optimal situation when users are not incentivized to offer service to the others. Actually, the unique Nash equilibrium of the game without incentives corresponds to the situation where $C_i^o = 0$ for all i, and the associated social welfare is 0.

In the rest of the paper, we investigate two kinds of incentive schemes, and study their performance in terms of social welfare. We first consider mechanisms without pricing, that simply impose users to provide at least as much memory space as the amount they intend to use (most of the existing related works support this kind of fairness providing approach). Then we turn to payment-based incentive mechanisms where users have to pay for using the service and are paid if they contribute. We finally compare the outcomes of those schemes in terms of social welfare, for some particular types of valuation and cost functions. Since for the two schemes under study the optimal situation cannot be reached in general, we measure the loss of welfare of those schemes with respect to the maximum value.

4.6 Comparison of symmetric and payment based schemes

In this section we compare the outcomes of implementing symmetric and payment based schemes in terms of social welfare under the Assumption B. First we focus on a symmetric scheme where no payment based incentives are necessary. Most of the existing related works support this kind of fairness providing approach. Then we turn to payment based incentives where users have to pay for using the service and may get paid if they contribute.To have a manageable heterogeneous user preference set, we have to make an other assumption regarding p_{min} and p_{max} .

Assumption C *For further analysis we assume that the following conditions are fulfilled regarding to the user set $\mathcal{I}$:*

$$\min_i(p^i_{max}) \geq \frac{p^* + \max_i(p^i_{max})}{2} \tag{4.16}$$

$$\max_i(p^i_{min}) \leq \frac{p^* + \min_i(p^i_{min})}{2}, \tag{4.17}$$

where $p^* = \frac{\sum a_i p^i_{min} + b_i p^i_{max}}{\sum a_i + b_i}$*. Moreover user profile values* a_i *(resp.* b_i*) of all users* $i \in \mathcal{I}$ *are independent and identically distributed, and* a_i *and* b_i *are independent.*

4.6.1 Symmetric scheme

As evoked in the introduction, the principle of the symmetric schemes is that users are invited to contribute to, at least as much as they take from, the others. So we consider that each user i chooses C^o_i and C^s_i so as to maximize $V_i(C^s_i) - P_i(C^o_i)$, subject to $C^o_i \geq C^s_i$ (ϵ_i being 0 in this case). As $P_i(\cdot)$ is increasing in C^o_i, no user has an interest to choose a strategy with $C^o_i > C^s_i$. Therefore a user will necessarily choose $C^o_i = C^s_i$. User i maximizes her utility[1] at the point $C^s_i = C^o_i = C^*_i$ where

$$V'_i(C^*_i) - P'_i(C^*_i) = 0. \tag{4.18}$$

So if the symmetric scheme is implemented, every user will maximize her utility "exchanging" $C^*_i = \frac{a_i b_i}{a_i + b_i}(p^i_{max} - p^i_{min})$ storage capacity at the *virtual unit price* $p^*_i = \frac{a_i p^i_{min} + b_i p^i_{max}}{a_i + b_i}$, so the social welfare is going to be

$$W_{sym} = \frac{1}{2} \sum_i \frac{a_i b_i}{a_i + b_i} (p^i_{max} - p^i_{min})^2. \tag{4.19}$$

Also under the Assumption B, we can express the difference between the social welfare levels of the optimal and the symmetric schemes. Each and every user loses $\frac{1}{2}(p^* - p^*_i)^2(a_i + b_i)$ benefit relatively to the optimal case, assuming that $\max_i(p^i_{min}) \leq p^* \leq \min_i(p^i_{max})$ which comes from Assumption C. Altogether the social welfare loss of the system is

$$W^* - W_{sym} = \sum_i \frac{a_i + b_i}{2} (p^* - p^*_i)^2. \tag{4.20}$$

4.6.2 Payment based scheme

To be able to define the user strategies in a payment based scheme, we have to know more about ϵ_i. Of course, it is supposed to be increasing regarding C^s_i, and decreasing as C^o_i grows. We consider the following very simple system: contributors are paid p^o per unit of storage capacity they offer to the system, and service users are charged a unit price p^s when they store their data onto the system. The amount

[1] Actually, the utility maximization problem for a user is a convex problem, that has the same form as the social welfare maximization problem studied in subsection 4.5, except that there are only two decisions variables (C^o_i and C^s_i) here.

that user i will be charged (this amount can be negative, in which case the user gets paid) is consequently

$$\epsilon_i = p^s C_i^s - p^o C_i^o. \tag{4.21}$$

The constraints of a payment based scheme are more strict than the ones we presented when seeking the optimal outcome. Here, the system wants to have the required storing capacity online (availability), and the operator does not intend to loose money (feasibility). To summarize, we end up at the formula below.

$$\frac{p_s}{p_o} \sum_i C_s^i \geq \sum_i C_o^i \geq \sum_i C_s^i \tag{4.22}$$

We analyze the model as a full information game, i.e. we assume that the entity that operates the service (the operator) has perfect knowledge of the users and their valuation and cost functions. Therefore, knowing that users will act so as to maximize their utility, it can predict user reactions, and drive the outcome of the game to the most profitable situation for itself. In this sense, the operator acts as the leader of a Stackelberg (or leader-follower) game [19]. We investigate two possibilities: either the coordinator aims at maximizing the user surplus, or it is a revenue-driven monopoly that chooses prices so as to maximize its revenue.

Welfare-Maximizing Operator

The theoretically highest level of social welfare (in subsection 4.5) can be attained by a payment based scheme. In fact it is reached when the selling and buying prices are the same and equal p^* (see (4.13)). In this case the operator monetary surplus is null: the operator has no income at all and just acts as a coordinator that redistributes money among users.

Profit-Oriented Monopoly

In this subsection, we assume that the monopoly strives to extract the maximum profit out of the business. The operator therefore faces the following maximization problem.

$$\max_{p^s, p^o} \left(p^s \sum_i d_i\left(p^s\right) - p^o \sum_i s_i\left(p^o\right) \right), \tag{4.23}$$

subject to $p^s \geq 0$, $p^o \geq 0$ and the feasibility constraint that writes $\sum_i s_i(p^o) \geq \sum_i d_i(p^s)$.

This problem is hard to solve for general utility functions, and even under Assumption B since it is not a convex problem. We therefore consider the case where Assumption C holds.

Proposition 1 *Under Assumptions B and C (which is sufficient, but not necessary condition) the performance ratio of the social welfare for a profit-maximizing monopoly (W_{mon}) to the maximum social welfare (W^*) reaches a relative level:* $\frac{W_{mon}}{W^*} \geq \frac{3}{4}$.

Proof A *If p^o and p^s unit prices fall on the linear part of $S(p)$ and $D(p)$ respectively, the profit maximization problem becomes a convex problem, and even a simple graphical argument is enough to conclude. Figure 4.1 plots two curves: the total supply $S = \sum_i s_i$ and the total demand $D = \sum_i d_i$ as functions of the unit price p. First remark that p^o and p^s must be chosen such that $S(p^o) = D(p^s)$: otherwise it is always possible for the operator to decrease p^o (if $S(p^o) > D(p^s)$) or increase p^s (if $S(p^o) < D(p^s)$) to strictly improve its revenue. The operator revenue with such prices is then the area of the rectangle displayed in Figure 4.1, embedded within a triangle whose area (with the non-linear tails) is the maximum value of social welfare. The area of the rectangle reaches its maximum when $S(p^o) = D(p^s) = C^*/2$ with $C^* = S(p^*) = D(p^*)$ and p^* given in (4.13). In that case the operator's profit is more than $W^*/2$, and the total social welfare is more than $3W^*/4$.*
Here we show that under the assumptions, the monopoly prices cannot fall on the non-linear parts of S and D. We give an upper bond to the non-linear part covered area by linking the points $(\min_i(p^i_{min}), 0)$ and $(\max_i(p^i_{min}), C(\max_i(p^i_{min})))$ of the supply function, and the points $(\max_i(p^i_{max}), 0)$ and $(\min_i(p^i_{max}), C(\min_i(p^i_{max})))$ of the demand function. By prolonging these sections up to the vertical line of p^*, the maximal area rectangular in each triangle lies between $\frac{\min_i(p^i_{min})+p^*}{2}$ ($\frac{\max_i(p^i_{max}+p^*}{2}$ respectively) and p^*. So the conditions must stand not to have the possible monopoly unit prices on the non-linear tails. Once we have the monopoly prices and quantity fallen on the linear parts, we can state that the social welfares' ratio is greater than 75%.

With a profit oriented scheme the highest reachable social welfare shows the following loss to the optimal case:

$$W^* - W_{mon} = \frac{1}{8}C(W^*)^2 \left(\frac{1}{\sum_i a_i} + \frac{1}{\sum_i b_i} \right) \tag{4.24}$$

if Assumption C is fulfilled, by expressing the difference (caused by halving the overall storage capacity) between W^* and W_{mon}.

To give necessary but not sufficient conditions to have the profit oriented scheme's operating prices out of the non-linear intervals, we state that $p^o \geq \max_i(p^i_{min})$ and $p^s \leq \min_i(p^i_{max})$ have to stand. The profit oriented payment based scheme's social welfare reaches its maximum at $C(W_{mon}) = \frac{C(W^*)}{2}$ on a triangle-type figure. If the supposition is fulfilled, we have the same characteristics even with the non-linear additional tails. The conditions are derived from $C(W_{mon}) = \frac{1}{2}\sum_i a_i(p^* - p^i_{min}) = \frac{1}{2}\sum_i b_i(p^i_{max} - p^*)$ to have $Q(SW_{mon}) \geq S(\max_i(p^i_{min}))$ and $Q(SW_{mon}) \geq D(\min_i(p^i_{max}))$:

$$\min_i(p^i_{max}) \geq \frac{p^* + \frac{\sum_i b_i p^i_{max}}{\sum_i b_i}}{2} \tag{4.25}$$

$$\max_i(p^i_{min}) \leq \frac{p^* + \frac{\sum_i a_i p^i_{min}}{\sum_i a_i}}{2}. \tag{4.26}$$

4.6.3 Comparison

In this subsection we compare the social welfare outcomes of the two practical schemes, i.e symmetric and payment based schemes, more precisely the losses of social welfare in symmetry based (4.20) and the profit oriented (4.24) schemes.

Proposition 2 *Under Assumptions A, B and C, the social welfare of a profit oriented scheme exceeds the symmetric scheme's if*

$$\frac{1}{4}\left(p^* - \sum_i \alpha_i p^i_{min}\right)\left(\sum_i \beta_i p^i_{max} - p^*\right) \leq \sum_i \omega_i \left(p^*_i - p^*\right)^2$$

stands.

Proof B *Focusing on the diminution in each case, to have a higher social welfare in the profit oriented scheme we get to the following inequality:*

$$\frac{1}{4}C(W^*)^2\left(\frac{1}{\sum_i a_i} + \frac{1}{\sum_i b_i}\right) \leq \sum_i (a_i + b_i)\left(p^* - p^*_i\right)^2,$$

which can be developed to the proposition's form by introducing $\alpha_i = \frac{a_i}{\sum_i a_i}$, $\beta_i = \frac{b_i}{\sum_i b_i}$ *and* $\omega_i = \frac{a_i + b_i}{\sum_i a_i + b_i}$. *Since* $C(W^*) = \sum_i a_i(p^* - p^i_{min}) = \sum_i b_i(p^i_{max} - p^*)$ *the proposition is straightforward. Moreover* $p^* = \sum_i \omega_i p^*_i$, *so we can state that the inequality includes the product of the relative distance of* p_{min}*'s and* p_{max}*'s weighted* *average from p^* on the left side, and a weighted variance-type expression of p_i^* on the right side.*

4.7 Special cases

To carry out a deeper analysis in this section, we will assume that the user heterogeneity is limited to a_i and b_i in the first case, and to p^i_{min} and p^i_{max} in the second case.

4.7.1 Homogeneous price thresholds

Assumption D *We suppose that the supply (resp. demand) functions are of the same form for all users, and only differ through the multiplicative constant, i.e. a_i (resp. b_i). For the sake of simplicity, we assume that $p^i_{min} = 0$ and $p^i_{max} = p_{max}$ for all users.*

Under Assumptions D we get the social welfare at the ideally fine-tuned unit price from (4.13):

$$p^* = p_{max} \frac{\sum_i b_i}{\sum_i (a_i + b_i)}. \tag{4.27}$$

Therefore with C_i^s and C_i^o from 4.8 and 4.9 respectively, after some simplifications Assumption B gives the social welfare:

$$W^* = \frac{1}{2} p_{max}^2 \frac{\sum_i a_i \sum_i b_i}{\sum_i (a_i + b_i)}. \tag{4.28}$$

Under our specific assumptions on demand and supply functions, the value of social welfare for both investigated schemes are derived in the followings.

Proposition 3 *Under Assumptions B and D, the ratio of the social welfare for the symmetric scheme W_{sym} to the maximum social welfare W^* is*

$$\frac{W_{sym}}{W^*} = \left(\frac{1}{\sum_i a_i} + \frac{1}{\sum_i b_i} \right) \sum_i \left[\frac{1}{\frac{1}{a_i} + \frac{1}{b_i}} \right]. \tag{4.29}$$

This ratio converges as the number of users tends to infinity

$$\frac{W_{sym}}{W^*} \xrightarrow[|\mathcal{I}| \to \infty]{} \left(\frac{1}{\mathbb{E}[a_i]} + \frac{1}{\mathbb{E}[b_i]} \right) \mathbb{E} \left[\frac{1}{\frac{1}{a_i} + \frac{1}{b_i}} \right] \tag{4.30}$$

Proof C *We straightforwardly obtain that for all i, $C_i^o = C_i^s = p_{max}\frac{a_i b_i}{a_i+b_i}$. The corresponding social welfare is then $W_{sym} = \frac{1}{2}p_{max}^2 \sum_{i\in\mathcal{I}} \frac{a_i b_i}{a_i+b_i}$, and (4.29) then comes from (4.28). The law of large numbers gives (4.30).*

As $f : x, y \mapsto \frac{1}{\frac{1}{x}+\frac{1}{y}}$ is strictly concave, Jensen's inequality implies that $W_{opt} \geq W_{sym}$, and that equality stands if and only if a_i, b_i are deterministic, i.e. identical for every user.

Proposition 4 *Under Assumptions B and D the performance ratio of the social welfare for a profit-maximizing monopoly W_{mon} to the maximum social welfare W^* is*

$$\frac{W_{mon}}{W^*} = \frac{3}{4}. \tag{4.31}$$

Proof D *Same proof as for Proposition 1. Without the non-linear tails here, the ratio is exactly $\frac{3}{4}$.*

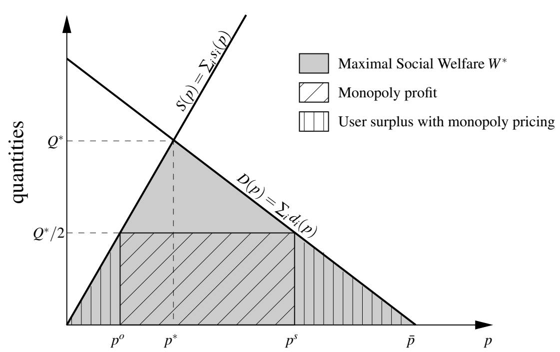

Figure 4.2: Optimal choices for a revenue-driven monopoly under Assumptions B and D.

Also, from Proposition 2 under Assumption D we have

$$\frac{1}{4}\frac{\sum_i a_i \sum_i b_i}{\sum_i (a_i+b_i)} \leq \sum_i (a_i+b_i)\left(\frac{b_i}{a_i+b_i} - \frac{\sum_i b_i}{\sum_i (a_i+b_i)}\right)^2, \tag{4.32}$$

since $p^* = p_{max}\frac{\sum_i b_i}{\sum_i (a_i+b_i)}$. After some simplifications we get to

$$\frac{1}{4}\frac{\sum_i a_i \sum_i b_i}{\sum_i (a_i+b_i)} + \frac{(\sum_i b_i)^2}{\sum_i (a_i+b_i)} \leq \sum_i \frac{b_i^2}{a_i+b_i}, \tag{4.33}$$

where the right side is equal to $\sum_i b_i - \sum_i \frac{1}{\frac{1}{a_1}+\frac{1}{b_i}}$, so at the end

$$\left(\frac{1}{\sum_i a_i} + \frac{1}{\sum_i b_i}\right)\sum_i \frac{1}{\frac{1}{a_1}+\frac{1}{b_i}} \leq \frac{3}{4}, \tag{4.34}$$

which means that the profit oriented scheme's social welfare is higher if this inequality stands.

This result highlights in particular the fact that if the population is homogeneous (i.e., a and b are Dirac distributions) then it is better to implement a scheme based on symmetry, since the maximal social welfare can be attained (from Jensen's equality case). On the contrary, user heterogeneity in terms of a and b will make the left-hand side of (4.34) decrease. If heterogeneity is too important then the left-hand side of (4.34) may take values below $3/4$, which implies that the system (users+coordinator) is better off being driven by a revenue-maximizing monopoly.

At this point of the analysis the distribution of a_i and b_i turns out to be the main characteristics of the game. Indeed, these parameters characterize the profile of each user, i.e. her utility and cost functions, and her associated demand and supply functions $(d_i(p), s_i(p))$ when p_{min} and p_{max} are fixed. In the following we consider two simple examples of distributions (uniform and exponential) for a_i and b_i to illustrate (4.34).

Uniform Distribution

We assume here that a (resp. b) is uniformly distributed over $[0, a_{\max}]$ (resp. $[0, b_{\max}]$). In that case we have $\mathbb{E}[a] = \frac{a_{\max}}{2}$, $\mathbb{E}[b] = \frac{b_{\max}}{2}$, and

$$\mathbb{E}\left[\frac{1}{\frac{1}{a}+\frac{1}{b}}\right] = \frac{1}{3}\left(a_{\max} + b_{\max} - \frac{a_{\max}^2}{b_{\max}}\ln(1+\frac{b_{\max}}{a_{\max}}) - \frac{b_{\max}^2}{a_{\max}}\ln(1+\frac{a_{\max}}{b_{\max}})\right). \tag{4.35}$$

The left-hand side of (4.34) and the plane $z = 3/4$ are displayed on Figure 4.3 *(left)*. We observe that inequality (4.34) *never* holds, thus it is always better for the system to impose symmetry than to introduce a profit-maximizing monopoly.

Exponential Distribution

We now consider the case where a (resp. b) follows an exponential distributions with parameter μ_a (resp. μ_b). Therefore $\mathbb{E}[a] = \frac{1}{\mu_a}$, $\mathbb{E}[b] = \frac{1}{\mu_b}$, and after some calculation we obtain

$$\mathbb{E}\left[\frac{1}{\frac{1}{a}+\frac{1}{b}}\right] = \begin{cases} \frac{1}{3\mu_a} & \text{if } \mu_a = \mu_b, \\ \frac{{\mu_a}^2 - {\mu_b}^2 - 2\mu_a\mu_b \ln(\frac{\mu_a}{\mu_b})}{(\mu_a-\mu_b)^3} & \text{otherwise.} \end{cases} \tag{4.36}$$

We again compare both terms of (4.34) on Figure 4.3 *(right)*.

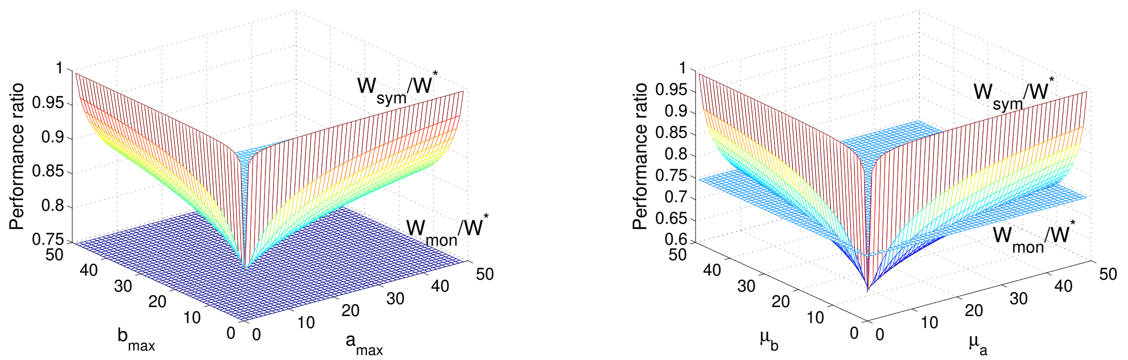

Figure 4.3: Social welfare performance ratio of monopoly pricing *(plane surfaces)* and symmetric scheme *(curved surfaces)* as given by Propositions 3 and 4, for uniform *(left)* and exponential *(right)* distributions of a and b.

This time, there is no scheme that always outperforms the other: depending on how much the two variables differ, the monopoly can provide higher welfare than a symmetric scheme. More precisely, when μ_a and μ_b are sufficiently close, then a revenue-maximizing monopoly will drive the system to a situation where the social welfare is larger than what a symmetry-based scheme would have yielded. Figure 4.4 indicates the scheme that gives the best system welfare, depending on the values of μ_a and μ_b.

4.7.2 Homogeneous price sensitivity

In Subsection 4.7.1 we evaluated a special case of the extended model, where p^i_{min} and p^i_{max} were uniform on the user set. Now we look into the case where these two variables are user specific, but a_i and b_i are the same for every user (a and b respectively).

Assumption E *We suppose that the supply (resp. demand) functions have the same slope for all users, i.e.* $a_i = a$ *(resp.* $b_i = b$*) for* $\forall i$.

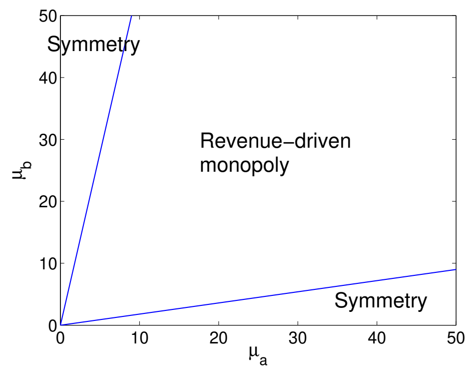

Figure 4.4: The higher social welfare performance resulting scheme for exponential distributions of a and b (μ_a and μ_b are the parameters of the exponential laws).

Proposition 5 *Under Assumptions B, C and E, always the symmetric scheme leads to higher social welfare.*

Proof E *In Proposition 2, α_i, β_i and ω_i become $\frac{1}{n}$, where n represents the number of users, moreover $p^* = \frac{a\tilde{p_{min}}+b\tilde{p_{max}}}{a+b}$, where $\tilde{p_{max}} = \frac{\sum_i p^i_{max}}{n}$ and $\tilde{p_{min}} = \frac{\sum_i p^i_{min}}{n}$. So for higher outcome in the profit oriented scheme,*

$$\frac{ab}{4}\left(\tilde{p_{max}} - \tilde{p_{min}}\right)^2 \leq a^2 var(p_{min}) + b^2 var(p_{max}),$$

must stand, since p_{min} and p_{max} are considered to be independent (under the given constraints) therefore the variance expression can be separated. We know that

$$var(p_{min}) \leq \frac{(p^* - \max(p_{min}))^2}{4} \leq \frac{(p^* - \tilde{p_{min}})^2}{4}$$

and

$$var(p_{max}) \leq \frac{(\min(p_{max}) - p^*)^2}{4} \leq \frac{(\tilde{p_{max}} - p^*)^2}{4}$$

from Assumption C. We get directly by replacing p^* that even with the high limit of the variances

$$(a+b)^2 \leq 2ab$$

must stand to have higher social welfare with profit oriented scheme, which is a contradiction.

4.8 Dynamic unit payment for storage offering

Even with the non-monetary punishment the user faces if often going offline the system needs more inciting means to reduce churn. One solution is to increase the paid unit price for the offering entities in the evolution of their uptime. In this section we present the main idea, and a short analysis how it changes our model. Let us assume that a given user who offers storage capacity for other members, will receive the counter payment from the system in the function of the time she has *continuously* spent in the system. This kind of appreciation values more a user who does not switch often her status than a user who introduces a high churn to the system while their overall π_is are same. We picked an exponential function to describe the unit remuneration, so the monetary transaction for a time unit is described as follows:

$$\epsilon_i(t_{on}^i) = p^s C_i^s - p^o \left(1 - e^{-\frac{t_{on}^i}{T}}\right) C_i, \tag{4.37}$$

where t_{on}^i is the uptime since the user came back online for the last time and T is an operator-tuned overall system variable.

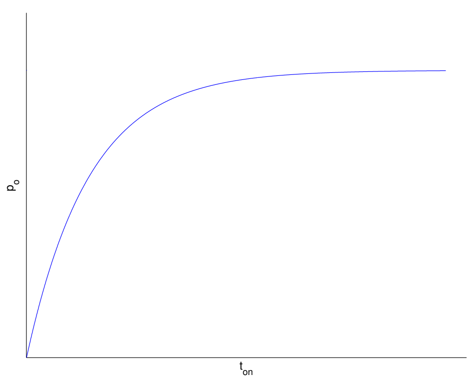

Figure 4.5: Increasing unit remuneration function

Like previously, we assume that each peer's behavior can be described by μ_{on}^i and μ_{off}^i stochastic variables which follow exponential distribution. Provided this assumption, determining the expected average unit remuneration for a given user is straightforward. The average earning per time unit is calculated by the following conditional expected value integral, and it shows the earning for the time being on-and offline:

$$E(\epsilon_i^o) = \frac{\int_0^\infty p^o \left[t_{on}^i + T\left(e^{-\frac{t_{on}^i}{T}} - 1\right)\right] \mu_{on}^i e^{-\mu_{on}^i t_{on}^i} dt_{on}}{\frac{1}{\mu_{on}^i} + \frac{1}{\mu_{off}^i}}. \tag{4.38}$$

The result is

$$E(\epsilon_i^o) = \frac{\mu_{off}^i}{\mu_{on}^i + \mu_{off}^i} \frac{p^o}{1 + T\mu_{on}^i}, \tag{4.39}$$

so

$$\epsilon_i = p^s C_i^s - \frac{p^o}{1 + T\mu_{on}^i} C_i^o, \tag{4.40}$$

because $C_i^o = \frac{\mu_{off}^i}{\mu_{on}^i + \mu_{off}^i} C_i$.

If this kind of remuneration is introduced, the fact that the user is incited to decrease her μ_{on}^i is not changed. Nevertheless, this kind of appreciation values more a user who does not switch often her status than a user who introduces a high churn to the system while their overall π_is are same.

4.9 Simulation

We created a simple simulation to study the functioning of the model. The users have different profiles, they may go offline from time to time, and than return online. The time is divided in rounds in which users make up their new strategies, and the operator defines new prices for the following round. By choosing a relatively short time interval for a round (e.g. 1 second), the simulation approaches to a continuous representation.

The operator's task is to determine the p_s, p_o prices to have the maximum profit at the equilibrium point. By varying the prices of storage, the operator forces the users to change their strategies to fully adapt to the new market. As the operator examines the feedback, it will lead the system towards an optimal equilibrium finally. As it is supposed to wear realistic attributes and thus no full information is available for the provider, the path to the equilibrium is carried out by converging to stability with small changes in price.

4.9.1 Implemented model

The implemented model is the same as the one we presented in the previous sections. The user variables $(a_i, b_i, p_{min}^i, p_{max}^i)$ are drawn randomly from uniform distribution normalized to the unit interval $[0,1]$. The randomly generated μ^i_{on}, μ^i_{off} parameters are picked from $[0,0.1]$ and $[0,10]$ respectively, to describe a relatively well-connected online backup community. The users vary their C^s_i, C^o_i to fully adapt to their market environment, i.e. to maximize their payoff, while the operator gives its *strategic variables*, the defined prices p^s, p^o.

4.9.2 Implementation issues

During the implementation in MATLAB, we had to face some practical issues which can be encountered even in a real life application.

- What should be the initial unit price/remuneration (both p^s and p^o) for the unit storage space during a time unit? - We started the simulation with $p = p^s = p^o$ at an average level (as every monetary parameter is drawn uniformly on the $[0,1]$ interval, we chose 0.5) and moved up and down these prices along the rounds according to the changes in offer-demand. However to determine the initial price without any a priori knowledge is not possible and a wrong choice can prolong the convergence time seriously.

- Basically the simulation is not a full information game, i.e. the operator does not know every user's preferences. Thus the system needs to converge to the equilibrium point by moving the prices appropriately. When offer goes up, the operator cuts prices, on the other hand when demand exceeds the available storage space, prices are increased. Avoiding to restrict the possibilities, in each case a heuristic deciding mechanism chooses the price to change, i.e. only one of p^s and p^o is altered at a time to adjust the situation to the availability and feasibility constraints. Evidently tuning the price causes strategy changes at users which again are realized in new C^s_i, C^o_i decisions in the following round and thus forces the operator to suit prices to the new situation again. This kind of interaction leads to iterations in the convergence periods even without churn (users leaving the system and coming back).

- Besides the effort of keeping the constraints fulfilled, the operator has an even more important motivation to change p^s and p^o. In the previous analysis, we saw that by expanding the gap between these two prices, and thus altering more and more from the optimal (social welfare) price p^*, the operator's profit can be increased (until a certain limit). Since local maximums may exist, seeking the best profit is not feasible with straightforward methods. Therefore we implemented a randomized successive optimal seeker algorithm with retro-correction, which tries (heuristically) to improve the operator's income by taking larger steps (but less often than the constraint observer algorithm) in the price tuning.

- It is more likely to be a technical implementation issue, but of course has an effect on the incentive model as well: if a backup peer took off, after how much time do we need to start data restoration? The backup service is carried out by the storage space offering peers and an organizational unit which role is played by the multi-functional operator. When a peer leaves the system (temporarily or definitely), the data management center can not know the time this given peer remains offline. So the question rises: assuming that every user's data is replicated in at least 2 copies in the system, when should we propagate the remaining copy to an other backup peer to ensure a continuous service? In our approach we suggest a policy of data restoration beginning right away at the moment a backup peer is detected to be gone. A worst case scenario study is required: during the gray zone time (where the departed peer is not detected yet) and the transfer time, the other backup peer may leave too which poses problems in case of only two copies. Here one has to make a compromise between the number of copies and the remuneration of backuping. If the files are saved at several peers, the probability of being able to retrieve them increases, but the backup service is more costly (due to the increased amount of backup).

- How to get the status of each peer? Again this question is a more technical one. As there is a central agent in the system for administration and authentication purposes, it is easily feasible to enroll the new clients through an initial period for observing their usual behavior, even testing them if they can be approved for backuping service. After a test period, when a client has already become a member, other clients in the system, or the operator itself, can verify time to time if she is online, e.g. by pinging, and really stores the backup load she claims. In this manner the central authority easily obtains data about the users.

4.9.3 Results

In Appendix A.1 we provide the MATLAB code we wrote to run the simulations. With the given comments inside the code should be easy to understand. We carried out a number of tests where we examined several scenarios by varying the parameters. In this section we briefly present the results. The main achievement is that in all cases the system converges towards an equilibrium, and noticeably toward a state with the possible and stable highest profit. In Appendix A.2 we present the graphs issued at the end of the different scenarios.

On Figure A.1 a number of randomly chosen users' availability traces are represented. We supposed an implemented initiative test period, where a probable online status profile is assigned to every newcomer peer, based on the observation during a relatively short period. Of course this gives just an estimate, and especially in the early period of time spent in the system, the real value of the user's availability will differ from it, but after a while, passing through a temporary phase to let the operator gather enough information about the new peers, the characteristic π_i becomes relatively stable. In a given case where an important change in a given user's behavior occurs, it can be considered as the final departure of the peer associated to the user, and the arrival of a new one, characterized by the user's new profile.

On the further simulation figures we present the profit oriented operator's prices, the optimal scheme's price and in some cases (where it is possible to determine) the maximum profit providing prices. Moreover we also present figures where we plot the different social welfares in a profit oriented system: utility of offering and of backuping (sum of the utility function of each peer in the system), their sum as user welfare, the sum of user welfare and operator profit as social welfare and at last but not least the optimal scheme's and symmetric scheme's social welfares for comparison. It is interesting to see how the storage capacities vary during the convergence time, so we provide figures on this issue as well. At the end of each case study, we show the evolution of the provider's profit.

General model

First we run simulations on the general model where we considered p^i_{min}, p^i_{max}, a_i and b_i randomly generated user parameters to be uniformly distributed and normalized. We considered a profit-oriented operator managed system with growing population (initial number of users is 100 which goes up to 1000 later), and also depicted the optimal and symmetric scheme's social welfares for comparison. The first case in the group is a fully general case, while the second one assumes no peer departs at all to create a fluctuation-free environment, and the third one is based on the general model respecting Assumption C.

On Figure A.2 one can see the evolution of the profit-oriented provider's prices. After a while when the initial prices are converging toward the maximum profit providing state, a relative price stability is observed. The fluctuations are caused by the peer departures, arrivals, re-arrivals, strategy changes. Sure, for a real life implemented system this kind of fluctuation should and can be expelled by applying a buffer of storage space and a margin of money circulating in the system to handle the relatively small variation over the time. In the second case where we excluded the peer departures, the fluctuation is less important. In the third case (with the constraints regarding to p^i_{min} and p^i_{max}) the rising p^s and the falling p^o do not differ from the optimal price (in the middle) as much as in the two previous cases.

Figure A.3 shows the social welfares in the given three cases. Naturally the optimal social welfare is the highest, the profit-oriented scheme's and the symmetric scheme's social welfares are below its level (outliers may have been generated by violating the constraints for short transient periods). In all cases the user social welfare drops and the profit grows to have a profit-oriented social welfare level similar to the symmetric case's social welfare at the end. In the third case the welfares are slightly less then in the first two cases because of the constraints on user parameters. In a symmetric scheme it is hard to handle user departures for a functional backup system, so we assumed really high online rates to make a comparison, therefore the it gives a flat line on the plot.

We represented the capacities on Figure A.4, where a dramatical drop (especially in the fist two cases) is seen in order to achieve the maximum profit out of the system which is plotted on Figure A.5. The third case seems to be special regarding the important fluctuation in the amount of shared capacities (on a higher level than the first two cases) and the profit.

Homogeneous price threshold model

The Figures A.6, A.7, A.8 contain three cases, each of them is a homogeneous price threshold scenario ($p^i_{min} = 0$ and $p^i_{max} = p_{max}$ $\forall i$). In the first case we run a 100000-round simulation to show the convergence state in details, the second case is more focused on the "static" state, the third one is just to show a scenario with properly tuned initial prices. The graphs of each case are placed respectively from top to bottom.

First regarding to the changing prices (p^o and p^s), the easily obtained (based on full information) optimal scheme price (p^*) and the maximum profit providing prices ($\max p^o$ and $\max p^s$) on Figure A.6, we can state that the final "stable" prices do not depend on the population dynamics, as new peers not just offer but also take storage capacity so the demand-offer ratio is balanced on average. If the initial prices are adjusted to the optimal scheme's price level, a long term convergence period can be seen through the rounds. On the second and third figure then we can state that if once the prices reached their ideal level (in terms of profit maximizing), they tend to stick around it in the converged state.

More interesting information can be seen on Figure A.7 where we traced the utilities for users, operator and for different schemes. Naturally all three cases end up in nearly identical states, the slight difference is caused by the randomly generated user preferences. When the prices start on the optimal scheme's price level, during the convergence period the user welfares drop dramatically, but the increasing profit keeps the social welfare on a relatively high level. Depending on the user heterogeneity, this level may rise above the symmetric scheme's social welfare.

Also on Figure A.8, where the sums of the taken and the offered capacities are shown, one can observe the same characteristics, i.e. the storage capacity is halved while getting from the nearly optimal initial to the profit oriented state. There is a little difference between the similar end results of the first, second and the third case.

On Figure A.9 we present the operator's profit in the three cases. Is is shown that the evolution of profit is unbreakable, and reaches the highest possible value in the final stable state.

Chapter 5

Conclusion

5.1 Summary

In the presented work we addressed the problem of user incitation in a p2p backup system. We have shown a game theoretical model which describes realistically user preferences, and their behavior under the two main existing incentive frameworks, i.e. symmetric and profit oriented payment based schemes. Not only the size of the offered storage space was targeted with incentives, but as the availability and reliability is a particularly important issue in backup systems, the model also aimed to reduce churn. By comparing the level of social welfare as outcome in the two cases, we arrived to the important conclusion: the profit oriented payment based incentive scheme may be better than the symmetric one under specific circumstances, namely if the heterogeneity among the user profiles is high. The performance measure we considered is the total social welfare. We derived a necessary and sufficient condition for the monopoly management to outperform the symmetry-based system management for a case where the peers' utility functions for offering memory space and storing data onto the system have given (quadratic) forms. We investigated two special situations where the user set's heterogeneity was limited and showed that the profit oriented scheme still can be more attractive in terms of social welfare.

5.2 Results

In the master thesis, after having carefully studied the related existing works, we propose an economic model for a p2p backup service. Assuming that users of such a service behave selfishly, we have justified the need for incentive schemes for the system to effectively exist. We have studied and compared two kinds of incentive schemes, namely a symmetry-based scheme (where users should contribute as much as they use the service) and a pricing-based scheme (introduction of a monopoly that fixes unit prices for selling and buying resource). Under some simplifying assumptions, we have highlighted conditions for one scheme to outperform the other in terms of social welfare. Some examples during the analytical evaluation have shown that user heterogeneity plays a crucial role on the better-suited scheme. Basically, it seems that more user heterogeneity would justify the use of profit-driven pricing.

For the practical realization, we have run simulation tests to evaluate the economic model, especially focusing on the profit maximizing approach, since the optimal and symmetric cases are straightforward to achieve. The results seem to fulfill our expectations as the system converges towards the state of equilibrium with a maximal value of profit. Thus the model could give ground to an excellent functional commercial p2p backup system implementation.

5.3 Future work

Still there remains a lot of work to be done on the subject. First, we have made quite restrictive assumptions on the form of the utility functions to derive some results (we used quadratic valuation and cost functions, which appear widely in related domains and may be considered realistic enough). It would be interesting to see to what extent the results develop to more general cases.

An additional user-friendly feature would be to offer the choice between a symmetric and a payment-based scheme to users. We found that if the prices, tuned by the operator, are too low (price earned by user) or extremely high (price paid by user), the users are better off sticking to the symmetric approach. Here also the operator experiences profit-loss due to “symmetric scheme following” users. This complex system is subject to future work.

Social connections could give a new track for the model. If social network externalities exist, they should be involved in the payoff function. More, deeper studies are required for this question to decide if it is a realistic issue or not.

Bibliography

[1] E. Adar and B. Huberman. Free riding on gnutella. Technical report, Xerox PARC, 2000.

[2] A. Adya, W. Bolosky, M. Castro, R. Chaiken, G. Cermak, J. Douceur, J. Howell, J. Lorch, M. Theimer, and R. Wattenhofer. Farsite: Federated, available, and reliable storage for an incompletely trusted environment. In *5th OSDI*, 2002.

[3] K. Anagnostakis and M. Greenwald. Exchange-based incentive mechanisms for peer-to-peer file sharing. In *Proc. of 24th International Conference on Distributed Computing Systems (ICDCS'04)*, Tokyo, Japan, Mar 2004.

[4] P. Antoniadis, C. Courcoubetis, and R. Mason. Comparing economic incentives in peer-to-peer networks. *Comput. Networks*, 46(1):133–146, 2004.

[5] P. Antoniadis, C. Courcoubetis, and B. Strulo. Incentives for content availability in memory-less peer-to-peer file sharing systems. *SIGecom Exch.*, 5(4):11–20, 2005.

[6] C. Batten, K. Barr, A. Saraf, and S. Treptin. pstore: A secure peer-to-peer backup system. Unpublished report, 2001.

[7] L. Buttyan and J. Hubaux. Stimulating cooperation in self-organizing mobile ad hoc networks. Tech. Rep. DSC.

[8] B. Cohen. Incentives build robustness in bittorrent, 2003.

[9] C. Courcoubetis and R. Weber. Incentives for large peer-to-peer systems. *IEEE JSAC*, 24(5):1034–1050, May 2006.

[10] L. Cox and B. Noble. Pastiche: Making backup cheap and easy. In *Proc. of Fifth USENIX Symposium on Operating Systems Design and Implementation, Boston, MA, December 2002.*, 2002.

[11] L. Cox and B. Noble. Samsara: Honor among thieves in peer-to-peer storage. In *Proc. of the ACM Symposium on Operating Systems Principles*, 2003.

[12] F. Dabek, M. Frans Kaashoek, D. Karger, R. Morris, and I. Stoica. Wide-area cooperative storage with CFS. In *Proc. of the 18th ACM Symposium on Operating Systems Principles (SOSP '01)*, Chateau Lake Louise, Banff, Canada, October 2001.

[13] P. Druschel and A. Rowstron. PAST: A large-scale, persistent peer-to-peer storage utility. In *HotOS VIII*, pages 75–80, Schloss Elmau, Germany, May 2001.

[14] J. Feigenbaum and S. Shenker. Distributed algorithmic mechanism design: Recent results and future directions. In *Proc. of the 6th International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications*, pages 1–13. ACM Press, New York, 2002.

[15] M. Feldman, K. Lai, I. Stoica, and J. Chuang. Robust incentive techniques for peer-to-peer networks. In ACM Electronic Commerce, 2004.

[16] M. Feldman, C. Papadimitriou, J. Chuang, and I. Stoica. Free-riding and whitewashing in peer-to-peer systems. In *3rd Annual Workshop on Economics and Information Security (WEIS04)*, 2004.

[17] D. Figueiredo, J. Shapiro, and D. Towsley. Incentives to promote availability in peer-to-peer anonymity systems. In *ICNP '05: Proc. of the 13TH IEEE International Conference on Network Protocols (ICNP'05)*, pages 110–121, Washington, DC, USA, 2005. IEEE Computer Society.

[18] FON. http://en.fon.com.

[19] D. Fudenberg and J. Tirole. *Game Theory.* MIT Press, Cambridge, MA, 1991.

[20] C. Gkantsidis and P. Rodriguez Rodriguez. Network coding for large scale content distribution. In *IEEE Infocom 2005*, Miami, USA, March 2005.

[21] P. Golle, K. Leyton-Brown, I. Mironov, and M. Lillibridge. Incentives for sharing in peer-to-peer networks. *Lecture Notes in Computer Science*, 2232:75+, 2001.

[22] F. Junqueira, R. Bhagwan, A. Hevia, K. Marzullo, and G. Voelker. Surviving internet catastrophe. In *Proc. of USENIX Annual Technical Conf.*, May 2005.

[23] K. Lai, M. Feldman, I. Stoica, and J. Chuang. Incentives for cooperation in peer-to-peer networks. In *Proc. of Workshop on Economics of Peer-to-Peer Systems*, 2003.

[24] M. Lillibridge, S. Elnikety, A. Birrell, M. Burrows, and M. Isard. A cooperative internet backup scheme. In *Proc. of the 2003.*, 2003.

[25] T. Ngan, A. Nandi, A. Singh, D. Wallach, and P. Druschel. On designing incentives-compatible peer-to-peer systems. In *Proc. FuDiCo'04*, Bertinoro, Italy, 2004.

[26] V. Vishnumurthy, S. Chandrakumar, and E. Sirer. Karma: A secure economic framework for peer-to-peer resource sharing. Papers published on Web site: http://www.sims.berkeley.edu/research/ conferences/p2pecon/index.html., 2003.

[27] B. Yu and M. P. Singh. Incentive mechanisms for peer-to-peer systems. In *AP2PC*, pages 77–88, 2003.

[28] S. Zhong, Y. Yang, and J. Chen. Sprite: A simple, cheat-proof, credit-based system for mobile ad hoc networks. Technical Report Yale/DCS/TR1235, Department of Computer Science, Yale University, 2002.

Appendix

A.1 Simulation code

Contents


```
clear all
close all

global n n_round Cs Co mu ab minmax ps po alpha delta
```

generate global variables, user preferences

```
n           = 1000;           % max possible number of players
Cs          = zeros(n);     % capacity to backup for each peer
Co          = zeros(n);     % capacity to share at each peer
mu          = zeros(n,6);   % existing?, \mu_{on}, \mu_{off}, online or
                            % offline, spent time online, spent time in system
ab          = rand(n,2);    % price sensitivity
minmax      = rand(n,2);    %[zeros(n,1) ones(n,1)] % price thresholds (linked to mu, but not much
                            % correlation because of random \delta, \gamma)
```

```
n_round     = 100000;
delta       = 1;            % time unit in sec
lambda      = 0.5;          % new users' arrival rate
%disp(['Number of players (n) =' int2str(n)]);
```

generate a decent number of arrivals for a start

```
for i = 1 : n/10

    arr()

end
```

generate an initial price

```
po          = 0.5;         % fee given for offered capacity
alpha       = 1;           % coeff for ps/po
ps          = alpha * po;  % price of backup capacity
```

start game

```
h=waitbar(0,'Rounds...');

average_profit = 0;
old_average_profit = 0;
interval = 100;
change = 0;

for round = 1:n_round %number of rounds

    waitbar(round/n_round,h);
    sum_u_offer(round)=0;
    sum_u_service(round)=0;
```

manage payoff calculation at peers (strategies)

```
    for i = 1:n
```

```
        if (mu(i,1) == 1)
            [u_offering,u_service] = payoff(i);
            sum_u_offer(round)=sum_u_offer(round) + u_offering;
            sum_u_service(round)=sum_u_service(round) + u_service;
        end
    end
```

optimal social welfare calculation

```
    p(round) = 0; % optimal price: p*
    A = 0; % sum of a_i*p_min
    B = 0; % sum of b_i*p_max

    for i = 1:n

        if (mu(i,1) == 1) % if user exists

            p(round) = p(round) + (ab(i,2)*minmax(i,2)); % service part of p*
            B = B + ab(i,2);
            if (mu(i,4) == 1) % if user is online
                p(round) = p(round) + (ab(i,1)*minmax(i,1)); % offer part of p*
                A = A + ab(i,1);
            end
        end
    end

    p(round) = p(round) / (A + B);
    sw(round) = 0;
    for i = 1:n

        if (minmax(i,2) > p(round))
            sw(round) = sw(round) + ab(i,2)*( (minmax(i,2))^2 - (p(round))^2 );
        end
        if (minmax(i,1) < p(round))
```

```
        sw(round) = sw(round) - ab(i,1)*( (p(round))^2 - (minmax(i,1))^2 );
    end
end
sw(round) = sw(round)/2;
```

symmetric social welfare calculation

```
sw_sym(round) = 0;

for i = 1:n
    sw_sym(round) = sw_sym(round) + ab(i,1)*ab(i,2)/(ab(i,1)+ab(i,2))*(minmax(i,2)-minmax(i,1))^2;
end

sw_sym(round) = 1/2 * sw_sym(round);
```

manage rounds (count time, calculate prob)

```
for i = 1:n

    if ( mu(i,1) == 1 )

        if ( mu(i,4) == 1 )
            % the peer has been in the system, and online now
            mu(i,5) = mu(i,5) + 1; % an other round the peer participated
        end

        mu(i,6) = mu(i,6) + 1; % even it was offline, it still member of the system

        proba(i,round) = mu(i,5)/mu(i,6);

    end

end
```

determine new prices

```
totalCs(round) = 0;
totalCo(round) = 0;

for i = 1:n

    totalCs(round) = totalCs(round) + Cs(i); % total required backup capacity

    if (mu(i,4) == 1)
        totalCo(round) = totalCo(round) + Co(i); % total offered capacity
    end

end

profit(round) = totalCs(round)*ps - totalCo(round)*po;
```

satisfying constraints

```
price_changed = 0;
if ((alpha * totalCs(round)) < totalCo(round))
    % feasibility constraint (not profitable)
    % too much offer, too little demand

    % decrease po or ps by 0.1%
    if (rand < 0.5)
        po = po * 0.999;
    else
        ps = ps * 0.999;
    end

elseif ( totalCs(round) > totalCo(round) )
```

```
        % availability constraint
        % not enough offer

        % increase po or ps by 0.1%
        if (rand < 0.5)
            ps = ps * 1.001;
        else
            po = po * 1.001;
        end

    end

    alpha = ps / po;
```

improving profit

```
    if ( mod(round,interval) == 1 )

        if ((round > n_round/100) & ((change==0) |
         (average_profit > old_average_profit)))

            r = rand;
            if (r < 0.25)
                po = 0.98*po;
                change = 1; % shows if last change was in po (1)-(3) or in ps (2)-(4)
            elseif ((0.25 < r) & (r < 0.5))
                ps = 1.02*ps;
                change = 2;
            elseif ((0.5 < r) & (r < 0.75))
                po = 1.02*po;
                change = 3;
            else
                ps = 0.98*ps;
                change = 4;
```

```
        end


    elseif ((round > n_round/100) & (average_profit < old_average_profit))

        if (change == 1)
            po = 1.02*po;
            change = 0;
        elseif (change == 2)
            ps = 0.98*ps;
            change = 0;
        elseif (change == 3)
            po = 0.98*po;
            change = 0;
        elseif (change == 4)
            ps = 1.02*ps;
            change = 0;
        end


    end

    old_average_profit = average_profit;
    average_profit = 0;

end

average_profit = average_profit + profit(round);

alpha = ps/po;

alphaO_vec(round) = po;
alphaP_vec(round) = ps;
```

manage peer departures and new/re arrivals

```
dep();

re_arr(); % online state lasts at least for one round

% new arrivals by lambda poisson, short descreet time intervals
% lambda is small --> new user comes relatively rarely, ONE at a time

if (rand < (lambda * delta))
    n_new = 1;
else
    n_new = 0;
end

no_current_users = 0;
no_current_online_users = 0;

for i = 1:n

    if (mu(i,1) == 1)
        no_current_users = no_current_users + 1;
        if (mu(i,4) == 1)
            no_current_online_users = no_current_online_users + 1;
        end
    end

end


if ((no_current_users < n) & (n_new == 1))
    arr()
end
```

```
end % end rounds

close(h);
```

figures about the system

```
figure;
plot(1:round, alphaO_vec);
hold on;
plot(1:round, alphaP_vec, 'r');
hold on;
plot(1:round, p, 'g');
hold on;
plot(1:round, p/2, 'g');
hold on;
plot(1:round, (1+p)/2, 'y');
legend('po','ps', 'p^*', 'max p^o', 'max p^s');
title('Price p^O, p^s, p^* and the profit maximizing p^o, p^s');
xlabel('round');
ylabel('price');


figure;
plot(1:round, totalCo);
hold on;
plot(1:round, totalCs, 'r');
legend('C^o','C^s');
title('Capacity C^o and C^s');
xlabel('round');
ylabel('capacity');

figure;
plot(1:round, sum_u_offer);
hold on;
plot(1:round, sum_u_service,'r');
hold on;
```

```
plot(1:round, sum_u_offer + sum_u_service, 'm');
hold on;
plot(1:round, sum_u_offer + sum_u_service + profit, 'y');
hold on;
plot(1:round, sw, 'g');
hold on;
plot(1:round, sw_sym, 'k')
legend('utilities for offering','utilities for backup', 'user welfare', 'social welfare', 'optimal social welfare', 'symmetric social welfare');
title('Welfares');
xlabel('round');
ylabel('monetary value');


figure;
plot(1:round,profit);
title('Profit');
xlabel('round');
ylabel('monetary value');

% figure;
% hold on;
% leg=[];
% for i=1:5
%     plot(1:round,proba(i,:));
%     leg=[leg ; ['player ' int2str(i)]];
% end
% legend(leg);
% title('User availability');
```

peer arrival function

```
function y = arr()

global n mu minmax
```

```
for i = 1:n         % search for the first empty place in vector for new peer

    if (mu(i,1) == 0)
        break
    end

end

mu(i,1) = 1;              % user i exists
mu(i,2) = 0.1 * rand;     % rate of user i to go offline from online (should be small)
mu(i,3) = 10 * rand;      % rate of user i to go back online (should be large number)
mu(i,4) = 1;              % user i is online
```

peer departure function

```
function y = dep()

global mu n delta

for i = 1:n % independent user behaviour

    if (mu(i,4) == 1) % which is online at the moment

        if (rand < (mu(i,2) * delta))

            mu(i,4) = 0; % peer is departed
            %disp('Peer is gone.');

        end

    end

end
```

peer re-arrival function

```
function y = re_arr()

global mu n delta

for i = 1:n % search for existing but offline peers

    if ((mu(i,1) == 1) & (mu(i,4) == 0))

        if (rand  < (mu(i,3) * delta))

            mu(i,4) = 1; % i comes back online

        end

    end

end
```

payoff function, strategy

```
function [u_offering,u_service] = payoff(i)

global n Cs Co mu ab minmax ps po

% try to increase user payoff
% search for max value


% payoff from service
```

```
if (ps <= minmax(i,2))

    Cs(i) = ab(i,2) * (minmax(i,2) - ps);

    u_service = 1/ab(i,2) * (ab(i,2) * minmax(i,2) * Cs(i) - ((Cs(i))^2)/2) - Cs(i) * ps;

else

    Cs(i) = 0;
    u_service = 0;

end

% payoff from offering
if ((mu(i,4) == 1) & (po >= minmax(i,1)))

    Co(i) = ab(i,1) * (po - minmax(i,1));

    u_offering = Co(i) * po -  1/ab(i,1) * ((Co(i))^2)/2 - minmax(i,1) * Co(i);

else

    Co(i) = 0;
    u_offering = 0;

end
```

A.2 Simulation figures

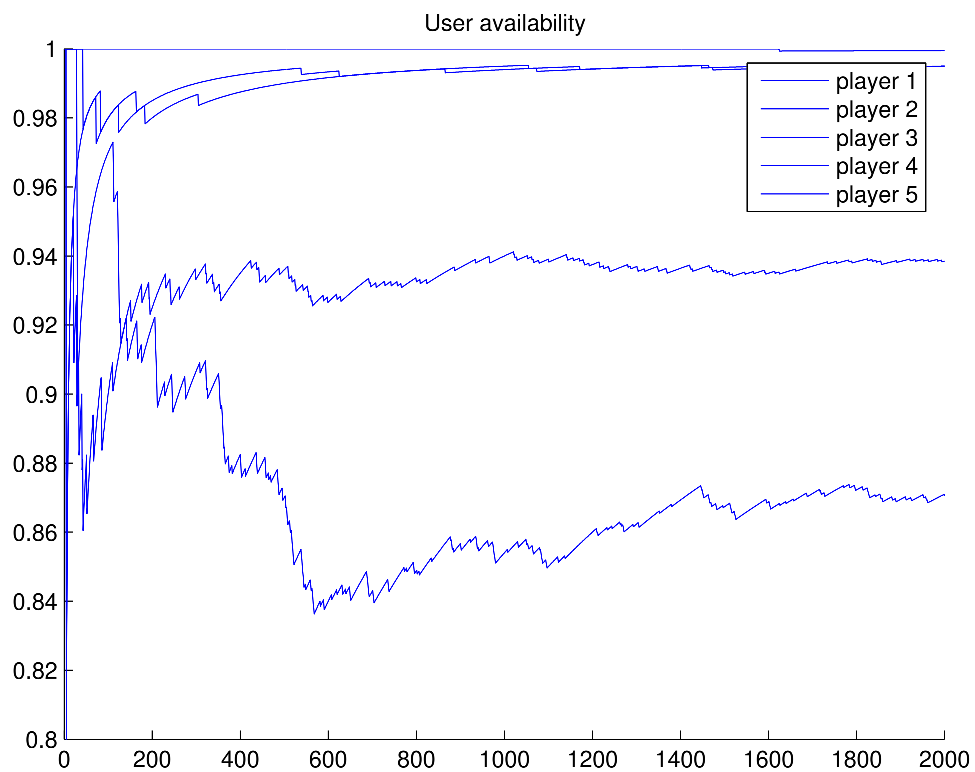

Figure A.1: Evolution of a sample user set availability over the rounds

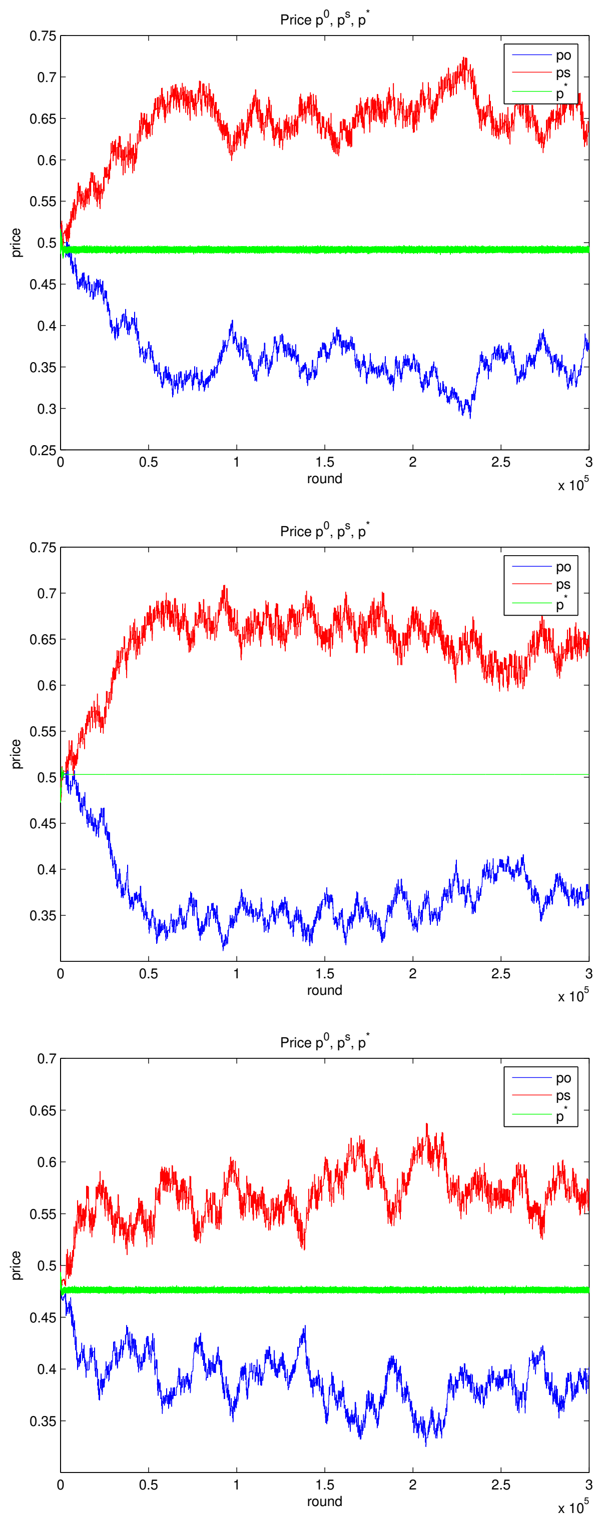

Figure A.2: Evolution of prices p^o and p^s over the rounds (with the optimal prices

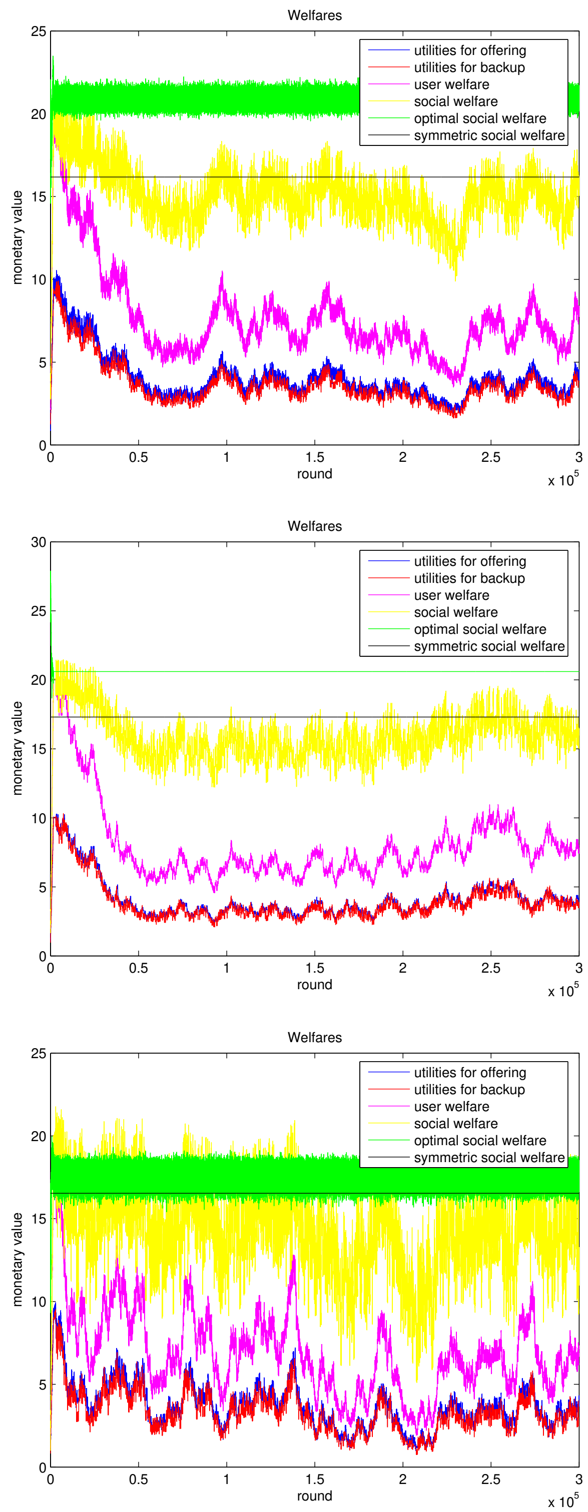

Figure A.3: Evolution of social welfare over the rounds

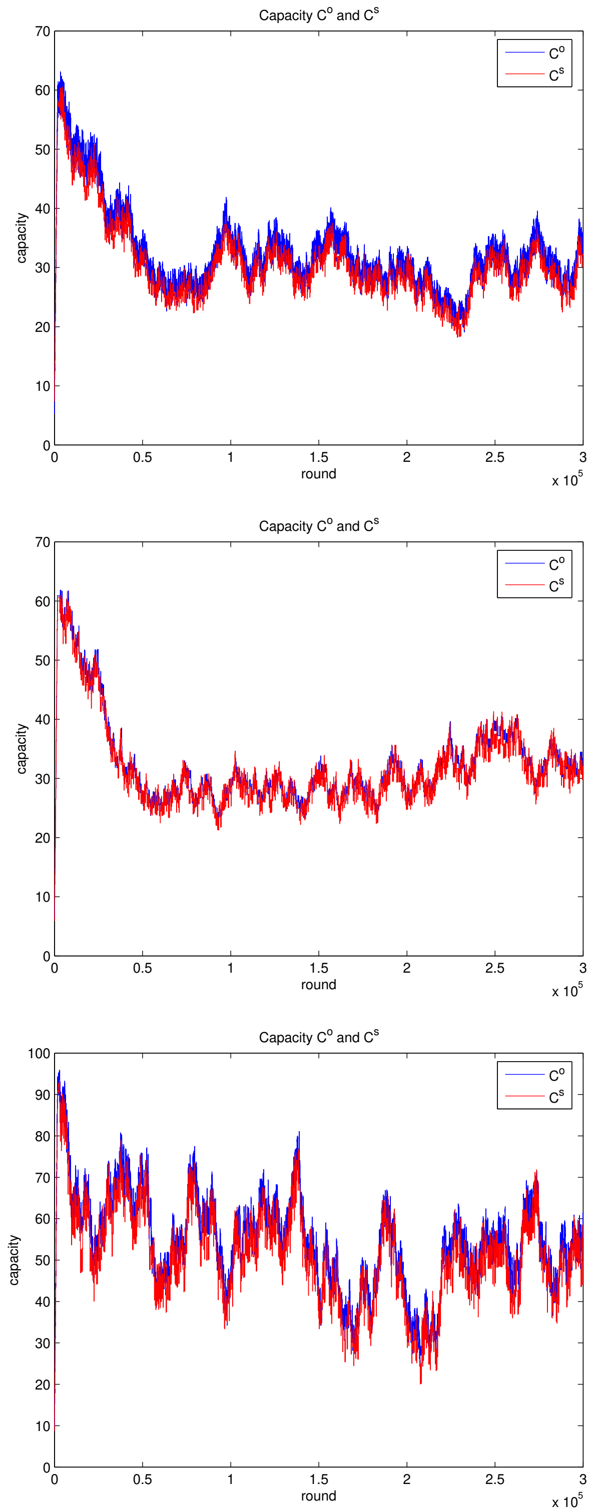

Figure A.4: Evolution of capacities over the rounds

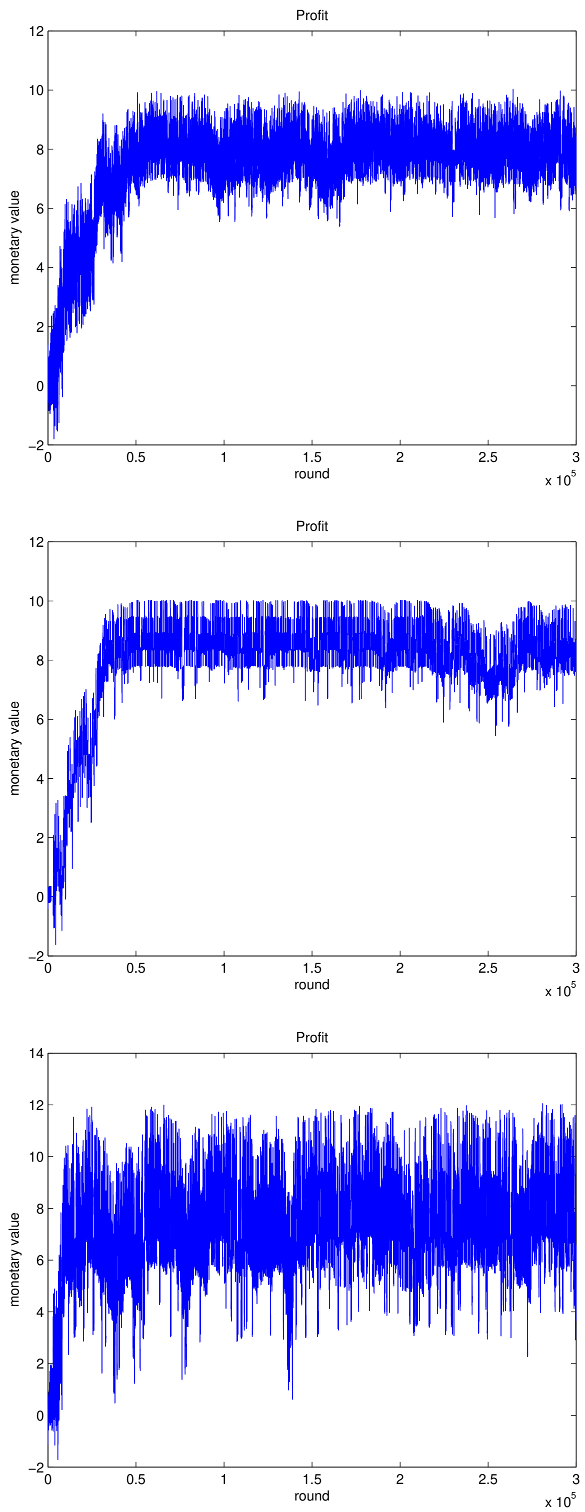

Figure A.5: Evolution of operator's profit over the rounds

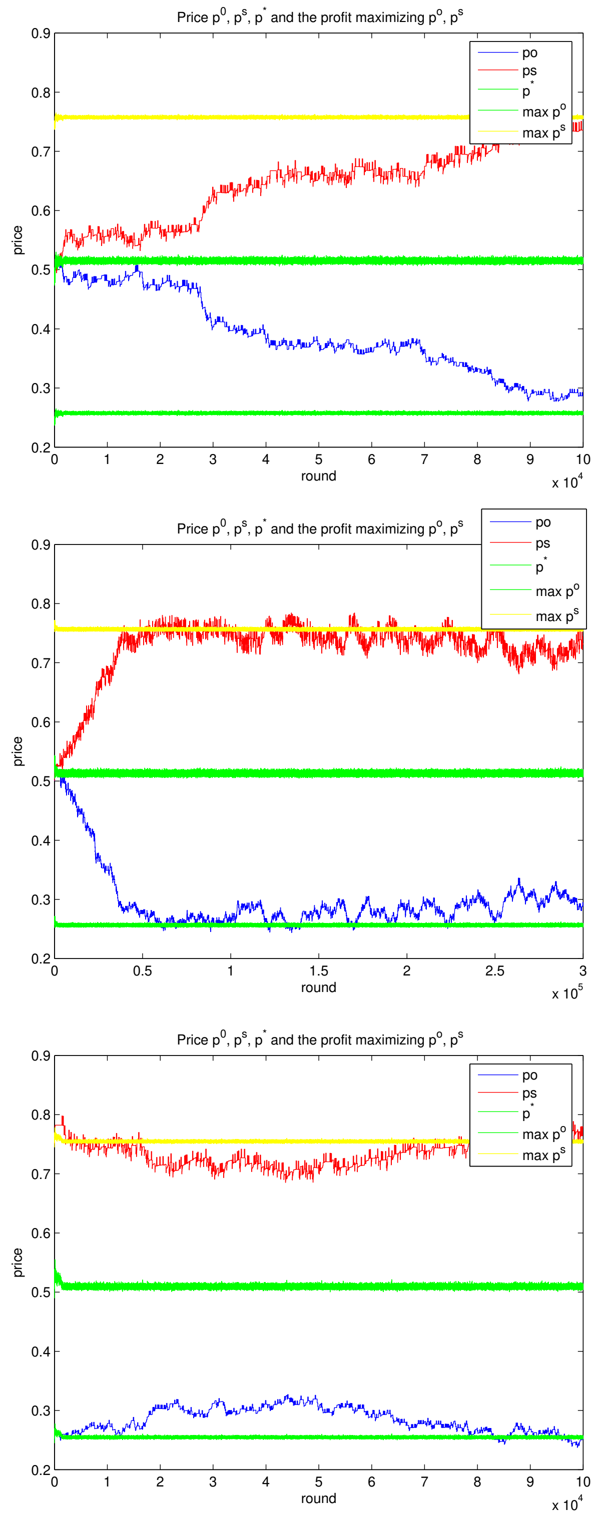

Figure A.6: Evolution of prices p^o and p^s over the rounds (with the optimal prices

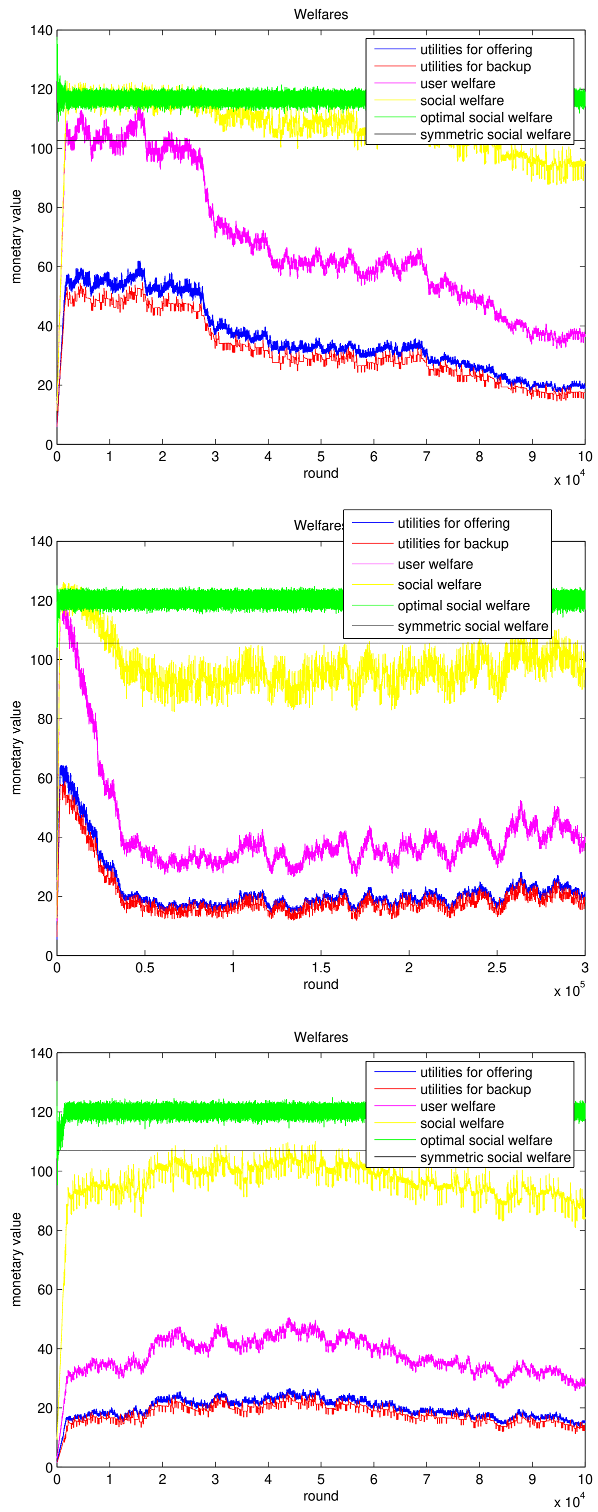

Figure A.7: Evolution of social welfare over the rounds

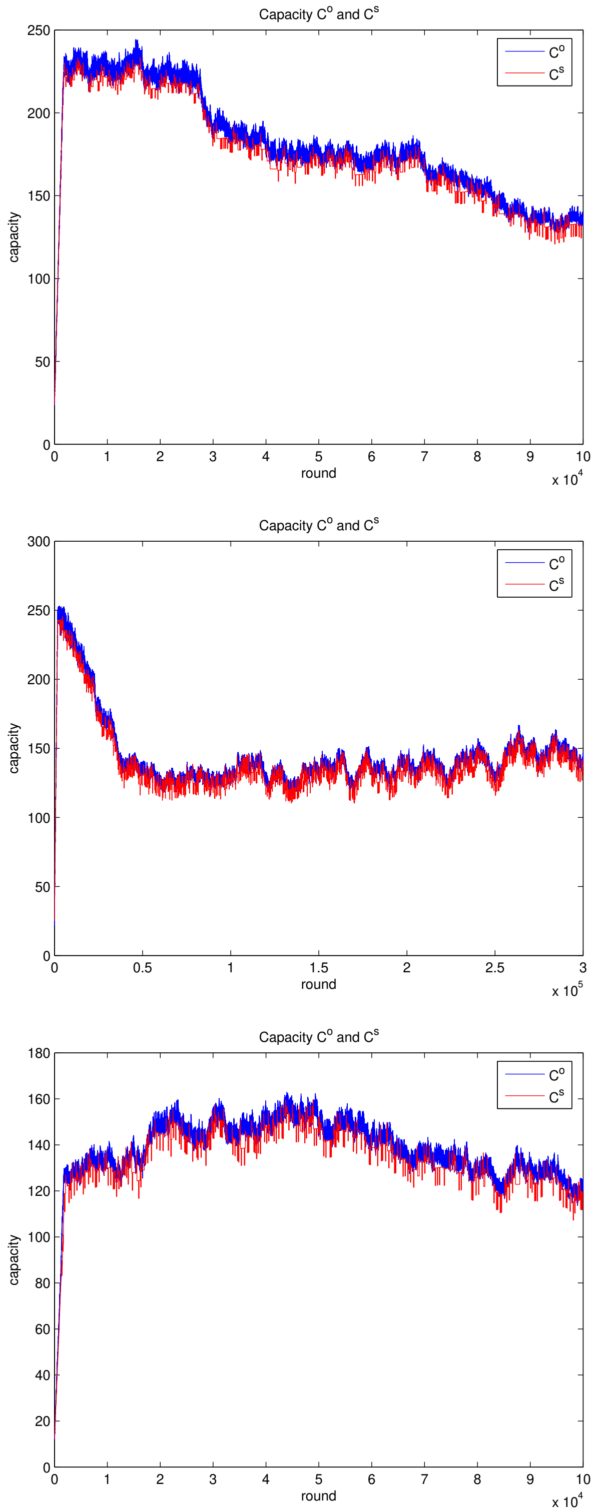

Figure A.8: Evolution of capacities over the rounds

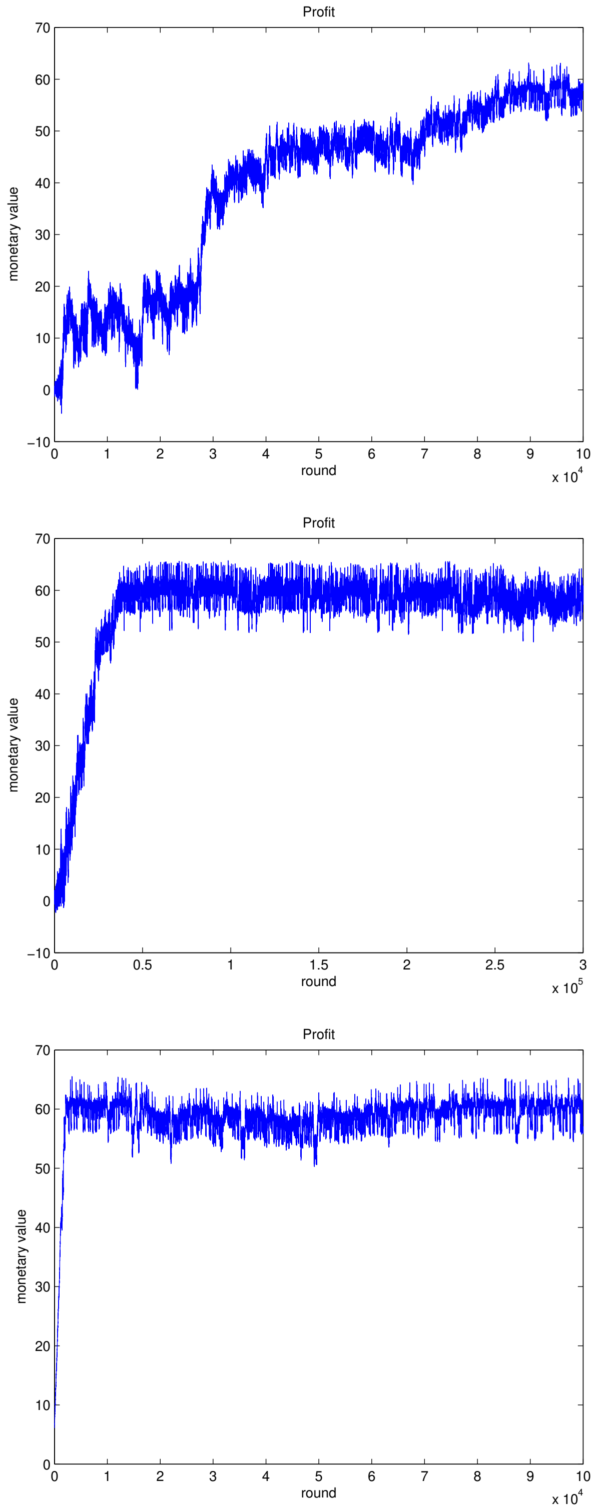

Figure A.9: Evolution of operator's profit over the rounds