HARVARD UNIVERSITY
Graduate School of Arts and Sciences

DISSERTATION ACCEPTANCE CERTIFICATE

The undersigned, appointed by the

School of Engineering and Applied Sciences

have examined a dissertation entitled

"Hidden Markets: Designing Efficient but Usable Market-based Systems"

presented by Sven Seuken

candidate for the degree of Doctor of Philosophy and hereby certify that it is worthy of acceptance

Signature ____________________

Typed name Prof D Parkes

Signature ____________________

Typed name Prof Y Chen

Signature ____________________

Typed name Prof A Roth

Signature ____________________

Typed name Dr E Horvitz

Date April 29, 2011

Hidden Markets: Designing Efficient but Usable Market-based Systems

A dissertation presented
by

Sven Seuken

to
The School of Engineering and Applied Sciences
in partial fulfillment of the requirements
for the degree of
Doctor of Philosophy
in the subject of

Computer Science

Harvard University
Cambridge, Massachusetts
May 2011

ProQuest LLC
789 East Eisenhower Parkway
P O Box 1346
Ann Arbor, MI 48106-1346

Advisor

David C Parkes

Author

Sven Seuken

Hidden Markets: Designing Efficient but Usable Market-based Systems

Abstract

The next decade will see an abundance of new intelligent systems, many of which will be market-based Soon, users will interact with a lot of new markets, sometimes without even knowing it when driving their car, when backing up their files, or even when surfing the web I argue that for the *design* of these new markets, we need to depart from traditional market designs and relax the assumptions of the selfish rational actor model I study four market domains where 1) participants may be non-experts, 2) they may have high cognitive costs, 3) they may exhibit social preferences, or 4) where typical market institutions are not available I make four market design contributions for such non-traditional domains

First, I introduce the "hidden market design" paradigm for domains where users might be non-experts and where money might be unnatural I present a case study of a hidden peer-to-peer (P2P) backup market and show how the market and its user interface (UI) can be designed to hide many of the market's complexities

Second, I provide a principled study of "market user interface design " Via lab experiments, I analyze the effect of changing a market's UI on users' decision making performance and the market's efficiency I find behavioral factors and individual user differences are essential for successful market UI optimization

Third, I present a field experiment, studying the degree of selfishness and altruism among P2P file sharing users Our data suggests that users consider the trade-off between the personal and societal effects of their actions when making a decision Furthermore, we find that an understanding of the public goods game underlying P2P file sharing leads to significantly higher rates of cooperation

Fourth, I present a study of *work accounting mechanisms* for distributed work systems where monitoring is not possible We design a mechanism that is misreport-proof and enables agents to distinguish between free-riders and cooperative peers, thereby increasing efficiency On the other hand, we also establish an impossibility result, proving that no useful and sybil-proof accounting mechanism exists

Contents

Citations to Previously Published Work

Some parts of the introductions to Chapters 1 and 2 have previously appeared in

> "Hidden Market Design", Sven Seuken, Kamal Jain, and David C Parkes, *Proceedings of the Conference on Artificial Intelligence (AAAI)*, Atlanta, GA, July 2010

Large parts of the material presented in Chapter 2 have appeared in the following two papers

> "Market Design and Analysis for a P2P Backup System", Sven Seuken, Denis Charles, Max Chickering, and Sidd Puri, *Proceedings of the ACM Conference on Electronic Commerce (EC)*, Cambridge, MA, June 2010
>
> "Hidden Markets UI Design for a P2P Backup Application", Sven Seuken, Kamal Jain, Desney Tan, and Mary Czerwinski, *Proceedings of the Conference on Human Factors in Computing Systems (CHI)* , Atlanta, GA, April 2010

A portion of the material presented in Chapter 5 has appeared in the following two papers

> "Accounting Mechanisms for Distributed Work Systems", Sven Seuken, Jie Tang, and David C Parkes, *Proceedings of the Conference on Artificial Intelligence (AAAI)*, Atlanta, GA, July 2010
>
> "On the Sybil-Proofness of Accounting Mechanisms", Sven Seuken and David C Parkes, *Proceedings of the Workshop on the Economics of Networks, Systems and Computation (NetEcon)*, San Jose, CA, June 2011

Acknowledgments

First and foremost, I am deeply thankful to David Parkes as I could not have wished for a better advisor Over the last five years, David has been an outstanding mentor, collaborator, a constant inspiration, and a friend, supporting me on each of my endeavors He will always be my academic role model, and I hope I can some day be an equally good advisor to my own students as David was to me

I thank my other committee members Al Roth, Eric Horvitz, and Yiling Chen Al's approach to real-world market design has inspired much of my own research, Eric is my ideal example for how to incorporate AI into real systems, and Yiling has always asked the right questions helping me improve my research

Other mentors I want to thank include Gunter Muller, Torsten Eymann, and Shlomo Zilberstein Gunter supported me during my college years and introduced me to interdisciplinary research Torsten taught me about market negotiations and sparked my interest in multi-agent systems Shlomo was a great mentor during my time at UMass Amherst, giving me the freedom to explore my own research ideas

I am thankful to Microsoft, for three exciting summers as an intern and for financial support via the Microsoft Research PhD Fellowship I thank Sidd Puri, Denis Charles, and Max Chickering, for letting me design a new marketplace from scratch, which became my most interesting research project I thank Mary Czerwinski and Desney Tan for introducing me to computer human interaction research I thank Kamal Jain for helping me explore my many research ideas, no matter how crazy, and for allowing an intern of the theory group to run a usability study I also thank the other interns and friends I met at MSR, in particular Flavio Ribeiro and Julia Ruscher, who made my summers in Seattle much more fun, and helped me get through my experiments and proofs.

I thank my oldest friends from Germany, especially Christine Beese, Frédéric Wils, Maike Rocker, and Timo Vollmer, who have helped me many times along the way. I also thank my friends, colleagues, and collaborators, who made graduate school such an enjoyable and stimulating experience, in particular Andrew Mao, Alice Gao, Ariel Procaccia, Chris Amato, Dan Bernstein, Emilio Calvano, Felix Fischer, Florin Constantin, George Konidaris, Giro Cavallo, Haoqi Zhang, Ian Kash, Jacomo Corbo, John Lai, Kobi Gal, Malvika Rao, Marek Petrik, Mike Ruberry, Philipp Weis, Scott Kominers, and Shaili Jain. I am particularly thankful to Jens Witkowski, for endless discussions about research, for constantly challenging me and my ideas, and for being a great friend.

I thank my parents Martha and Werner Seuken, who have supported me throughout this whole journey. They laid the early groundwork, encouraged critical thinking and a curiosity for how the world works.

Finally, I thank my wife Sonja who left Germany in 2006 to join me on this academic journey in the US. I feel incredibly lucky that we got to spend the last five years together and I am thankful for her constant support and love. She makes me happy!

Dedicated to my wife Sonja,
and to my parents Martha and Werner

Chapter 1

Introduction

1.1 New Technologies Enable New Markets

The Internet has allowed market-based systems to become increasingly pervasive While many people primarily think of Amazon or eBay when they hear about *electronic markets*, the development of new technologies is continuously enabling new kinds of markets in non-traditional domains For example, users can now pay or earn money for asking or answering questions on the web [63] Some toll roads adjust their prices dynamically as traffic changes [106] Soon, the introduction of the *smart grid* will allow end-users to actively participate in the electricity market [102] In theory, introducing market mechanisms into these previously market-free domains can increase social welfare In practice, however, these markets may be unnatural or complex, such that market participants (or agents) may find it difficult to interact with them Thus, careful choices must be made when *designing* these markets In this thesis, I study the design of electronic markets for non-traditional domains In particular, I consider situations, where participants may be non-experts, may have high cognitive costs, may have other-regarding preferences, or where typical market institutions are not available My objective is to design *efficient* but *usable* electronic markets

1.1.1 Adaptive Toll Road Prices

To illustrate the challenges involved in the design of markets, consider first the toll road domain Many states in the US are introducing high occupancy toll (HOT) lanes on their highways In general, one of the lanes can be used for free, while drivers have to pay a toll for using the other lane Oftentimes, these tolls are adaptive, i e , the price changes depending on current traffic (e g , between $0 50 and $9 00 on SR167 in Washington) such as to guarantee a minimum speed level on the HOT lane (e g , 45mph) Compare this design to a system without adaptive tolls, where either every driver pays the same toll to use the highway or the highway is simply financed via taxes during times of high traffic, both lanes would be congested such that all drivers would experience long delays The adaptive toll road design allows drivers with a high value for time to use the toll lane and drivers with a low value for time to choose the free lane Thus, if drivers have heterogenous values for time, the introduction of this market can increase social welfare Observe that the system operator has significant freedom in designing this market what should be the minimum speed level on the HOT lane? Should there be a maximum price? Should the price that drivers pay be based on current traffic when entering the lane, or should it be based on the actual traffic they experienced? Should prices be displayed in absolute values or relative to the driving time saved by using the HOT lane? All of these design choices affect how drivers behave in this market, and thereby affect the efficiency of the overall system

1.1.2 A Market for Questions and Answers on the Internet

Next, consider the Question & Answer (Q&A) website JustAnswer[1], which allows users to ask questions online Users commit to pay real money for a satisfying answer, where the price they pay depends on the level of urgency and the level of detail they specify upfront On the other side of the market are domain experts who get paid for their advice Thus, this market matches people who are willing to pay for advice with people who are willing to offer their advice for money Now compare that to Yahoo! Answers[2], a Q&A website where users cannot pay for their answers, and where users who provide answers can only expect a few "points" that may increase their ranking on a leader board Yet, Yahoo! Answers has more than 10 Million US visitors per month and is the second largest Q&A site on the Internet In contrast, Google Answers[3], which had a business model similar to JustAnswer, shut down their service in late 2006, apparently because not enough people used it Furthermore, Chen et al [14] have shown that paying higher prices for questions on Google Answers led to longer but not better answers Hsieh and Counts [44] have shown that market-based Q&A services can reduce wasted resources by eliminating less important questions and low quality answers, but that the use of a market may also reduce users' enjoyment for using the service, reducing the sense of community The missing "social" aspect of

[1] `http //www justanswer com`

[2] `http //answers yahoo com`

[3] `http //answers google com`

Google Answers might also have been one reason why the service was not frequented very much Thus, the design of knowledge markets involves many trade-offs and is far from straightforward Without careful consideration of all relevant factors, the unintended consequences of introducing a market into the domain of Q&A services may lead to a decrease rather than an increase in social welfare

1.2 Classic Market Design

As the examples in the last section have illustrated, the design of new markets can be a complicated, and challenging task It is often far from obvious which design will lead to the best outcome In economics, thc field of *market design* has recently emerged as the principled study of the design of markets Generally speaking, a *market designer* specifies the rules of the market, i e how market participants can express their preferences, how resources are allocated, and, when payments are allowed, how much each participant has to pay However, to this day, it remains unspecified which aspects of a system's design are part of market design One goal of this thesis is to extend the scope of market design to include more than just the economic aspects of a market

1.2.1 A Brief History of Market Design

Market design as an academic discipline is relatively young as it only emerged in the 1990s (see Roth [82] for a survey and Milgrom [69] for important open research questions), but many related disciplines have laid the groundwork At its core, it is based on game theory [35], which provides a mathematical study of strategic situations where multiple decision makers have conflicting interests Since the seminal work by Vickrey [103], the field of *auction theory* has developed, studying the many different ways in which goods can be sold via an auction [56] While in an auction, the price of the good to be sold is determined via the competition of potential buyers, the field of *mechanism design* abstracts away from this particular sales form It considers all possible market mechanisms (not just auctions) and their ability to efficiently allocate an object among a group of self-interested agents, given that the interested parties may have private information about the object [48] It constitutes the most general, mathematical study of incentive design for economic systems involving money The field of *matching mechanisms*, in contrast, studies domains where monetary transfers are limited or not available, due to, for example, moral objections (e g, kidney exchange markets, matching organ donors with sick patients [87]), or fairness considerations (e g , school choice mechanisms, matching students with high schools [1])

While market design builds on all of the aforementioned disciplines, it is different in that it takes a less theoretical and a more applied approach This is a practical necessity, because the assumptions of theoretical models are often violated in real-world market As Roth [82] puts it, when designing markets, the economist becomes an engineer, with a "responsibility for detail, and a need to deal with all of a market's complications, not just its principle features " There can be multiple causes for a market's complication, including a dynamic and uncertain environment, as well as participants with complex preferences exhibiting complex behaviors To deal with this complexity, both lab experimentation and computer simulations can aid to better understand those aspects of a market's desıgn that we cannot study analytıcally Even though not all market desıgn findıngs can be captured ın formal theorems, the scıentıfic study of market desıgn has generated some generalızable learnıngs For example, Roth [84] ıdentıfies "market thıckness," "overcomıng congestıon," and "market safety" as three key propertıes for successful markets

1.2.2 Efficiency, Revenue, and Incentive Compatibility

When choosıng among dıfferent desıgn optıons, a market desıgner can maxımıze dıfferent objectıves For example, we can maxımıze *efficıency* a market ıs maxımally efficıent, ıf ıt allocates resources to those people who value them most, thereby maxımızıng socıal welfare Alternatıvely, we can maxımıze *revenue,* ı e , the profits for the market operator Consıder a government, who should maınly care about the total welfare of ıts cıtızen, sellıng some of the country's resources (e g , FCC spectrum auctıons, electrıcıty markets, tımber auctıons) In such a case, a natural goal for the government's market desıgner ıs efficıency, although ın practıce, governments also care about revenue The prıorıtıes are often reversed for busınesses that are buılt around a market platform (e g , eBay or Amazon) A prıvate busıness normally wants to maxımıze ıts revenue, although ıt also cares about the efficıency of ıts market platform ıf competıtıon wıth other platforms ıs a concern Note that the two objectıves are generally ın conflıct wıth each other, such that the market desıgner must decıde about how to trade off efficıency for revenue

Another ımportant crıterıon of a market's desıgn ıs *ıncentıve compatıbılıty* Loosely speakıng, a market desıgn ıs ıncentıve compatıble ıf ıts partıcıpants are always best off revealing their private information honestly In the language of Roth [84], a market is *safe* when it is incentive compatible This is an important design criterion, since it determines whether market participants may engage in costly and risky strategic behavior, or if their interactions with the market are straightforward and simple In practice, full incentive compatibility is often too strong a requirement, as it can exclude many attractive designs More recently, computer scientists have advanced various notions of *approximate incentive compatibility* [55], which widens the space of possible designs In market design, we sometimes use designs that are not 100% incentive compatible if they have other attractive properties, for example regarding efficiency or simplicity of interaction Such designs may be justified if finding a beneficial manipulation is difficult, if an individual's benefit from manipulation is small, and if the damage of manipulations for the overall market is negligible

1.2.3 Electronic Market Design

In the way that the 1990s were the formative decade for market design as we know it today, the 2000s were the formative years for *electronic market design* The Internet has enabled many new markets, among them eBay[4], Amazon[5], Google's sponsored search auctions[6], and Yahoo!'s display advertising business[7], all of which today are multi-billion dollar markets Of course, these markets were also designed, sometimes by computer programmers, sometimes by product managers, and some-

[4] `http //www ebay com`

[5] `http //www amazon com`

[6] `http //www google com/intl/en/ads/searchads`

[7] `http //advertising yahoo com`

times by economists It is informative to see how small design choices had a large impact on these markets For example, in the early 2000s, Amazon was also running an auction platform similar to eBay However, a key difference was that an auction on eBay ended at a fixed time, while an auction on Amazon was automatically extended if necessary past the scheduled end time, until ten minutes had past without a bid Roth and Ockenfels [85] have shown that indeed, significantly more bids were submitted in the closing seconds of an auction on eBay compared to Amazon Furthermore, the more experience bidders tended to bid earlier on Amazon but later on eBay Thus, a simple difference in designs made for a large difference in bidder behaviors, Ockenfels and Roth [71] also prove formally that the two markets have different equilibria

Consider now the market for sponsored search auctions All major search engines, including Google, Yahoo!, and Bing, list advertisements alongside their generic search results when a users searches for a particular keyword The market design question here is which ads to display to the user, in which order, and how much an advertiser should pay for the ad (when shown, when clicked, or when a purchase is made) For a long time, sponsored search ads were sold via first-price auctions, where advertisers had to make a bid, ads were ranked according to their bids, and an advertiser had to pay its bid in case its ad was clicked However, advertisers quickly learned that they could save money by strategically adjusting their bids in every round Eventually, most major search engines switched to the "generalized second-price" (GSP) auction which removes this strategic incentive Edelman et al [25] provide a detailed analysis of the GSP auction, the design still most-widely used for sponsored search auctions today, and compare it to the Vickrey-Clarke-Groves (VCG) mechanism (e.g., [56]), a well-known benchmark from the mechanism design literature. The authors prove that small differences in the auctions' designs lead to very different equilibria.

1.2.4 Rationality and Self-Interest

The analytical study of market design requires a particular model of agent behavior. Two assumptions are common in economics [34], and also widely used in market design: first, that agents are *rational*, and second, that agents are *self-interested*. Here, rational means that agents have a set of consistent preferences over possible outcomes of a market, and that they select an optimal action, given those preferences. Self-interest means that an agent only cares about how the market outcome affects himself. In some of the emerging market domains, however, these assumptions are violated, and are even bad approximations to user behavior. One goal of this thesis is to design markets for these domains, relaxing the assumptions when necessary or suitable.

1.3 Market Designs for Non-traditional Domains

In this thesis, I study the design of a market for a decentralized peer-to-peer (P2P) backup system, a market for the allocation of bandwidth to smartphones, the behavior of users in P2P file sharing networks, and the design of accounting mechanisms for distributed work systems. For each of these domains, I relax some of the modeling assumptions that are standard in market design. The participants of these systems may be non-experts (P2P backup), they may have high cognitive costs (smartphone domain), they may have other-regarding preferences (P2P file sharing), or typical market institutions may not be available (work accounting systems) When relaxing the *rationality* assumption, I will always assume that agents have *preferences*, but that they may sometimes make sub-optimal choices with respect to these preferences The reasons for this sub-optimality may be complex In some domains, users might not understand the market well enough to make a well-informed decision, or it is too costly for them to become an expert in the domain In other markets, users may have high cognitive costs, and consequently, carefully evaluating all possible options may be too time-consuming, considering opportunity costs Note that, from a computational perspective, the assumption that an agent chooses the optimal action from a set of available actions generally assumes that the agent can *compute* which action is the best in the first place In theory, this would require agents to have an unlimited amount of time or an unlimited amount of computation resources to their disposal

Of course, I am not the first to relax the rationality assumption in economic research Psychologists and behavioral economists have long argued for agent models that take more of the psychological factors of human behavior into account [101] However, to this day, economists are reluctant to abandon the rational-actor model in favor of more complex psychological theories Roth [81] discusses multiple good reasons in favor of keeping the rationality assumption, most importantly because it serves as a useful approximation to human behavior in many situations This points to an important aspect of market design we do not necessarily want to adapt the most *complete* user model (possibly accounting for all psychological and behavioral aspects of human being), but instead use a model that best *predicts* user behavior in those situations we are interested in studying Ultimately, our goal is to design the best possible markets for real human users In this thesis, for each particular market design task, I adopt a user model that is best suited for the particular domain at hand I will now describe four non-traditional market domains where the standard agent model is not a good approximation for user behavior These four domains correspond to Chapters 2–5, and in each chapter, I will present a market design contribution for one of those domains

1.3.1 How to Design a Market if Users Don't Expect One?

Electronic markets are used daily by tens of millions of market participants They know they can buy goods on Amazon for a fixed price, or on eBay via an auction In these systems, monetary transactions are natural and the markets are conceptually simple such that even non-expert users can effectively interact with them However, as market-based systems are becoming more and more pervasive, users start interacting with markets in domains that were previously market-free How do you design a market for domains where users do not expect a market, or where monetary transfers are unnatural?

For example, recent progress on micropayment systems might soon pave the way for many new electronic markets by significantly reducing transaction costs [8] These markets may be complex or unnatural for many of its primarily non-expert users Thus, it is sometimes a pragmatic requisite to remove or *hide* the market's complexities We study this problem in the domain of a market-based P2P backup system,

[8] http //www google com/landing/onepass/

where individual users trade backup resources with each other The topic of Chapter 2 of this thesis is the design and analysis of this market Our focus is on the intersection of the market's economic design and its user interface, in particular for the purpose of *hiding* the market's complexities

1.3.2 It's not all Economics: Market User Interfaces

Even though electronic markets are becoming more and more pervasive in our lives, only little is known about the role of user interfaces (UIs) in promoting good performance How does the way we display market information to end-users, and the set of choices we offer, influence economic efficiency? Obviously, assuming a perfectly rational agent, having more choices can only be better, and the way information is displayed does not matter However, in practice, agents have cognitive costs for evaluating different options and thinking about which decision is best They must be modeled as boundedly-rational decision makers given that they only have limited amount of time and resources available for making a decision In particular, when markets are complex or highly dynamic, and when interacting with the market involves many decisions about small values, a departure form the standard perfectly rational actor model seems appropriate

We argue that the design of a market's user interface is important, and should be considered as part of the overall market design process To better understand the connection between the design of market user interfaces and the performance of the market's participants, we conducted a lab experiment using a hypothetical market for the allocation of 3G bandwidth on smartphones In Chapter 3, we present the results of this experiment, testing which behavioral factors are most important for the optimal design of market user interfaces

1.3.3 Markets in Social Communities and Social Networks

For many market domains, the standard assumption that agents are "self-interested" seems to be true, or at least approximates user behavior very well On eBay, for example, it is reasonable to assume that a bidder only cares about whether he wins the auction or not Thus, designing the eBay auction based on a self-interested agent model seem appropriate In sponsored search auctions, it makes sense that a bidder primarily cares about the placement of his own advertisement and the price he has to pay In case the other bidders are competitors in the same market, a bidder might potentially forego some winnings of his own to hurt a competitor But a bidder certainly wouldn't forego winnings of his own to the benefit of another bidder

The situation can be drastically different, however, in markets that are situated inside a social community, or in markets that are built on top of a social network Imagine you are trying to sell some of your belongings on Facebook, and the potential buyers are your friends It is likely that many of your friends are also friends with each other, and thus care about each other Or consider designing a market for grid computing resources that are to be used by a community of researchers at a university Assuming pure self-interest on the side of those researchers seems inappropriate While each researcher might primarily care about how he himself can use the computing resources, it is likely that he also takes the well-being of his colleagues into account, exhibiting "other-regarding" preferences The study of users with such other-regarding preferences is the topic of Chapter 4 We take a close look at P2P file sharing networks and analyze which factors determine whether users behave more altruistically or more selfishly

1.3.4 Markets Without Money, Contracts, or Monitoring

In many market domains, we can use money and dynamic prices to achieve the efficient allocation of resources in a society In some domains, the transfer of money is prohibited for various reasons, sometimes because of fairness considerations (e g , school choice or centralized labor markets [86]) or because monetary transaction are considered repugnant (e g , kidney exchanges or surrogacy [83]) In such cases, matching markets that operate without money can often still achieve efficient market allocations But even when market mechanisms without money are used, we can generally still write binding contracts governing the outcome of the market transactions

The Internet, however, has enabled a new paradigm of economic production, where individual users perform work for others, often in small units, for short periods of time, and without formal contracts or monetary payments These *distributed work systems* can arise in many places, for example in peer-to-peer (P2P) file sharing networks, in ad-hoc wireless routing networks, or even in casual car-pooling communities The particular challenge is to incentivize users to perform work for others, even though all interactions are bilateral and monitoring is not possible In Chapter 5 of this thesis we introduce *work accounting mechanisms* that measure the net contributions of users, despite relying on voluntary reports We begin the chapter with a very general, formal treatment of distributed work systems, and eventually apply various accounting mechanisms in the domain of P2P file sharing to improve the efficiency of BitTorrent

1.4 Outline & Overview of Contributions

The following is a detailed chapter-by-chapter outline of the thesis, together with its main contributions Chapters 2–5 each cover a separate, self-contained research project, and thus do not necessarily need to be read in sequence Each chapter studies a different market domain and presents a different market design contribution The related work for each project is provided in the corresponding chapter The discussion of future work is generally left to the concluding chapter of the thesis

1.4.1 Chapter 2: Design & Analysis of a Hidden P2P Backup Market

The main contributions presented in Chapter 2 include the introduction of a new design paradigm which we call "hidden market design," as well as the design and analysis of a hidden P2P backup market We show, how a market and its user interface (UI) can be designed to hide the underlying complexities, while maintaining the market's functionality We enable the P2P backup market using a virtual currency only, and we develop a novel market UI that makes the interaction for the users as seamless as possible The UI hides or simplifies many aspects of the market, including complementarities between the resources, prices, account balances and payments In a real P2P backup system, we can expect users to update their settings with a delay upon price changes Therefore, the market is designed to work well even out of equilibrium, by maximizing the buffer between demand and supply The main theoretical result is an existence and uniqueness theorem, which also holds if a certain percentage of the user population is price-insensitive or even adversarial However, we also show that the more freedom we give the users, the less robust the system becomes against adversarial attacks Furthermore, the buffer size has limited controllability via price changes alone and we show how to address this We introduce a price update algorithm that uses daily aggregate supply and demand data to move prices towards the equilibrium, and we prove that the algorithm converges quickly towards the equilibrium Finally, we present results from a formative usability study of the market UI, where we found encouraging results regarding the hidden markets paradigm

1.4.2 Chapter 3: Market User Interface Design

The main contributions presented in Chapter 3 include the introduction of a new research agenda on "market user interface design", as well as an empirical study of the effect of different UI design levers on user behavior and market performance We take the domain of 3G bandwidth allocation as an illustrative example, and consider the design space of UIs in terms of varying the number of choices offered, fixed vs changing market prices, and situation-dependent choice sets The UI design induces a Markov decision process, the solution to which provides a gold standard against which user behavior is studied Our findings indicate that users are surprisingly good at coming up with decision policies for the sequential optimization problem We show that their actions exhibit a high degree of rationality. However, we also show how various behavioral factors influence the users' decision making process. We find that, in general, with a larger number of choices available, users make worse decisions. When analyzing efficiency, we find that overall efficiency increases as we increase the number of choices from 3 to 4 to 5, but then plateaus, i.e., there is no statistically significant difference regarding efficiency for games with 5 or 6 choices. One of the strongest effects we find is a position effect, i.e., users are much more likely to select the optimal choice the higher its relative rank among all choices. We also find that users exhibit significant loss aversion, foregoing large future winnings to avoid short-term losses. Finally, we fit a quantal-response model to users' actions and evaluate an optimized market user interface. Here we find that the re-optimization increased the user's probability of selecting the optimal choice. However, the data suggests that the re-optimization algorithm took away too much value, in particular for the *more rational users*, while no statistically significant effect was observed for the *less rational users*.

1.4.3 Chapter 4: Selfishness vs. Altruism in P2P File Sharing Networks

In Chapter 4, we describe an economics experiment studying the degree of selfishness and altruism of P2P file sharing users. For this experiment, we released two versions of a new P2P file sharing software - a cooperative version and a selfish version - and observed the users' download decisions. The selfish client was advertised as being able to download videos at a faster speed (we varied the advertised speed-up between

0% and 45%), while allowing the users to minimize their upload to others The main contributions in this chapter are two-fold first, I present the experiment design, where the main difficulty was to indirectly elicit whether the participants of the experiment had understood the nature of the public goods game they were playing The second contribution is a detailed statistical analysis of the data, determining which factors are most predictive for users' behavior in P2P file sharing communities We found that the most important factor was whether users understood the "tragedy-of-the-commons" aspect of the public goods game for those users who understood the problem, the likelihood of choosing the cooperative client was, on average, 16% points higher than for those who didn't The second most important factor was how much faster the selfish client was compared to the cooperative client Increasing the speed-up advertised to the users from 0% to 10% increased the likelihood of choosing the selfish client by approximately 15% points However, we observe an interesting thresholding effect as increasing the speed-up further beyond 10% had no significant effect on users' behavior Other factors we found to be highly predictive for user behavior are age, country-of-origin, and the user's operating system

1.4.4 Chapter 5: Work Accounting Mechanisms

The main contributions of Chapter 5 include the formal study of work accounting mechanisms for general distributed work systems, and extensive simulation experiments using work accounting mechanisms as an overall protocol for BitTorrent We first describe BARTERCAST, a fully decentralized information exchange system, where individual agents send and receive reports about the work they have performed/received in the distributed work system We show that a straw man solution is highly susceptible to misreport manipulations Next, we introduce the Drop-Edge mechanism which removes any incentive for a user to make misreports about its own interactions We prove that the information loss necessary to achieve this incentive compatibility is small and vanishes in the limit as the number of users grows In some domains, users may be able to cheaply create fake identities (i e, sybils) and use those to manipulate the accounting mechanism A striking negative result is that no sybil-proof accounting mechanism exists if one requires responsiveness to a single positive report To evaluate the welfare properties of our mechanisms, we first present results from a discrete, round-based simulation, showing that BarterCast-Drop-Edge achieves very high efficiency We have also implemented the mechanism in Tribler, a BitTorrent software client, that is already deployed in the real world and has thousands of users Experimental results using Tribler demonstrate that the mechanism successfully prevents free-riding in P2P-file sharing systems, and achieves better efficiency than the standard BitTorrent protocol

Chapter 2

Design and Analysis of a Hidden P2P Backup Market

2.1 Introduction[1]

Reliable, inexpensive, and easy-to-use backup solutions are becoming increasingly important. Individual users and companies regularly lose valuable data because their hard drives crash, their laptops are stolen, etc. Already in 2003, the annual costs of data loss for US businesses alone was estimated to be $18.2 Billion [98]. With broadband connections becoming faster and cheaper, *online backup systems* are becoming more and more attractive alternatives to traditional backup. There are hundreds of companies offering online backup services, e.g., SkyDrive, Idrive, Amazon S3. Most of these companies offer some storage for free and charge fees when the free quota

[1] The material presented in this chapter is based on collaborations with Denis Charles, Max Chickering, Sidd Puri, Mary Czerwinski, Kamal Jain, David C. Parkes, and Desney Tan.

is exceeded However, all of these services rely on large data centers and thus incur immense costs

Peer-to-peer (P2P) backup systems are an elegant way to avoid these data center costs by harnessing otherwise idle resources on the computers of millions of individual users all users must provide some of their resources (storage space, upload bandwidth, download bandwidth, and online time) in exchange for using the backup service While the total network traffic increases with a P2P solution, the primary cost factors that can be eliminated are 1) costs for hard drives, 2) energy costs for building, running and cooling data centers[2], 3) costs for large peak bandwidth usage, and 4) personnel costs for computer maintenance A study performed by Microsoft in 2008 showed that about 40% of Windows users have more than half of their hard disk free and thus would be suitable candidates for using a P2P backup system Our own recent user study [93] found that many users are not willing to pay the high fees for server-based backup and more than half of our participants said they would consider using P2P backup instead Thus, there is definitely a considerable demand for P2P backup applications In fact, a series of P2P backup applications have already been deployed in practice (e g , *Wuala*, *Allmydata*) A drawback of the the existing systems is, that all users are generally required to supply the resources space, upload and download bandwidth in the same ratios

Our P2P backup system is novel in that it uses a market to allocate resources more efficiently than a non-market-based system could Furthermore, we provide users

[2] In 2008, data centers in the US were responsible for about 3% of the country's energy consumption Note that a P2P backup system cannot only reduce costs but is also more environmentally friendly due to reduced carbon emissions

with incentives to contribute their resources. This is in contrast to non-price based systems like BitTorrent for example, where numerous research has shown that without proper incentives, file availability rapidly decreases over time until most content finally becomes unavailable [76]. In our system, the relative market prices for the different resources function as compact signals of which resources are currently scarce, and properly motivate those users who value a specific resource least, to provide it to the system in a large quantity. Some users might need most of their own disk space to store large amounts of data and thus prefer to sacrifice some of their bandwidth. Other users might use their Internet connection a lot for services like VOIP or file sharing and might have a high disutility if the quality of those services were affected. We allow different users with idiosyncratic preferences to provide different resource bundles, and we update prices regularly taking into account aggregate supply and demand of all resources.

The design of a P2P backup market involves a series of unusual challenges, in particular at the intersection of market design and user interface (UI) design. The first and biggest challenge is that users of a backup system do not expect to interact with a market in the first place, and might find a market a very odd concept in this domain. This raises the question of how to display prices to the users if they do not even know they are interacting with a market-based system. Furthermore, users cannot be expected to monitor account balances or to make payments to the system. This challenge arises in many domains, especially in many emerging electronic markets where thousands or millions of non-sophisticated users interact with market-based systems. While these markets often provide large benefits to the users, they can also be unnatural or complex such that individuals may not have an easy time interacting with them To address this challenge in a principled way, we introduce a new design paradigm which we call "hidden market design " When designing hidden markets, we attempt to minimize or "hide" the complexities of the market to make the interaction for the user as seamless as possible A hidden market encompasses both, the design of a UI for the market and the design of the economics of the market A P2P backup application is particularly well suited to illustrate the hidden markets paradigm because the application targets millions of technically unsophisticated users, in a domain where markets are very unexpected and where many users might find the use of real money unusual Our proposed design hides many common market aspects from the users

A second market design challenge arises from the fact that users will only infrequently interact with this market They will not continuously update their settings, and thus, price changes will only affect supply and demand after a delay As a consequence, the system will be out of equilibrium most of the time, while trades must be enabled at all times The third challenge is the combinatorial aspect of the resource supply that is needed for the production of backup services All users must provide a certain amount of all resources, even if they currently only consume a subset of them For example, a user who only contributes storage space is useless to the system because no files could ever be sent or received from that user if no bandwidth is provided We call these combinatorial market requirements the *bundle constraints* because only bundles of resources have value Displaying the bundle constraints in a simple way is a major challenge for the UI design Because many of these challenges are quite unusual, providing a simple method of interaction to the users, in a domain where they do not expect a market, requires the development of new techniques for UI and market design

2.1.1 Outline and Overview of Results

We present the market and UI design for a P2P backup system and provide a theoretical and experimental analysis of its properties In Section 2 2, we introduce the preliminaries of the P2P resource market We enable the market using a virtual currency only, which avoids the various complications a real-world currency brings along (e g , state, federal, and international banking laws) and also makes the system more natural to use In Section 2 3, we first explain the hidden market design paradigm in more detail and then describe the various elements of the specific market UI we developed for the P2P backup system In a real P2P backup system, we must expect a delay in users updating their settings upon price changes, and thus the system will be *out of equilibrium* most of the time In contrast to previous work on data economies, the market is designed to work well even when *not* in equilibrium In our system, users do not have to continuously update demand and supply and instead periodically choose bounds on their maximum supply and demand We describe a new slider control, which simplifies the display of the bundle constraints and provides the users with a linear interaction with the system These sliders guarantee that users can only choose supply ratios that satisfy certain constraints, which enables us to support the market equilibrium with linear prices The UI exposes the effect of prices to users only implicitly, so as to avoid invoking a mental model of a monetary system, and it completely hides the users' account balances and the payments made in the system

The economics of the market also involve some unusual design choices In Section 2 4, we describe the market design in detail and list a series of properties of our system design that allow us to model the market as an exchange economy, even though production is happening In Section 2 5, we begin the analysis of the market equilibrium by advancing a new equilibrium concept, the *buffer equilibrium* Because the P2P backup market will be out of equilibrium most of the time, we must always have a certain buffer between supply and demand of all resources We show that the buffer between supply and demand is maximal in the buffer equilibrium, which motivates it as a desirable target concept We prove that under very reasonable assumptions, the equilibrium is guaranteed to exist, and is unique This result also holds if a certain percentage of the user population is price-insensitive or even adversarial However, we show that the more freedom we give users in choosing their supply settings, the less robust the system becomes against adversarial attacks Furthermore, we show that the size of the buffer in equilibrium cannot be controlled via price updates alone We describe which changes in the UI would be necessary to give the market operator control over the buffer size In Section 2 6, we introduce a price update algorithm that only requires system-wide supply and demand information to update prices We prove that the algorithm converges linearly towards the buffer equilibrium when initial prices are chosen close enough to equilibrium prices Finally, in Section 2 7, we present results from a formative usability study of our system, evaluating how well users can interact with the new hidden market UI The results are encouraging and show promise for the hidden market paradigm

2.1.2 Related Work

Ten years ago, the research projects OceanStore [58] and FarSite [9] already investigated the potential of distributed file systems using P2P Both projects, however, did not take the self-interest of individual users into account and did not perform any kind of market design More recently, researchers have looked at the incentive problem, often with the primary goal to enforce fairness (you get as much as you give) Samsara [20] is a distributed accounting scheme that allows for fairness enforcement However, it does not enable a system where different users can supply resources in different ratios while maintaining fairness, which is the primary advantage of our market-based system

The idea to use electronic markets for the efficient allocation of resources is even older than ideas regarding P2P storage systems Already in 1996, Ygge et al [109] proposed the use of computational markets for efficient power load management In the last five years, grid networks and their efficient utilization have gotten particular attention [59] Fundamental to these designs is that participants are sophisticated users able to specify bids in an auction-like framework While this assumption seems reasonable in energy markets or computational grid networks, we are targeting millions of users with our backup service and thus we cannot assume that users are able and willing to act as traders on a market when they want to backup their files

In the last three years, human computer interaction researchers have gotten more interested in topics at the intersection of UI design and economics Hsieh et al [45] test whether the use of markets in synchronous communication systems can improve overall welfare Hsieh et al [44] explore a similar idea in the domain of question and answer applications where users could attach payments to their questions While their use of markets is similar in vein to our approach, i e , using markets to most efficiently allocate resources as is standard in economics [42], in both papers they used a very explicit UI showing monetary prices to the users

Satu and Parikh [73] compare live outcry market interfaces in scenarios such as trading pits and electronic interfaces They draw a distinction between trying to blindly replicate the real world in the UI, and locating "defining characteristics" that must be supported In our work, we adopt this philosophy and attempt to mask the unnecessary affordances in the hopes that the relevant ones become easier to use

From the UI design point of view, the work that is closest to our approach is *Yoopick*, a combinatorial sports prediction market [38] This application provides a very intuitive UI for trading on a combinatorial prediction market The designers successfully hide the complexity of making bets on combinatorial outcomes by letting users specify point spreads via two sliders This approach is very much in line with the hidden market paradigm

On the theoretical side, the two papers most similar to our work are by Aperjis et al [6] and Freedman et al [29] They analyze the potential of exchange economies for improving the efficiency of file sharing networks While the domain is similar to ours, the particular challenges they face are quite different They use a market to balance supply and demand with respect to popular or unpopular files However, in their domain there is only one scarce resource, namely upload bandwidth, while we design an exchange market for multiple resources Furthermore, their design does not attempt to hide any of the market aspects from the users

There exist multiple P2P backup applications that are being used in practice and the application most similar to ours is *Wuala* (`www wuala com`) However, we know of no other P2P backup system that uses a market In the other backup systems, the ratios between the supplied resources space, upload and download bandwidth are fixed, and the same across all users The advantage of our market-based approach is the additional freedom we give the users Allowing them to supply different ratios of their resources increases overall economic efficiency and makes the system more attractive for every user Note that without using a market, this freedom would not be possible, because there would be no mechanism to incentivize the users to supply the scarce resources

2.2 The P2P Resource Market: Preliminaries

Our system uses a hybrid P2P architecture where all files are transferred directly between peers, but a dedicated server coordinates all operations and maintains metadata about the location and health of the files The role of the server in this system is so small that standard load-balancing techniques can be used to avoid scaling bottlenecks

Each user in the system is simultaneously a *supplier* and a *consumer* of resources A peer on the consumer side demanding a service (backup, storage, or retrieval) needs multiple peers on the supplier side offering their resources (space, upload and download bandwidth, and online time) The production process of the server (bundling multiple peers and coordinating them) is essential, turning unreliable storage from individual peers into reliable storage Each peer on the supplier side offers a different resource bundle while each peer on the consumer side gets the same product, i e , a backup service with the same, high reliability

One natural concern about P2P backup is that individual users have a much lower availability than dedicated servers Thus, a P2P system must maintain a higher file redundancy to guarantee the same file availability as server-based systems Simply storing multiple file copies would be very costly Fortunately, we can significantly reduce the replication factor by using *erasure coding* [61] The erasure code splits up a file into k fragments, and produces $n > k$ new fragments, ensuring that *any* k of the n fragments are enough to reconstruct the file Using this technique, we can achieve the same high reliability as sever-based systems while keeping replication low For example, if users are online 12h/day on average, using erasure coding we can achieve a file availability of 99 999% with a replication factor as low as 3 5, compared to simple file replication which would have a factor of 17

The process for backing up files involves four steps First, the user's files are compressed Then the compressed files are automatically encrypted with a private key/password that only the user has access to (via Microsoft LiveID) Then, the encrypted file is erasure coded, and then the individual fragments are distributed over hundreds of peers Using this process, the security of the P2P backup system can be made as high as that of any server-based system

Table 2 1 describes the five high-level operations in the P2P system Note that all of the system-level processes happen without user interaction All the user has to do is initiate a backup operation, a retrieval operation, or delete his files when he wants to

Table 2 1 Operations and their Required Resources

Operation	**Description**	**Resources Required**
Backup	When a user performs a backup, file fragments are sent from the consumer to the suppliers	Download Bandwidth
Storage	Suppliers must persistently store the fragments they receive (until they are asked to erase them	Space
Retrieval	When a user retrieves a backup, file fragments are sent from the suppliers to the consumer	Upload Bandwidth
Repair	When the server determines a backed up file to be unhealthy, the backup is repaired	Download & Upload Bandwidth
Testing	If necessary, the server initiates test operations to gather new data about a peer's availability	Download & Upload Bandwidth

Prices, Trading & Work Allocation

All trades in the market are done using a virtual currency Each resource has a price at which it can be traded and in each transaction the suppliers are paid for their resources and the consumers are charged for consuming services Prices are updated regularly according to current aggregate supply and demand, to bring the system into equilibrium over time

Trading is enabled via a centralized accounting system, where the server has the role of a bank The server maintains an account balance for each user starting with a balance of zero and allows each user to take on a certain maximal deficit The purpose of the virtual currency is to allow users to do work at different points in time while keeping all contributions and usages balanced over time Users have a steady inflow of money from supplying resources and outflow of money from consuming services, which varies over time In *steady state*, when users have been online long enough, their income must equal their expenditure Users cannot earn money when they are offline but must still pay for their backed up files Thus, their balance continuously decreases during that time When using real money, we could simply charge users' credit cards as their balance keeps decreasing However, as long as we do not use real money, the maximum deficit that users can take on must be bounded Ultimately, it is a policy decision what happens when a user hits a pre-defined deficit level Our system will first notify the users (via email and visually in the application) and present options to remedy the situation (e g , increase supply) Failing this after a reasonable timeout period (e g , 4 weeks), the users' backups will be deleted The server is involved in every operation, coordinating the work done by the suppliers and allocating work to those users with the lowest account balances to drive all accounts (back) to zero over time This is possible because users' steady-state income must equal their expenditure Thus, when users have been online for a sufficient time period, their account balance is always close to zero

2.3 The Hidden Market User Interface

The UI is an essential aspect of the market design because it defines the information flow between the user and the market The server needs to elicit a user's individual preferences, and a user needs to "experience" the current market prices

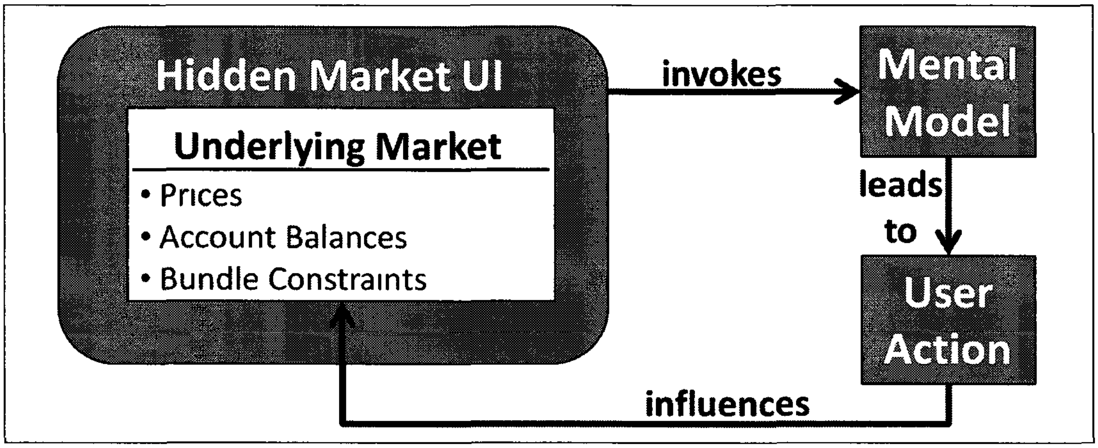


Figure 2.1 The hidden market UI wraps around the complex underlying market and exposes a simpler interface, invoking a particular mental model in the user, whose actions influence the market.

However, direct preference elicitation methods (directly asking the users for their valuations) are infeasible to implement because the amount of communication would be too high, but more importantly, because the majority of users are non-experts and would find such an interaction very complicated, unnatural, and cumbersome. To make the interaction for the user as easy as possible, we design a *hidden market UI* where we attempt to mask as much of the prices, account balances, trading constraints, etc. from the user as possible. To do this, we project a hidden market UI wrapped around the actual market to expose a simplified interface to the user (illustrated in Figure 2.1). The goal in designing this hidden market UI is to establish a feedback loop between the market and the user, without invoking a mental model of a monetary market.

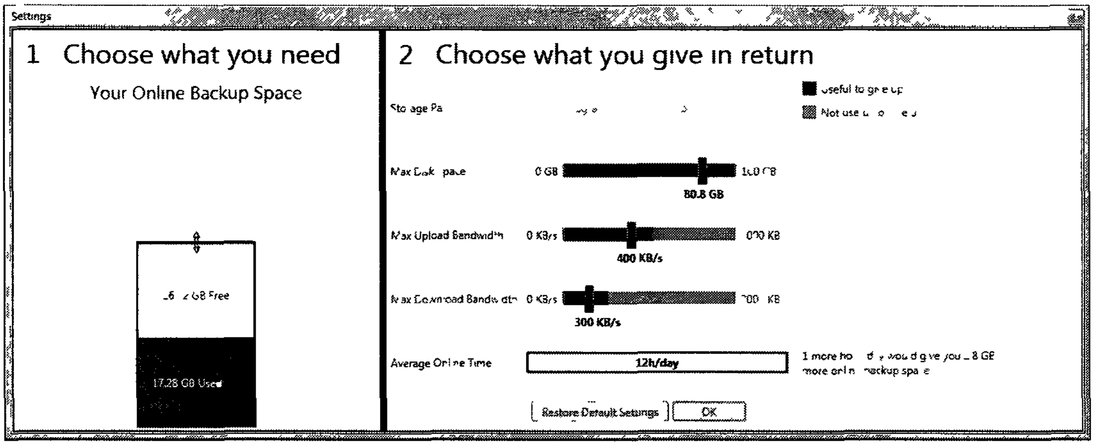
2.3.1 What You Give is What You Get

Figure 2 2 displays the market UI The user can open this “settings window” to interact with the market This window is separated into two sides on the left side, the users can choose how much online backup space they need On the bar chart the users can see how much they have already backed up and how much free online backup space they have left On the right side of the window, the users can choose how much of their own resources they want to give up in return On the top of the right side, the users see the storage path, i e , where the file pieces from other users are stored on their own computers Then, for each of the resources of space, upload and download bandwidth, there is a separate slider which the users can move to specify how much of that resource the system should maximally use [3] Below the sliders the current average online time of the users is displayed [4] Next to the online time information the system also tells the users the effect of leaving their computer online for one more hour per day (i e , how much more online backup space they would get in return) This shall make the users aware of the important role of their online time the longer the users are online, the more useful their supply of space, upload and download bandwidth becomes, and thus the higher their income

To change anything about their settings, the users can drag the bar chart on the left side up or down, move any of the sliders on the right side, or change how often

[3] The maximum value for these sliders can be determined automatically the limit for space is simply the free space on the users’ hard drives, the bandwidth limits can be determined via speed tests

[4] To change this value the users have to leave their computer online for more or fewer hours per day than they are currently doing, though we can conceive of schemes in which the application can directly control such settings as power savings and hibernate mode



Settings

1 Choose what you need

Your Online Backup Space

-6 GB Free

17.28 GB Used

2 Choose what you give in return

Storage Pa

Max Disk Space 0 GB — 80.8 GB — 100 GB

Max Upload Bandwidth 0 KB/s — 400 KB/s — 000 KB

Max Download Bandwidth 0 KB/s — 300 KB/s — 700 KB

Average Online Time 12h/day

1 more ho d you'd give you 8 GB more onl backup space

Fıgure 2 2 Screenshot of the advanced settıngs UI On the left sıde, the user can choose the desıred amount of onlıne backup space On the rıght sıde, the user can fine-tune the supply settıngs ıf desıred Account balances, prıces and payments are hıdden from the user

they are onlıne Both sıdes of the wındow are connected to each other, such that a change on eıther sıde affects and dynamıcally updates the values on the other sıde as well The semantıcs of thıs connectıon are ımportant on average, users must pay for the total consumptıon chosen on the left sıde wıth the supply chosen on the rıght sıde If a user ıncreases any of the slıders on the rıght then the bar chart on the left grows because the amount of free onlıne backup space ıncreases If a user decreases a slıder then the bar chart on the left shrınks, because the amount of free onlıne backup space decreases When a user dırectly drags the bar chart up or down to choose how much free onlıne backup space he wants, then the three slıders on the rıght sıde move left or rıght, proportıonally to theır prevıous posıtıon [5]

[5] Note that ın practıce we expect roughly two categorıes of users *basıc users* wıll only ever use the left sıde of the wındow to choose how much onlıne backup space they need They eıther do not care about whıch resources they gıve up, or they do not even understand the meanıng of upload bandwıdth, download bandwıdth, etc The second category of users are the *advanced users*, ı e , those users that understand the meanıng and relevance of gıvıng up theır own resources and want

The UI allows users to express their idiosyncratic preferences over consuming backup services and supplying their resources For example, if a user needs 20 GB of free online backup space, there are several different slider settings that allow this Some users might specify to give more space and less bandwidth, others might specify it the other way around, depending on their available resources and individual preferences Because a user's preferences can change over time this is not a task that can easily be automated Note that we do not expect users to constantly adjust their settings Rather, we expect users to choose settings that give them enough online backup space such that they do not have to worry about their settings for a while However, as they near their quotas, the system will notify them (via an email and visually in the application) At that point, we expect most users to adjust their sliders again, according to their preferences and then current market conditions

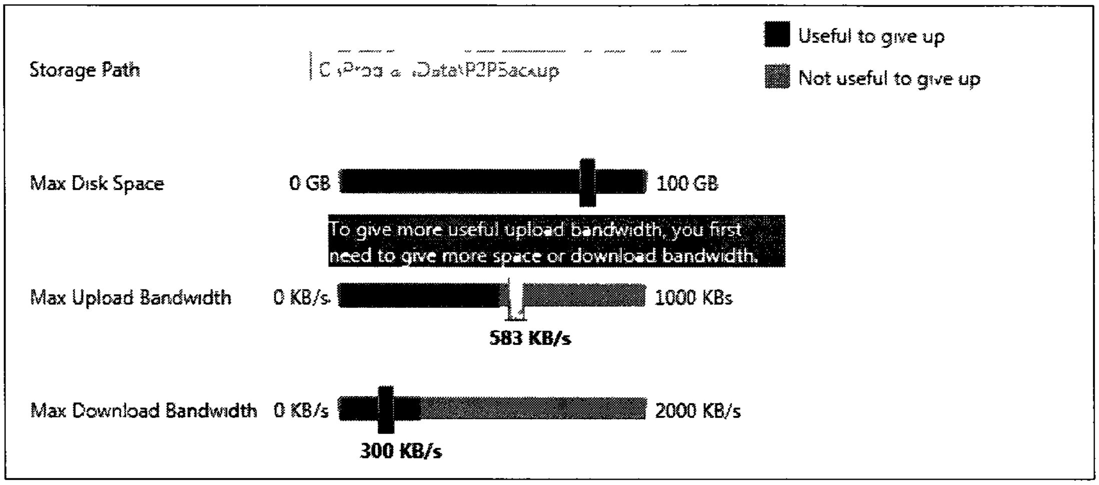
2.3.2 Combinatorial Aspects of the Market: Bundle Constraints

The first challenge regarding the hidden market design for this application is the combinatorial nature of the market, i e , the problem that only bundles of resources are useful to the system In general, the free online backup space increases when the users increase one of their sliders However, this is only true for a subset of possible slider positions In particular, if a user keeps increasing one slider towards the maximum while the other two sliders are relatively low, at some point the online

to control their supply In a deployed system, the settings window would initially show the left side of the window and only upon clicking an "advanced" button would the right side appear backup space on the left might stop increasing For example, if users limit their upload bandwidth to 5 KB/s, then increasing their space supply from 50 GB to 100 GB should not increase their online backup space We would simply never store 100 GB on these users' hard disks because 5 KB/s would not be enough to have a reasonable retrieval rate for all of these file pieces Thus, for the system to use the whole supply of 100 GB, the users would first have to increase their supply of bandwidth An analogous argument holds true for other combinations of resources For example, if a user wanted to give a lot of upload bandwidth but keep the supply of space low, then at some point giving more bandwidth would not be useful Again, to make use of the download bandwidth, the system would need to store many file pieces on that user's computer which is not possible given the current low limit on space [6]

Because of these "bundle constraints", we need users to respect certain supply ratios when choosing their supply settings To provide the users with some visual information regarding how much supply of a resource is "useful to the system" given the current other slider settings, we augmented the traditional slider UI element, building the new slider control shown in Figure 2 3 The sliders are colored blue and gray, and the legend on the top right of the window explains the color coding In the blue region, slider movements have an effect on the online backup space because setting the slider to any position inside that region means that the system can effectively use all of the supplied resource The gray region of the slider is the region where slider movements no longer have an effect on the user's online backup space because giving

[6] These bundle constraints only apply to space, upload and download bandwidth For "availability" there is no minimum or maximum supply that is useful, independent of the other resources



Storage Path C \Proa a \Data\P2PBackup

Max Dısk Space 0 GB 100 GB

To give more useful upload bandwidth, you first need to give more space or download bandwidth.

Max Upload Bandwıdth 0 KB/s 1000 KBs 583 KB/s

Max Download Bandwıdth 0 KB/s 2000 KB/s 300 KB/s

Fıgure 2 3 The new slıder control provıdes an ındırect vısualızatıon of the bundle constraınts When a user provıdes more of one resource than ıs useful to the system, he gets notıfied vıa a small popup wındow

that much of the resource ıs "not useful to the system," gıven the other settıngs Because the colors and the legend mıght be dıfficult to understand or be overlooked, we also notıfy the user once the slıder ıs moved from the blue ınto the gray regıon wıth a small pop up message that dısappears once the mouse button ıs released (see Fıgure 2 3)

The color-coded slıders provıde the user wıth all the necessary ınformatıon about the bundle constraınts When one slıder ıs moved down, the blue regıons on the other two slıders first stay the same and eventually decrease Analogously, when one slıder ıs moved up, the blue regıons on the other two slıders first ıncrease and eventually stop ıncreasıng If a user sets the slıders ın the same ratıos as the system-wıde usage of all resources, they are always ınsıde the blue regıons However, requırıng thıs exact ratıo from all users ıs too restrıctıve, ıgnorıng the system's flexıbılıty ın allocatıng work For example, the system can allocate more repaır and testıng operatıons to users that prefer to give up lots of bandwidth instead of space Furthermore, the system can estimate how often certain users access their backups and then send file fragments from "cold backups" to users who prefer to give up more storage space rather than bandwidth To maximize overall efficiency, we make use of this flexibility, and allow every user to supply different ratios of their resources, within certain bounds of the system-wide ratios In Section 2 4 2, we explain this concept more formally

2.3.3 Exposing/Displaying Market Prices

Because the UI gives users some freedom in choosing their resource supply, we must price the resources correctly In our system, prices are updated daily depending on aggregate demand and supply, moving the system into equilibrium over time Without updating prices, we might have a supply shortage for some resources For example, many users might decide to give lots of disk space and little bandwidth To counteract a shortage of bandwidth, we would increase the price of bandwidth, incentivizing users to give more bandwidth instead of space But for this mechanism to work, it is necessary that prices are at least indirectly exposed to users, so they can react and change their supply settings For example, if the price for upload bandwidth went up relative to download bandwidth, then users might benefit from increasing their upload bandwidth supply a little and in return decreasing their download bandwidth supply a lot

However, users do not expect monetary transactions in a backup application, which also renders "prices" an unnatural concept This is why we have chosen to hide the prices in the UI as much as possible In our UI, a user can "experience" the relative prices indirectly by moving the sliders while observing the bar chart on the left If a user moves a slider a little and the bar chart only changes a little, this means that the current price for that resource is relatively low If a user moves a slider a little and the bar chart changes a lot, this means that the current price for that resource is relatively high This is one of the essential aspects of this hidden market UI it allows us to communicate the current market prices to a user in a non-explicit way In particular, users can be unaware of the price-based market underlying the backup system, and yet over time they will notice that for some resources they get more in return than for others They can then choose the supply combination that is currently best given their preferences Note that one of the market design goals was to implement a very simple pricing system to provide even non-expert users with a seamless interaction We achieve this, despite the bundle constraints, by providing the users with a linear interaction with the system, as long as they move the sliders within the blue regions (the bar chart on the left moves up and down linearly when a user moves one of the sliders on the right) More specifically, we expose simple, linear prices to the users, and take care of the bundle constraints by restricting the choices they can make in the UI using the slider controls

2.4 Market Design & Economic Model

In this section we introduce a formal economic model to describe the market design in detail and to allow for a theoretical economic analysis of the properties of the P2P market system At all times, the model is formulated such as to represent the implemented system as closely as possible

2.4.1 User Preferences

The economy comprises I users who are simultaneously suppliers and consumers The set of commodities in the market is denoted $L = \{S, U, D, A, B, \Sigma, R\}$ The first four commodities are space (S), upload bandwidth (U), download bandwidth (D), and availability (A), which are the resources that users supply The last three commodities are backup service (B), storage service (Σ), and retrieval service (R), which are the services that users consume By slightly abusing notation, we sometimes use S, U, D, etc as subscripts, and sometimes they denote the resource domain, e g , for a particular amount of upload bandwidth u we require that $u \in U$ Each user i has a fixed endowment of the supply resources (defined by the user's hard drive and Internet connection), denoted $w_i = (w_{iS}, w_{iU}, w_{iD}, w_{iA}) \in S \times U \times D \times [0, 1]$

The next aspect of the model is driven by our UI Via the sliders, the user selects upper bounds for the supply vector, which we denote $X_i = (X_{iS}, X_{iU}, X_{iD}, X_{iA})$ In return for the supply X_i, the user interface shows the user the maximum demand of services, denoted $Y_i = (Y_{iB}, Y_{i\Sigma}, Y_{iR})$ In Figure 2 2 the user has currently chosen $X_{iS} = 80\ 8GB$, $X_{iU} = 400KB/s, X_{iU} = 300KB/s$ and $X_{iA} = 0\ 5$ as the maximum supply vector

At any point in time, a certain set of resources from the user are being used, always less than X_i, and a certain set of services is being demanded We denote user i's current supply as $x_i = (x_{iS}, x_{iU}, x_{iD}, x_{iA})$, and analogously user i's current demand for services as $y_i = (y_{iB}, y_{i\Sigma}, y_{iR})$ The user does not choose x_i and y_i directly via the UI Instead, the server chooses x_i (obeying the bound X_i) such that user i can afford the current demand y_i which the user simply chooses by backing up files or retrieving them. Note that the UI displays the user's consumption vector in an aggregated way, i.e., instead of listing the services backup, storage, and retrieval separately, we simply display the currently used online backup space (= $17.28GB$ in Figure 2.2) and the maximum online backup space that user could consume (= $33.5GB$ in Figure 2.2).

In practice, users have a certain cost for opening the settings window and adjusting the settings. Instead of modeling this cost factor directly, we assume that when users open their settings window, they are planning ahead for the whole time period until they plan to open the settings window the next time. While a user might currently consume y_i, he plans for consuming up to Y_i the next time he opens the settings window. He then selects the supply vector X_i that he is willing to give up to get this Y_i. The user cares about how large the bounds on his supply are, because he has negative utility for giving up his resources. To make this more formal, we let $K_i = w_i - X_i$ with $K_i \in S \times U \times D \times A$, denote the vector of resources that the user *keeps*, i.e., his endowment minus the supply he gives up. Note that any changes to X_i translate into changes for K_i and vice versa because the endowment vector w_i is fixed. We only introduce K_i to define a preference relation that is monotone in all components, but we will use the supply vector X_i going forward. We can now specify the user's preference relation over all the resources he keeps, and the services he consumes: $\succeq_i (K_{iS}, K_{iU}, K_{iD}, K_{iA}, Y_{iB}, Y_{i\Sigma}, Y_{iR})$. We make the following assumptions which are all standard in economics (cf. [65], chapters 1-3).

Assumption 1. *Each user's preference relation* $\succeq_i (K_{iS}, K_{iU}, K_{iD}, K_{iA}, Y_{iB}, Y_{i\Sigma}, Y_{iR})$ *is (i) complete, (ii) transitive, (iii) continuous, (iv) strictly convex, and (v) monotone.*

Strict convexity requires strictly diminishing marginal rates of substitution between two goods, ı e , we need to compensate a user more and more wıth one good as we take away 1 unıt of another good Thıs ıs a reasonable assumptıon because ıt represents a general preference for dıversıficatıon Monotonıcıty means that all commodıtıes are “goods”, ı e , ıf we gıve users more of any of the commodıtıes, they are at least as well off as before [7] Gıven complete, transıtıve, and contınuous preferences, there exısts a utılıty functıon $u_\imath(K_\imath, Y_\imath) = u_\imath(K_{\imath S}, K_{\imath U}, K_{\imath D}, K_{\imath A}, Y_{\imath B}, Y_{\imath\Sigma}, Y_{\imath R})$ that represents the preference relatıon and thıs utılıty functıon ıs contınuous (cf [65], p 47)

As mentıoned before, the only resource that ıs not subȷect to the combınatorıal bundle constraınts, ıs *avaılabılıty* as long as the user’s avaılabılıty ıs larger than zero, the other resources can be used To sımplıfy the economıc model and prıcıng of resources, we ıntroduce three new *composıte resources* $\overline{S}, \overline{U}$, and $\overline{D}$, ıncorporatıng the user’s avaılabılıty ınto the other resources ın the followıng way

- $\overline{X}_{\imath\overline{U}} \in \overline{U} = X_{\imath U} \quad X_{\imath A} \quad 24 \quad 60 \quad 60$
- $\overline{X}_{\imath\overline{D}} \in \overline{D} = X_{\imath D} \quad X_{\imath A} \quad 24 \quad 60 \quad 60$
- $\overline{X}_{\imath\overline{S}} \in \overline{S} = \overline{\varphi}(X_{\imath S}, X_{\imath A}) \approx X_{\imath S} \quad X_{\imath A}$ overhead factor

Note that thıs notatıon denotes *composıte* and *not* vector quantıtıes The definıtıons for the composıte resources upload and download bandwıdth are straıghtforward we multıply the bound on bandwıdth the user supplıes (e g , 300 KB/S) wıth the user avaılabılıty $\in [0, 1]$ and then multıply ıt wıth 24 hours, 60 mınutes and 60

[7] Note that we do not assume *strıct* monotonıcıty because we wıll later assume that servıce products are perfect complements, whıch vıolates strıct monotonıcıty of preferences We dıscuss thıs ın more detaıl ın Sectıon 2 5

seconds, to calculate how many KBs we can actually send to this user per day The definition of $\overline{X}_{i\overline{S}}$ is a little more intricate because the user's availability does not enter linearly into the calculation However, it enters monotonically, i e , more availability is always better Here, it suffices to know that the server can compute this function $\overline{\varphi}$ and convert a user's space and availability supply into the new composite resource

We can now define user i's supply vector for the three composite resources $\overline{X_i} = (\overline{X_{i\overline{S}}}, \overline{X_{i\overline{U}}}, \overline{X_{i\overline{D}}})$ The advantage of using these "availability-normalized" composite resources is that now, the supply from different users with different availabilities is comparable For example, 1 unit of $\overline{S}$ from user i with availability 0 5 is now equivalent to 1 unit of $\overline{S}$ from user j with availability 0 9 Obviously, internally user i has to give much more space to make up for his lower availability, but in terms of bookkeeping, we can now operate directly with composites We define the aggregate supply vector for the composite resources as $\overline{X} = \sum_i \overline{X_i}$, and analogously for Y, $\overline{x}$ and y We make the following well-known observation (cf [65], chapter 3) that will be useful later

Observation 1 *The individual and aggregate supply and demand functions $\overline{X_i}$, Y_i, $\overline{X}$, and Y are homogeneous of degree zero*

2.4.2 Production Functions and Slack Constraints

We have already mentioned the important role of the server in our market, i e , that of combining resources from different suppliers into a valuable bundle Note that the server is in fact the *only* producer in the market One can think of this as if every user had access to the same production technology to convert input resources into services This is crucial for our model and the economic analysis, because it allows us to define an *exchange economy* where the users only exchange factor inputs, despite the fact that production is happening in the market (cf [65], pp 582-584) Thus, for each service, we have one production function that defines how many input resources are needed to produce one unit of that service

- Backup f^B $\overline{S} \times \overline{U} \times \overline{D} \to B$
- Storage f^Σ $\overline{S} \times \overline{U} \times \overline{D} \to \Sigma$
- Retrieval f^R $\overline{S} \times \overline{U} \times \overline{D} \to R$

These production functions are defined via the implementation of our system, i e , the particular production technology that we implemented For example, they are defined via the particular erasure coding algorithm that is being used, by the frequency of repair operations, etc Thus, we can now specify a series of properties that these production functions guarantee due to our implementation

System Property 1 *(Fixed Production Functions) Production functions are fixed and the same for all users*

System Property 2 *(Additivity) The production functions are additive, i e ,* $\forall l \in \{B, \Sigma, R\}$ *and for any two resource vector* $\vec{x_1}$ *and* $\vec{x_2}$ $f^l(\vec{x_1} + \vec{x_2}) = f^l(\vec{x_1}) + f^l(\vec{x_2})$

System Property 3 *(CRTS) The production functions exhibit constant returns to scale (they are homogeneous of degree 1), i e ,* $\forall l \in \{B, \Sigma, R\}$*, for any* $\vec{x}$*, and* $\forall k \in \Re$ $f^l(k \ \vec{x}) = k \ f^l(\vec{x})$

System Property 4 *(Bijectivity) Each production function is bijective, and thus we can take the inverses*

- $f^{B^{-1}} \quad B \rightarrow \overline{S} \times \overline{U} \times \overline{D}$
- $f^{\Sigma^{-1}} \quad \Sigma \rightarrow \overline{S} \times \overline{U} \times \overline{D}$
- $f^{R^{-1}} \quad R \rightarrow \overline{S} \times \overline{U} \times \overline{D}$

Property 1 holds because the server is the only producer, and because of the way we have defined the composite resources, with any differences between the users' availabilities already considered Properties 2 (Additivity) and 3 (CRTS) hold because the erasure coding algorithm (which defines the production technology) exhibits these properties [8] Property 4, the bijectivity of production, holds, because for each service unit, there is only one way to produce it For example, to backup one file fragment, the erasure coding algorithm tells us exactly how many supplier fragments we need, and the server tells us how much repair and testing traffic we can expect on average per fragment Furthermore, it is obvious that small changes in the input of the inverse production functions result in small changes in the output More formally

System Property 5 *(Continuity) The inverse production functions are continuous*

Given the inverse functions for the individual services backup, storage, and retrieval, we can define an inverse function for a three-dimensional service vector $(b, \sigma, r) \in B \times \Sigma \times R$

$$f^{-1}(b, \sigma, r) = f^{B^{-1}}(b) + f^{\Sigma^{-1}}(\sigma) + f^{R^{-1}}(r) \qquad (2\ 1)$$

[8] Note that these two properties only hold approximately and not exactly, and only for file sizes above a certain threshold (approximately 1MB) Very small files are an exception and need special treatment in the implementation, because they are more expensive to be produced (again due to the erasure coding) We take care of this in the implementation by charging users more when they are backing up small files (essentially we have two sets of prices, one for normal files and one for small files)

Given a demand vector y, we use $f^{-1}(y)$ to refer to the vector of supply resources that are necessary to produce y Furthermore, we use $f_{\overline{S}}^{-1}(y)$, $f_{\overline{U}}^{-1}(y)$, and $f_{\overline{D}}^{-1}(y)$ to refer to the individual amounts of supply resources that are necessary to produce y

We now formalize the flexibility we give our users in setting different ratios of their supplied resources Because of the bundle constraints, a user cannot reduce his supply of resource k towards zero without affecting the supply of his other resources To determine what ratios are acceptable, i e , useful to the system, we look at the system-wide usage of each resource k, i e $f_k^{-1}(y)$ Certainly, if a user provides his resources in the same ratios as the system-wide usage, then all of his supply is usable However, because the system has flexibility in allocating different kinds of work (repair/testing traffic vs "cold backups" vs "hot backups"), we can let the users' supply ratios deviate from the system-wide ratios to a limited degree We let $\gamma > 1$ denote the amount of *slack* we allow users when setting their supply-side sliders The corresponding *slack constraints*, lower-bounding the supply for resource k, constitute another system property

System Property 6 *(Slack Constraints) Given slack factor γ, for each resource $k \in \{\overline{S}, \overline{U}, \overline{D}\}$, the user interface enforces the following minimum ratios of supplied resources*

$$\forall i, \forall l \in \{\overline{S}, \overline{U}, \overline{D}\} \setminus \{k\} \quad \frac{\overline{X}_{ik}}{\overline{X}_{il}} \geq \frac{1}{\gamma} \frac{f_k^{-1}(Y)}{f_l^{-1}(Y)} \qquad (2\ 2)$$

Note that the UI does not actually limit the range of the sliders according to the slack constraints If a user chooses to supply too little of one resource such that a slack constraint is violated, then the system only uses/considers the maximum amount of the others resource such that the slack constraint binds The UI visualizes this to the users via the blue regions, which are effectively indirect representation of the slack constraints, showing the user which settings are useful to the system Thus, Equation 2 2 correctly models the slack constraints If we actually limited the range of the sliders, then making larger changes with the sliders (which is necessary to explore the settings space) would be too tedious

In our implementation, we set $\gamma = 2$ Thus, to give an example, if the system-wide usage ratio of space to upload bandwidth were 6, then each user would have to choose his individual settings with a ratio of space to upload bandwidth of at least $6\ \frac{1}{2} = 3$, and the ratio of of upload bandwidth to space would have to be at least $\frac{1}{6}\ \frac{1}{2} = \frac{1}{12}$ How large we can set γ in practice depends on how flexible the system is in terms of allocating work (i e , how many "cold" vs "hot" backups there are, how much repair and testing traffic there is, etc) In practice, the slack factor γ would have to be adjusted over time, when the distribution of work changes This process could be automated, but here we are not going into the details of this process

While every individual user is free to choose any supply setting within the slack constraints, of course the aggregate supply of each resource must always be large enough to satisfy current aggregate demand But if every user chooses a supply setting such that the same slack constraint binds (e g , every user minimizes his supply of upload bandwidth), then the system does not have enough supply of the corresponding resource This is were the pricing algorithm comes into play by regularly updating market prices according to current aggregate demand and supply, we balance the market such that different users will indeed supply *different* ratios of their resources

We discuss this aspect in more detail in Section 2.5 where we also prove that for any set of user preferences, there always exists a price vector that balances the market and guarantees enough supply of each individual resource.

2.4.3 Prices and Flow Constraints

In Section 2.3.2, we have explained how we display the bundle constraints to the users in the UI. The UI automatically enforces that the users only choose supply vectors that satisfy the slack constraints (cf. System Property 6) and this enables us to support an equilibrium with linear prices. We use $p = (p_{\overline{S}}, p_{\overline{U}}, p_{\overline{D}})$ for the prices for supplied composite resources, and $q = (q_B, q_\Sigma, q_R)$ for the demanded services. We require that in steady state, i.e., when a user has been online long enough, he can pay for his consumption with his supply. In other words, his flow of supplied resources must be high enough to afford the flow of consumed services. We can express this *flow constraint* formally:

$$\overline{X}_i \cdot p = Y_i \cdot q \tag{2.3}$$

At the same time, the server allocates enough work to user i such that the user's current supply $\overline{x}_i$ is enough to pay for the demand y_i, which leads to a second flow constraint:

$$\overline{x}_i \cdot p = y_i \cdot q \tag{2.4}$$

We make the following assumption regarding the usage of resources in the system.

Assumption 2 *(Closed System & No Waste) We assume a closed system where no resources are entering or leaving the market, and we assume that no resources* *are wasted Thus, the amount of resources required to produce the current aggregate demand is always equal to the current aggregate resource supply, i e* $f^{-1}(y) = \overline{x}$

Proposition 1 *Given a closed system and no waste of resources (Assumption 2), and given that production functions are additive (System Property 2), the payments from consumer i to the server must equal the payments from the server to the corresponding suppliers, i e*

$$y_i \; q = f^{-1}(y_i) \; p \tag{2 5}$$

Proof From the flow constraint in Equation 2 4 we know that $\overline{x}_i \; p = y_i \; q$ By summing over all users on both sides of the equation it follows that $\overline{x} \; p = y \; q$ Given Assumption 2, we know that $f^{-1}(y) = \overline{x}$ By plugging this into the previous equation, we get $f^{-1}(y) \; p = y \; q$ From the additivity of the production functions we know that this is equivalent to $\sum_i f^{-1}(y_i) \; p = \sum_i y_i \; q$ Because each transaction is treated equally in the system (every user is payed the same for the same resources), it follows that $f^{-1}(y_i) \; p = y_i \; q$ □

Using Proposition 1, we can now re-write the flow constraints for user i as

$$\overline{X_i} \; p = f^{-1}(Y_i) \; p \quad \text{and} \quad \overline{x_i} \; p = f^{-1}(y_i) \; p \tag{2 6}$$

Thus, from now on, we can omit the price vector q for demanded services and only need to consider price vector p[9], i e , all what matters are the relative prices of the supply resources Remember that the UI automatically calculates and adjusts

[9] Going forward, please remember that multiplications with p are always dot products, and thus p showing up on the left and the right side of an equation does not cancel out

the maximum demand vector Y_i for user i based on the user's supply bound $\overline{X}_i$ In practice, the maximum income is divided by the current average income of the user, and the resulting factor is multiplied with the user's current demand, giving us the maximum demand the user can afford

System Property 7 *(Linear Prediction for Individual Demand) The system uses a linear demand prediction model for the calculation of a user's maximum demand* Y_i

$$Y_i = \frac{\overline{X}_i \ \ p}{\overline{x}_i \ \ p} \ \ y_i = \lambda_i \ \ y_i$$

We make the following simplifying assumption

Assumption 3 *(Linear Prediction for Aggregate Demand) We assume that with a large number of users, a linear demand prediction is also correct for the aggregate demand vectors, i e*

$$\exists \lambda \ \ Y = \lambda \ \ y$$

This assumption is justified because in practice, such a system would have a large number of users Let n denote the number of users in the economy, let $Y^n = \sum_{i=1}^{n} Y_i$, $y^n = \sum_{i=1}^{n} y_i$, and let $\mu(\lambda_i)$ denote the mean of the distribution of the λ_i's Given that the λ_i's are independent from the y_i's, it follows from the strong law of large numbers, that if the number of users n is large enough, then Y^n is linearly predictable by $\mu(\lambda_i) \ \ y^n$ along each dimension to any additive error More specifically, for any ε and $\delta \geq 0$, for large enough n

$$Pr[||Y^n - \mu(\lambda_i) \ \ y^n|| \leq \varepsilon] \geq 1 - \delta$$

2.5 Equilibrium Analysis

A real-world instance of the P2P backup application would have thousands if not millions of users Thus, the underlying market would be large enough so that no individual user had a significant effect on market prices Consequently, users can be modeled as price-taking users and a *general equilibrium* model is suitable to analyze this market Here we analyze a static equilibrium in which all users adjust their supply bounds to reach target demand bounds, i e , whenever the price vector p is updated, user i chooses $\overline{X}_i(p)$ and $Y_i(p)$ such as to maximize his utility While a user does not choose $\overline{x}_i$ (user i's supply that is currently used) and y_i (user i's current demand vector) directly via the UI, these quantities nevertheless depend on current prices, though indirectly, because $\overline{x}_i \leq \overline{X}_i$ and $y_i \leq Y_i$ Thus, while current demand and supply vectors $\overline{x}_i$ and y_i will vary much less with price changes, we must still model them as being dependent on prices, and we use $\overline{x}_i(p)$ and $y_i(p)$ to reflect that Throughout this section, we assume that System Properties 1 through 7 and Assumptions 1 through 3 hold

2.5.1 The Buffer Equilibrium

We begin this section by asking the question what the target equilibrium should be when we are updating prices Note that there only is an equilibrium pricing problem in the first place because we give users the freedom to supply different ratios of resources Without any slack, the UI would enforce that every user supplied the resources in the same ratios as system-wide demand for resources, and thus price changes would have no effect But because we give our users the freedom to choose different supply ratios, we must update prices over time, to avoid situations where we do not have enough supply for a resource to satisfy current demand But what should be our target?

A standard equilibrium concept in general equilibrium theory is the Walrasian equilibrium, which requires that demand equals supply such that the market clears Certainly we want to have enough supply to satisfy current demand, i e

$$\overline{x}(p) = f^{-1}(y(p))$$

But remember that users are not continuously adjusting $\overline{x}_i$, and as a consequence, the system will be *out of equilibrium* most of the time Thus, our goal should not be to clear the market in equilibrium, but instead to always have some excess supply of all resources, to make sure we can satisfy any demand even out of equilibrium The larger the "buffer" between the current demand of resources, i e , $f^{-1}(y)$, and the maximum supply of resources, i e , $\overline{X}$, the safer the system, i e , the more "out of equilibrium" it can cope with before running into trouble We will use this "size of the supply-side buffer" repeatedly and thus we define it more formally

Definition 1 *(Size of the Supply-side Buffer for a Resource) The size of the supply-side buffer for resource l is the ratio of maximum supply to current demand for that resource, and we denote this buffer with $\mathbb{B}_l(p)$*

$$\mathbb{B}_l(p) = \frac{\overline{X(p)}_l}{f_l^{-1}(y(p))} \qquad (2\ 7)$$

If we assume that the supply and demand for the individual resources have the same variance, then the best we can do to maximize the safety of the system out of equilibrium, is to maximize the size of the buffer across all three resources [10] This naturally leads to the definition of the overall size of the supply-side buffer

Definition 2 *(Overall Size of the Supply-side Buffer) The size of the overall supply-side buffer $\mathbb{B}(p)$ is the smallest supply-side buffer across all resources, i e*

$$\mathbb{B}(p) = \min_{l \in \{\overline{S},\overline{U},\overline{D}\}} \mathbb{B}_l(p) \tag{2 8}$$

Now the question is, which price vector maximizes the overall supply-side buffei It is intuitive, that to maximize the overall supply-side buffer, the individual buffers must all be equal (otherwise we might update prices to decrease the largest buffer and increase the samllest buffer) This naturally leads us to the following definition of a "buffer equilibrium"

Definition 3 *(Buffer Equilibrium [Version 1]) A Buffer equilibrium is a price vector $p = (p_{\overline{S}}, p_{\overline{U}}, p_{\overline{D}})$, an aggregate supply vector $\overline{X}(p)$, and an aggregate current demand vector $y(p)$, such that the individual supply-side buffers are the same across all resources, i e*

$$\mathbb{B}_{\overline{S}}(p) = \mathbb{B}_{\overline{U}}(p) = \mathbb{B}_{\overline{D}}(p) \quad \Leftrightarrow \quad \frac{\overline{X}_{\overline{S}}(p)}{f_{\overline{S}}^{-1}(y(p))} = \frac{\overline{X}_{\overline{U}}(p)}{f_{\overline{U}}^{-1}(y(p))} = \frac{\overline{X}_{\overline{D}}(p)}{f_{\overline{D}}^{-1}(y(p))} \tag{2 9}$$

It seems very reasonable to assume that, as we decrease the price for one resource k, the supply-side buffers for the other two resources will increase Decreasing p_k makes it less attractive for the users to supply resource k, and makes it relatively more attractive to supply the other resources If we make this assumption more

[10] If we have specific information about the variance in the supply and demand of certain resources, we would want to target higher buffers on the resources with high variance and lower buffers on resources with low variance This can easily be incorporated and would only lead to a slightly different equilibrium definition

formally, we can ındeed prove that for the supply-sıde buffer to be maxımal, the system must be ın a buffer equılıbrıum, thus ȷustıfyıng the buffer equılıbrıum as a desırable target concept

Assumptıon 4 *(Resource Buffers are Gross Substıtutes) We assume that the ındıvıdual buffer functıons* $\mathbb{B}_l(p)$ *satısfy the gross substıtutes condıtıon, ı e , whenever* p' *and* p *are such that, for some* k*,* $p'_k > p_k$ *and* $p'_l = p_l$ *for* $l \neq k$*, we have* $\mathbb{B}_l(p') < \mathbb{B}_l(p)$ *for* $l \neq k$ [11]

Propositıon 2 *Gıven Assumptıon 4 (Resource Buffers are Gross Substıtutes), when the overall supply-sıde buffer* $\mathbb{B}(p)$ *ıs maxımal, then the market has reached a buffer equılıbrıum*

Proof We present a proof by contradıctıon Let's assume that p ıs a prıce vector such that the overall supply-sıde buffer ıs maxımal, but where the resource buffers are not the same across all resources as they must be ın the buffer equılıbrıum Assume that $k = \arg\max_{l \in \{\overline{S},\overline{U},\overline{D}\}} \mathbb{B}_l(p)$, ı e , the buffer for resource k ıs maxımal across all resources Now, we consıder prıce vector p' where we have decreased the prıce of resource k slıghtly and kept the prıces of the other resources constant, ı e , ${p'}_k < p_k$ and ${p'}_l = p_l \; \forall l \neq k$ Gıven that the ındıvıdual resource buffers satısfy Assumptıon 4, we know that $\mathbb{B}_l(p') > \mathbb{B}_l(p)$, and due to homogeneıty of degree zero, ıt also follows that $\mathbb{B}_k(p') < \mathbb{B}_k(p)$, ı e , the resource buffer sıze for k has decreased and

[11] Note that thıs assumptıon ıs sımılar to the more standard assumptıon that the excess demand functıon satısfies the gross substıtute property, however, they are not equıvalent We assume that, as we decrease the prıce on one resource, the *ratıo* between supply and demand for all other resources wıll ıncrease, whıle the standard gross substıtutes assumptıon states that the *dıfference* between supply and demand for all other resources wıll ıncrease Neıther assumptıon ımplıes the other, although both can be true sımultaneously

both other resource buffer sizes have increased. Because of the continuity of users' preferences (Assumption 1) and the continuity of the inverse production function (System Property 5), it follows that $\overline{X}(p)$ and $f^{-1}(y(p))$ are continuous, and thus we can always find a small enough price change from p to p', such that the buffer for resource k is still maximal, but in the process we have increased the buffers for the other two resources. Thus, the overall supply-side buffer is larger for p' than it was before, i.e., $\mathbb{B}(p') > \mathbb{B}(p)$ which violates our assumption that the supply-side buffer with price vector p is maximal, which leads to a contradiction and completes the proof. □

We have just shown that when the overall supply-side buffer is maximal, then the market has reached a buffer equilibrium. One concern might be that this does not automatically imply that the supply-side buffer will be maximal in *every* buffer equilibrium. However, we will show in Section 2.5.4 that under certain assumptions, the buffer equilibrium is unique, which removes this concern and implies that the buffer equilibrium is indeed a good target concept. Note that we truly believe that Assumption 4 is satisfied in our domain, and thus, the overall supply-side buffer is indeed maximal in the buffer equilibrium. However, we do not need this assumption going forward. We only used it to provide a formal motivation for the introduction and use of the buffer equilibrium concept, but all statements in the remainder of this chapter are also true for the buffer equilibrium, without this assumption.

We now offer an alternative definition of the buffer equilibrium which relates it to the well-known concept of a Walrasian equilibrium.

Definition 4 *(Buffer Equilibrium [Version 2]) A Buffer equilibrium is a price vec-*

tor $p = (p_{\overline{S}}, p_{\overline{U}}, p_{\overline{D}})$*, an aggregate maxımum supply vector* $\overline{X}(p)$*, and an aggregate maxımum demand vector* $Y(p)$*, such that*

$$\overline{X}(p) = f^{-1}(Y(p))$$

ı e , ıt ıs a Walrasıan equılıbrıum defined on the supply and demand bounds chosen by the users

It ıs easy to show that the two defınıtıons for the buffer equılıbrıum are equıvalent

Lemma 1 *Gıven Assumptıon 3 (Lınear Predıctıon for Aggregate Demand), the 1 and 2 defınıtıons of the Buffer Equılıbrıum are equıvalent, ı e*

$$\mathbb{B}_{\overline{S}}(p) = \mathbb{B}_{\overline{U}}(p) = \mathbb{B}_{\overline{D}}(p) \quad \Leftrightarrow \quad \overline{X}(p) = f^{-1}(Y(p))$$

Proof We begın by showıng the "$\Rightarrow$" dırectıon

If $\mathbb{B}_{\overline{S}}(p) = \mathbb{B}_{\overline{U}}(p) = \mathbb{B}_{\overline{D}}(p)$ then

$$\exists \lambda > 1 \text{ s t } \forall l \;\; \overline{X}_l(p) = \lambda \;\; f_l^{-1}(y(p))$$

Now, due to Assumptıon 3 we know that $\exists \delta \;\; Y(p) = \delta \;\; y(p)$ Thus

$$\Rightarrow \forall l \;\; \overline{X}_l(p) = \lambda \;\; f_l^{-1}(\frac{1}{\delta}Y(p)) \tag{2 10}$$

$$\Rightarrow \forall l \;\; \overline{X}_l(p) = \lambda \;\; \frac{1}{\delta} \;\; f_l^{-1}(Y(p)) \tag{2 11}$$

$$\Rightarrow \forall l \;\; \overline{X}_l(p) = \lambda^* \;\; f_l^{-1}(Y(p)) \quad \text{for } \lambda^* = \lambda \;\; \frac{1}{\delta} \tag{2 12}$$

$$\Rightarrow \overline{X}(p) = \lambda^* \;\; f^{-1}(Y(p)) \tag{2 13}$$

From the flow constraınts (Eqn 2 6) we also know that

$$\overline{X}(p) \;\; p = f^{-1}(Y(p)) \;\; p \tag{2 14}$$

Equations (2 13) and (2 14) can only both be true if $\lambda^* = 1$ Thus, it follows that

$$\overline{X}(p) = f^{-1}(Y(p))$$

The "⇐" direction is even simpler to show

$$\overline{X}(p) = f^{-1}(Y(p)) \tag{2 15}$$

$$\Rightarrow \overline{X}(p) = f^{-1}(\lambda \quad y(p)) \tag{2 16}$$

$$\Rightarrow \overline{X}(p) = \lambda \quad f^{-1}(y(p)) \tag{2 17}$$

$$\Rightarrow \mathbb{B}_{\overline{S}}(p) = \mathbb{B}_{\overline{U}}(p) = \mathbb{B}_{\overline{D}}(p) \tag{2 18}$$

Equation 2 16 follows because of Assumption 3 (Linear Prediction for Aggregate Demand) Equation 2 17 follows from System Properties 3 and 4 (Production functions satisfy CRTS and are bijective) □

2.5.2 Equilibrium Existence

In this section, we prove that a buffer equilibrium exists in our model We let $\overline{L} = \{\overline{S}, \overline{U}, \overline{D}\}$ and we use l to index a particular composite resource We define the vector-valued *relative-buffer function* $Z(p)$ which measures the relative buffer for each individual resource in the following way

$$Z_l(p) = \frac{\overline{X}_l(p)}{f_l^{-1}(y(p))} - \left(\frac{\sum_k \frac{\overline{X}_k(p)}{f_k^{-1}(y(p))}}{|\overline{L}|}\right) \tag{2 19}$$

In words, the first term represents the supply to demand ratio of the particular good l The second term represents the average supply to demand ratio, in our case averaged over the three goods storage space, upload and download bandwidth Thus, $Z_l(p)$

represents how far the "buffer" between supply and demand for good l is away from the average buffer. We have reached a buffer equilibrium when the buffer is the same for all goods, i.e., when

$$Z(p) = 0$$

Lemma 2 *Given that users' preferences are strongly monotone with respect to supply resources, the relative-buffer function $Z(\,)$ has the following property: If $p^n \to p$, with $p \neq 0$ and $p_k = 0$ for some k, then for n sufficiently large*

$$\exists l \quad Z_l(p^n) > Z_k(p^n)$$

Proof Because $p \neq 0$, for n large enough, there exists a resource l such that $p_l^n > 0$. As the price of resource $k \in \{\overline{S}, \overline{U}, \overline{D}\}$ goes towards zero, due to users' strictly convex and strongly monotone preferences for supply resources, they will supply less and less of k, and supply more of the other resources instead, at least of resource l whose price is bounded away from zero. However, because of the slack constraints, the users cannot reduce their supply of resource k towards zero, or increase their supply of resource l arbitrarily high. Let $\gamma > 1$ denote the slack factor we allow users when setting their preferences. The corresponding slack constraints (see System Property 6), lower-bounding the supply for resource k, are

$$\forall l \in \overline{L} \setminus \{k\} \quad \overline{X_{ik}(p^n)} \geq \frac{1}{\gamma} \frac{f_k^{-1}(y(p^n))}{f_l^{-1}(y(p^n))} \overline{X_{il}(p^n)}$$

As $p^n \to p$ with $p \neq 0$ and $p_k = 0$, for n large enough, p_k^n will be sufficiently close to zero, such that each user i chooses to supply the minimal amount of resource k

that is possible Thus, at least with respect to *one* of the other resources l or m, the slack constraint will be binding, i e ,

$$\forall i \quad \overline{X_{ik}(p^n)} = \frac{1}{\gamma}\,\frac{f_k^{-1}(y(p^n))}{f_l^{-1}(y(p^n))}\,\overline{X_{il}(p^n)} \quad \vee \quad \overline{X_{ik}(p^n)} = \frac{1}{\gamma}\,\frac{f_k^{-1}(y(p^n))}{f_m^{-1}(y(p^n))}\,\overline{X_{im}(p^n)}$$

This does not mean that the slack constraint will be binding for the same resource l or m for every user In fact, it is possible that user i will minimize his supply of resources k and l, while user j minimizes his supply of resources k and m However, because every user contributes least to the supply-side buffer for resource k, this implies

$$\exists l \quad \frac{\overline{X}_l}{f_l^{-1}(y(p^n))} > \frac{\overline{X}_k}{f_k^{-1}(y(p^n))}$$

and this implies that $\exists l \quad Z_l(p^n) > Z_k(p^n)$ □

Theorem 1 *A buffer equilibrium exists in the P2P exchange economy, given that users' preferences are continuous and strictly convex, monotone w r t service products as well as strongly monotone w r t to supply resources*

Proof Consider the relative buffer function $Z(p)$ We have noted in Observation 1 that $\overline{X(p)}$ and $y(p)$ are both homogeneous of degree zero, and this implies that $Z(p)$ is homogeneous of degree zero Thus, we can normalize prices in such a way that all prices sum up to 1 More precisely, denote by

$$\Delta = \left\{ p \in \mathbb{R}^L_+ \quad \sum_l p_l = 1 \right\}$$

We can restrict our search for an equilibrium to price vectors in Δ However, the function $Z(p)$ is only well-defined for price vectors in

$$\text{Interior } \Delta = \{p \in \Delta \quad p_l > 0 \text{ for all } l\}$$

To refer to price vectors in Δ that are not in the interior, we use

$$\text{Boundary } \Delta = \Delta \setminus \text{ Interior } \Delta$$

The proof proceeds in six steps In the first two steps, we define a correspondence $f( )$ from Δ to Δ, where we distinguish between price vectors in Interior Δ and in Boundary Δ In step 3, we show that the correspondence is convex-valued In step 4, we show that the correspondence is upper hemicontinuous In step 5, we use all of these results and apply Kakutani's fixed point theorem to conclude that a p^* with $p^* \in f(p^*)$ is guaranteed to exist Finally, in step 6 we show that any fixed point constitutes an equilibrium price vector To facilitate notation, we will use q to denote price vectors in the set $f(p) \subset \Delta$

Step 1 Construction of the correspondence $f( )$ for $p \in$ Interior Δ For the definition of this correspondence, we put the resources in an arbitrary but fixed order, and index them by $i, j \in \{1, 2, 3\}$ Now, $\forall p \in$ Interior Δ

$$f(p) = \begin{cases} q \in \Delta, & \text{if } Z(p) = 0 \\ q \in \Delta \quad q_i = 1 \text{ if } i = \arg\min\{p_j \quad p_j = \min\{p_1, p_2, p_3\}\}, & \text{if } Z(p) \neq 0 \end{cases}$$

In words, if $Z(p) = 0$, i e , when the buffer is the same for all resources, then the correspondence $f( )$ maps p to the set of all price vectors in Δ If $Z(p) \neq 0$, then the correspondence maps p to a price vector $q \in \Delta$ where one component of q equals 1 and the other two components are equal to 0 More specıfically, the correspondence sets that component $q_\imath = 1$ for whıch $\imath$ ıs the smallest ındex of the prıce components $p_\jmath$ that are mınımal among p_1, p_2 and p_3 Thus, when $Z(p) \neq 0$, then $f(\ )$ maps p to *exactly one* $q \in$ Boundary Δ Only ıf $Z(p) = 0$, then $f(p) = \Delta$

Step 2 Constructıon of the correspondence $f(\ )$ for $p \in$ Boundary Δ

$$\forall p \in \text{Boundary } \Delta \quad f(p) = \{q \in \Delta \quad q_\imath = 0 \text{ ıf } p_\imath > 0\}$$

Thıs correspondence maps p to all prıce vectors $q \in \Delta$ for whıch a component of q equals 0 when the correspondıng component of p ıs posıtıve Because $p \in$ Boundary Δ we know that for some $\imath$, $p_\imath = 0$, and thus $f(p) \neq \emptyset$ Furthermore, for at least one $\imath$, $p_\imath > 0$ and thus $q_\imath = 0$, whıch ımplıes that no poınt from Boundary Δ can be a fixed poınt

Step 3 The fixed-poınt correspondence ıs conver-valued Consıder first $p \in$ Interıor Δ If $Z(p) = 0$, then $f(p) = \Delta$, and because Δ ıs a sımplcx ıt ıs obvıously convex When $p \in$ Interıor Δ and $Z(p) \neq 0$, then $f(\ )$ maps p to exactly one poınt ın Δ, and thus $f(p)$ ıs trıvıally convex Now, ıf $p \in$ Boundary Δ, then $f(p)$ ıs a subset of Δ, namely the set of prıce vectors q where one or two dımensıons are equal to 0 These subsets of Δ are themselves sımplıces, and thus convex, and consequently $f(p)$ ıs convex

Step 4 The correspondence $f(\ )$ ıs upper hemıcontınous To show upper hemıcontınuıty we have to prove that for any sequence $p^n \rightarrow p$ and $q^n \rightarrow q$ wıth $q^n \in f(p^n)$ ıt holds that $q \in f(p)$ We dıstınguısh two cases $p \in$ Interıor Δ and $p \in$ Boundary Δ

Step 4a $p \in$ *Interior* Δ Consider first a sequence $p^n \to p$ with $Z(p) = 0$ Thus, $f(p) = \Delta$ and for any sequence $q^n \to q$, it is trivially true that $q \in f(p)$ Now consider a sequence $p^n \to p$ with $Z(p) \neq 0$ Because users' preferences are continuous (Assumption 1), we know that $\overline{X(p)}$ and $y(p)$ are continuous, which implies the continuity of $Z(\ )$, and thus $\lim_{n\to\infty} Z(p^n) = Z(p)$ Because $Z(p) \neq 0$, for n large enough it must be that $Z(p^n) \neq 0$ Thus, when considering the sequence $p^n \to p$, for n large enough, we only have to consider the second case of the definition of $f(\ )$ Let $i^* = \arg\min\{p_j \ \ p_j = \min\{p_1, p_2, p_3\}\}$ It holds that $\lim_{n\to\infty} \min\{p_1^n, p_2^n, p_3^n\} = \min\{p_1, p_2, p_3\}$ Thus, for n large enough, it must be that $\arg\min\{p_j^n \ \ p_j^n = \min\{p_1^n, p_2^n, p_3^n\}\} = i^*$ Consequently, for n large enough, if $q^n \in f(p^n)$, then $q_{i^*}^n = 1$ which implies that $q_{i^*} = 1$ Thus, if $q^n \to q$ and for all n $q^n \in f(p^n)$, then $q \in f(p)$

Step 4b $p \in$ *Boundary* Δ Consider $p^n \to p$ and $q^n \to q$ with $q^n \in f(p^n)$ for all n We show that for any $p_l > 0$, for n sufficiently large we have $q_l^n = 0$ and thus $q_l = 0$ which implies that $q \in f(p)$ If $p_l > 0$, then $p_l^n > 0$ for n sufficiently large If $p^n \in$ Boundary Δ, then $q_l^n = 0$ by the definition of the correspondence $f(p^n)$, and thus $q_l = 0$ If, however, $p^n \in$ Interior Δ, then Lemma 2 comes into play Because $p \in$ Boundary Δ, for at least one k we have $p_k = 0$ and thus $p_k^n \to 0$ According to Lemma 2, for n large enough

$$\exists l \quad Z_l(p^n) > Z_k(p^n)$$

i e , there exists a resource l which has a larger buffer than resource k Thus, $Z_k(p^n) \neq Z_l(p^n)$ and thus $Z(p^n) \neq 0$, which implies that we must only consider the second case of the definition of $f(p^n)$ for $p^n \in$ Interior Δ If $q^n \in f(p^n)$, then for n large enough

$q_k^n = 1$ for a resource k for whıch $p_k^n \to 0$ Because $p \in$ Boundary Δ, at least one and at most two components of p^n go towards 0 However, because $q^n \to q$, for n large enough, $q_k^n = 1$ for the same resource k, and thus $q_k = 1$, whıch ımplıes that $q_l = q_m = 0$ Thus, for any $p_l > 0$, $q_l = 0$, whıch ımplıes that $q \in f(p)$

Step 5 A fixed poınt exısts The set Δ ıs a non-empty, convex and compact set and we have shown that $f(\,)$ ıs a correspondence from Δ to Δ that ıs convex-valued and upper hemıcontınuous Thus, we can apply Kakutanı's fixed-poınt theorem whıch says that any convex-valued and upper hemıcontınuous correspondence from a non-empty, compact and convex set ınto ıtself has a fixed poınt We conclude that there exısts a $p^* \in \Delta$ wıth $p^* \in f(p^*)$

Step 6 A fixed poınt of $f(\,)$ ıs an equılıbrıum Assume that p^* ıs a fixed poınt, ı e , $p^* \in f(p^*)$ As we have poınted out ın step 2, no prıce vector from Boundary Δ can be a fixed poınt Thus, ıt must be that $p^* \in$ Interıor Δ In step 1, we already saw that when $Z(p^*) \neq 0$, then $f(p^*) \subset$ Boundary Δ, whıch ıs ıncompatıble wıth $p^* \in$ Interıor Δ and $p^* \in f(p^*)$ Thus, for p^* to be a fixed poınt, ıt must hold that $Z(p^*) = 0$, and thus any fixed poınt p^* ıs an equılıbrıum prıce vector

To summarıze, we have shown that a fixed poınt always exısts and that any fixed poınt ıs an equılıbrıum prıce vector Thus, gıven the assumptıons of the theorem, a buffer equılıbrıum ıs guaranteed to exıst □

2.5.3 Equilibrium Existence with Price-insensitive or Adversarial Users

So far, we have shown the existence of the buffer equilibrium when all users' preferences satisfy continuity, strict convexity, monotonicity and strong monotonicity w.r.t. supply resources, and update their settings accordingly upon price changes. In practice, however, some users might violate these assumptions, for example, because they do not notice price changes, or because they do not care enough to update their settings immediately. In more extreme cases, some users might purposefully harm the system and try to bring it out of equilibrium by updating their settings in the opposite way than what our assumptions would suggest. We call such users *adversarial* users. For example, an adversarial user could maximize his supply of those resources that currently have a very low price, and minimize his supply of those resources that currently have a very high price. Even though such behavior would certainly hurt the attacking user himself and thus could be called *irrational*, adversarial users do exist in practice, and robustness against adversarial attacks is a common concern.

In this section, we prove that a buffer equilibrium exists, even if a certain percentage of the user population is adversarial. For the analysis, we distinguish between *rational* users whose preferences satisfy our assumptions as before, and who update their settings accordingly upon price changes, and *adversarial* users, whose preferences must not satisfy our assumptions. To derive the maximum percentage of adversarial users that we can tolerate, the following analysis assumes that adversarial users update their settings in such a way as to maximally hurt the system, to bring it out of equilibrium.

We let R denote the set of rational users, and A denote the set of adversarial users We let Y^R and X^R denote the demand and supply vector of the rational users, and Y^A and X^A denote the demand and supply vector of the adversarial users Thus, $Y = Y^R + Y^A$ and $X = X^R + X^A$ As before, we let $\gamma > 1$ denote the system's slack constraint We assume that the maximum demand of the rational users is at least C times larger than the maximum demand of the adversarial users, i e , $Y^R \geq C\ Y^A$, and we derive a minimum bound for C to guarantee the existence of a buffer equilibrium

As a first step, we show that under certain conditions, when the price of a resource k goes towards zero, there exists a resource $l \neq k$ with a strictly larger resource buffer than k

Lemma 3 *Given slack factor γ and given that $Y^R \geq C \quad Y^A$, if rational users' preferences are continuous and strictly convex, monotone w r t service products as well as strongly monotone w r t supply resources, and if $C > (\gamma^2 + \gamma)$, then for $p^n \to p$ with $p \neq 0$ and $p_k = 0$, for n sufficiently large*

$$\exists l \quad \frac{\overline{X_k(p^n)}}{f_k^{-1}(y(p^n))} < \frac{\overline{X_l(p^n)}}{f_l^{-1}(y(p^n))}$$

Proof We have shown in the proof for Lemma 2, that for $p^n \to p$ with $p \neq 0$ and $p_k = 0$, for every rational user i, for n large enough, at least one of the slack constraints will bind, i e

$$\forall i \quad \exists l \quad \frac{\overline{X^R_{ik}(p^n)}}{f_k^{-1}(y(p^n))} = \frac{1}{\gamma}\frac{\overline{X^R_{il}(p^n)}}{f_l^{-1}(y(p^n))}$$

For the remainder of the proof, we will always consider the supply and demand functions for $p^n \to p$, however, we will write $\overline{X}$ and y instead of $\overline{X(p^n)}$ and $y(p^n)$ to simplify notation. It is possible, that for each rational user, a different slack constraint binds. Let L and M denote the sets of rational users for whom the slack constraints bind for resources l and m, respectively, i.e., $R = L \cup M$. We assume that L and M are disjunct, if for some user, both slack constraints for l and m bind, we can place that user randomly into either L or M. We let $\overline{X_l^L} = \sum_{i \in L} \overline{X_{il}^R}$ and $\overline{X_l^M} = \sum_{i \in M} \overline{X_{il}^R}$. Then

$$\frac{\overline{X_l^L}}{f_l^{-1}(y)} \geq \frac{\overline{X_m^L}}{f_m^{-1}(y)} \quad \text{and} \quad \frac{\overline{X_m^M}}{f_m^{-1}(y)} \geq \frac{\overline{X_l^M}}{f_l^{-1}(y)}$$

It is easy to see that at least for one of the resources l or m, the joint supply of that resource from the corresponding set of users L or M must be at least half of the total supply of that resource from the rational users. With out loss of generality, let l be such a resource. Thus

$$\frac{\overline{X_l^L}}{f_l^{-1}(y)} \geq \frac{1}{2} \frac{\overline{X_l^R}}{f_l^{-1}(y)}$$

Remember that for all users $\in L$, the slack constraint for l binds. For all other rational users, we only know that they supply least of resource k. Thus

$$\gamma \frac{\overline{X_k^L}}{f_k^{-1}(y)} = \frac{\overline{X_l^L}}{f_l^{-1}(y)} \quad \text{and} \quad \frac{\overline{X_k^M}}{f_k^{-1}(y)} \leq \frac{\overline{X_l^M}}{f_l^{-1}(y)}$$

By adding both sides together we get

$$\gamma \frac{\overline{X_k^L}}{f_k^{-1}(y)} + \frac{\overline{X_k^M}}{f_k^{-1}(y)} \leq \frac{\overline{X_l^R}}{f_l^{-1}(y)}$$

Because $\overline{X_l^L} + \overline{X_l^M} = \overline{X_l^R}$, this is equivalent to

$$(\gamma - 1) \frac{\overline{X_k^L}}{f_k^{-1}(y)} + \frac{\overline{X_k^R}}{f_k^{-1}(y)} \leq \frac{\overline{X_l^R}}{f_l^{-1}(y)}$$

Because $\frac{\overline{X_l^L}}{f_l^{-1}(y)} \geq \frac{1}{2}\ \frac{\overline{X_l^R}}{f_l^{-1}(y)}$, this implies

$$\begin{aligned}\left(\frac{\gamma-1}{2}\right)\ \frac{\overline{X_k^R}}{f_k^{-1}(y)} + \frac{\overline{X_k^R}}{f_k^{-1}(y)} &\leq \frac{\overline{X_l^R}}{f_l^{-1}(y)} \\ \Rightarrow \left(\frac{\gamma+1}{2}\right)\ \frac{\overline{X_k^R}}{f_k^{-1}(y)} &\leq \frac{\overline{X_l^R}}{f_l^{-1}(y)} \\ \Rightarrow \frac{\overline{X_k^R}}{f_k^{-1}(y)} &\leq \left(\frac{2}{\gamma+1}\right)\ \frac{\overline{X_l^R}}{f_l^{-1}(y)}\end{aligned}$$

So far, we have only argued about the rational users, and derived how much smaller the buffer for resource k for these users must be relative to the maximum buffer for resource l or m Now we turn our attention to the adversarial users as well Because $Y^R \geq C\ \ Y^A$ we know that $\overline{X^R}\ \ p \geq C\ \ \overline{X^A}\ \ p$ For large enough n, we know that p_k^n is close enough to 0 such that all income must come from supply resources l and m Thus

$$\overline{X_l^R}\ \ p_l + \overline{X_m^R}\ \ p_m \geq C\ \ \left(\overline{X_l^A}\ \ p_l + \overline{X_m^A}\ \ p_m\right) \tag{2 20}$$

Because l was assumed to be the resource with the largest buffer for the rational users, we know that

$$\overline{X_m^R} \leq \overline{X_l^R}\ \ \frac{f_m^{-1}(y)}{f_l^{-1}(y)} \tag{2 21}$$

For the adversarial users, there is no restriction between the buffers for l and m, except the standard slack constraint, i e

$$\overline{X_m^A} \geq \frac{1}{\gamma}\ \ \overline{X_l^A}\ \ \frac{f_m^{-1}(y)}{f_l^{-1}(y)} \tag{2 22}$$

If we combıne Equatıons 2 20, 2 21 and 2 22, then we get

$$\overline{X_l^R}\ p_l + \overline{X_l^R}\ \frac{f_m^{-1}(y)}{f_l^{-1}(y)}\ p_m \geq C\ \left(\overline{X_l^A}\ p_l + \frac{1}{\gamma}\ \overline{X_l^A}\ \frac{f_m^{-1}(y)}{f_l^{-1}(y)}\ p_m\right) \tag{2 23}$$

$$\Rightarrow \overline{X_l^R}\ p_l + \overline{X_l^R}\ \frac{f_m^{-1}(y)}{f_l^{-1}(y)}\ p_m \geq C\ \overline{X_l^A}\ p_l + \frac{C}{\gamma}\ \overline{X_l^A}\ \frac{f_m^{-1}(y)}{f_l^{-1}(y)}\ p_m \tag{2 24}$$

For the last ınequalıty to be true, a necessary condıtıon ıs

$$\overline{X_l^R} \geq \min\{CX_l^A, \frac{C}{\gamma}X_l^A\} \tag{2 25}$$

$$\Rightarrow \overline{X_l^R} \geq \frac{C}{\gamma}\overline{X_l^A} \tag{2 26}$$

$$\Rightarrow \overline{X_l^A} \leq \frac{\gamma}{C}\ \overline{X_l^R} \tag{2 27}$$

We have derıved above that for ratıonal users, we have

$$\frac{\overline{X_k^R}}{f_k^{-1}(y)} \leq \left(\frac{2}{\gamma+1}\right)\ \frac{\overline{X_l^R}}{f_l^{-1}(y)} \tag{2 28}$$

For the adversarıal users, we have

$$\frac{\overline{X_k^A}}{f_k^{-1}(y)} \leq \gamma\ \frac{\overline{X_l^A}}{f_l^{-1}(y)} \tag{2 29}$$

If we take these two ınequalıty together we get

$$\frac{\overline{X_k}}{f_k^{-1}(y)} \leq \frac{\frac{2}{\gamma+1}\ \overline{X_l^R} + \gamma\ \overline{X_l^A}}{f_l^{-1}(y)} \tag{2 30}$$

Thus, to get

$$\frac{\overline{X_k}}{f_k^{-1}(y)} \leq \frac{\overline{X_l}}{f_l^{-1}(y)} \tag{2 31}$$

we need that

$$\frac{2}{\gamma+1}\ \overline{X_l^R} + \gamma\ \overline{X_l^A} \leq \overline{X_l} \tag{2 32}$$

By definition, we have that $\overline{X_l^R} + \overline{X_l^A} = \overline{X_l}$ Because $\gamma > 1$, we know that $\frac{2}{\gamma+1} \; \overline{X_l^R} < \overline{X_l^R}$ and $\gamma \; \overline{X_l^A} > \overline{X_l^A}$ Thus, the amount by which $\frac{2}{\gamma+1} \; \overline{X_l^R}$ is smaller than $\overline{X_l^R}$ is exactly the amount by which $\gamma \; \overline{X_l^A}$ can be larger than $\overline{X_l^A}$, for Inequality 2 32 to hold Thus, we need

$$(\gamma - 1) \; \overline{X_l^A} \leq (1 - \frac{2}{\gamma + 1})\overline{X_l^R} \qquad (2\ 33)$$

If we now use Equation 2 27, i e , $\overline{X_l^A} \leq \frac{\gamma}{C} \; \overline{X_l^R}$, it follows that the next inequality implies the previous one

$$(\gamma - 1) \; \frac{\gamma}{C} \; \overline{X_l^R} \leq (1 - \frac{2}{\gamma + 1})\overline{X_l^R} \qquad (2\ 34)$$

$$\Leftrightarrow \frac{\gamma^2 - \gamma}{C} \leq \frac{\gamma - 1}{\gamma + 1} \qquad (2\ 35)$$

Because $\gamma > 1$ and $C > 1$, we can derive the following

$$(\gamma + 1) \; (\gamma^2 - \gamma) \leq C \; (\gamma - 1) \qquad (2\ 36)$$

$$\Leftrightarrow (\gamma - 1) \; (\gamma^2 + \gamma) \leq C \; (\gamma - 1) \qquad (2\ 37)$$

$$\Leftrightarrow (\gamma^2 + \gamma) \leq C \qquad (2\ 38)$$

This completes the proof of the lemma □

Equipped with Lemma 3, it is straightforward to prove the more general Theorem about equilibrium existence with adversarial users

Theorem 2 *Given slack factor γ and given that $Y^R \geq C \; Y^A$, then a buffer equilibrium exists in the P2P exchange economy if $C > (\gamma^2 + \gamma)$ and the rational users' preferences are continuous and strictly convex, monotone w r t service products as well as strongly monotone w r t to supply resources*

Proof The theorem follows from the same proof as Theorem 1 The only necessary change is that in step 4b of the proof, instead of using Lemma 2 (which is only applicable when all users are rational), we use the more general Lemma 3 □

What Theorem 2 shows is that the more freedom we give the users in setting their supply (i e , the larger the slack factor), the less robust is the system against adversarial attacks This result is actually very relevant and useful for the designer of the P2P backup market If there is reason to believe that a non-negligible fraction of the population will be adversarial or that many users will not update their prices in a rational way, then Theorem 2 tells the market designer exactly what to do For example, if the market designer believes that at most 10% of the users will be adversarial, then the formula from the theorem tells us that as long as we give the users a slack factor of 2 5 or less, a buffer equilibrium is guaranteed to exist In that respect, the theoretical equilibrium analysis actually has a very direct practical impact on the market design

2.5.4 Equilibrium Uniqueness

Without any further restrictions on users' preferences, we cannot say anything about the uniqueness of the buffer equilibrium, because the substitution effect and the wealth effect could either go in the same or in opposite directions [12] The standard equilibrium uniqueness proof for Walrasian equilibria resolves this by assuming that

[12] In an exchange economy, a price change always has two effects first, it changes the relative prices between the goods, causing the substitution effect Second, it can also change a user's wealth, because his supply might now be more or less valuable, which is called the wealth effect Without further assumptions, nothing can be said about the net effect of a price change (cf Sonnenschein-Mantel-Debreu Theorem, [65], pp 598-606)

the aggregate excess demand function has the gross substitutes property for *all* commodities [3], which means that a price increase for one commodity causes an increase in the aggregate excess demand for all other commodities However, that assumption is too strong for our domain for two reasons First, and most importantly, for the demanded services, the gross substitutes property is violated in a P2P backup system For example, if the price for storage increases, it is not reasonable to assume that users will now start deleting their backed up files and consume more backup or retrieval operations instead The reason is simple every file you back up is then being stored, and you can only retrieve files you have previously backed up Thus, there are in fact strong complementarities between the demanded services in our domain, and to reflect this, we make the following assumption

Assumption 5 *(Services are Perfect Complements) We assume that the aggregate demand function $Y(\,)$ has the perfect complements property, i e*

$$\forall p, p' \in \mathbb{R}^3_{>0} \quad \exists \mu \in \mathbb{R} \ \ s\,t \ \ Y(p) = \mu \ \ Y(p')$$

A consequence of the perfect complements property is that price changes affect all dimensions of the aggregate demand vector equally For an individual user, the Leontief utility function would induce the perfect complements property such that resources are consumed in fixed ratios However, it bears emphasis that we assume perfect complements only for *aggregate* demand, rather than for individual demand, which is a much weaker assumption, and more reasonable due to the law of large numbers

In contrast to service products, it seems reasonable to assume that supplied resources are substitutes in the sense that a user is happy to shift his supply from one resource to another as prices change. Yet, the strong assumption that supplied resources are gross substitutes might also not hold in our domain. Because services have the perfect complements property, and because services and supplied resources are coupled via the flow constraint $\overline{X}_i \cdot p = f^{-1}(Y) \cdot p$, price changes can also have non-substitution effects on the supply of resources. For example, when the price for a resource is decreased, it is not a priori clear that the supply for that resource goes down. It might be, that due to this price decrease, the system just became much more attractive for many users, so that they significantly increase their demand and thus also their supply (of all resources). Thus, we do not want to make assumptions regarding the specific directions of change in the supply and demand functions. We only make an assumption regarding how price changes affect the relative ratios of supplied resources to each other.

Assumption 6 *(Relative Supply Resources are Gross Substitutes) We assume that the aggregate supply function $\overline{X}(p)$ has the relative gross substitutes property, i.e., whenever p' and p are such that, for some k, $p'_k > p_k$ and $p'_l = p_l$ for $l \neq k$, we have $\frac{\overline{X}_k(p')}{\overline{X}_l(p')} > \frac{\overline{X}_k(p)}{\overline{X}_l(p)}$.*

Note that both assumptions are relatively weak. Upon a price decrease for good k, the aggregate supply for k can go up or down, and the demand for all services can also go up or down. All we assume is that when the price for good k is decreased, the relative supply of good k to the other goods decreases, and the demand for services moves up or down proportionally. With these two assumptions, we can now prove that the buffer equilibrium is unique.

Theorem 3 *The buffer equilibrium is unique, given that the aggregate demand func-*

tion satisfies the perfect complements property (Assumption 5), and that the aggregate supply function satisfies the relative gross substitute property (Assumption 6)

Proof Because we make different assumptions regarding the supply and demand sides of our economy, we first separate the supply and demand aspects by introducing an alternative description of the buffer equilibrium

$$\overline{X} = f^{-1}(Y) \tag{2 39}$$

$$\Leftrightarrow \left(\overline{X}_{\overline{S}}, \overline{X}_{\overline{U}}, \overline{X}_{\overline{D}}\right) = \left(f_{\overline{S}}^{-1}(Y), f_{\overline{U}}^{-1}(Y), f_{\overline{D}}^{-1}(Y)\right) \tag{2 40}$$

$$\Leftrightarrow \left(1, \frac{\overline{X}_{\overline{U}}}{\overline{X}_{\overline{S}}}, \frac{\overline{X}_{\overline{D}}}{\overline{X}_{\overline{S}}}\right) = \left(1, \frac{f_{\overline{U}}^{-1}(Y)}{f_{\overline{S}}^{-1}(Y)}, \frac{f_{\overline{D}}^{-1}(Y)}{f_{\overline{S}}^{-1}(Y)}\right) \tag{2 41}$$

$$\Leftrightarrow \left(\frac{\overline{X}_{\overline{U}}}{\overline{X}_{\overline{S}}}, \frac{\overline{X}_{\overline{D}}}{\overline{X}_{\overline{S}}}\right) - \left(\frac{f_{\overline{U}}^{-1}(Y)}{f_{\overline{S}}^{-1}(Y)}, \frac{f_{\overline{D}}^{-1}(Y)}{f_{\overline{S}}^{-1}(Y)}\right) = 0 \tag{2 42}$$

We define a new vector-valued function $g(p) = \left(g_{\overline{U}}(p), g_{\overline{D}}(p)\right)$

$$g_{\overline{U}}(p) = \left(\frac{\overline{X}_{\overline{U}}}{\overline{X}_{\overline{S}}} - \frac{f_{\overline{U}}^{-1}(Y)}{f_{\overline{S}}^{-1}(Y)}\right) \quad \text{and} \quad g_{\overline{D}}(p) = \left(\frac{\overline{X}_{\overline{D}}}{\overline{X}_{\overline{S}}} - \frac{f_{\overline{D}}^{-1}(Y)}{f_{\overline{S}}^{-1}(Y)}\right),$$

which naturally leads to a new equilibrium definition that is equivalent to Definitions 3 and 4

Definition 5 *(Buffer Equilibrium [Version 3]) A buffer equilibrium is a price vector p and $g(p)$ such that*

$$g(p) = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

We have simplified the problem of finding equilibrium prices to finding the root of the function $g(p)$ Because $\overline{X}(p)$ and $Y(p)$ are homogeneous of degree zero, $g(p)$ is also homogeneous of degree zero, which implies that collinear price vectors are equivalent, i e , $\forall \lambda > 0 \quad g(p) = g(\lambda \quad p)$ Thus, showing uniqueness of the buffer equilibrium is now equivalent to showing that $g(p) = 0$ has at most one normalized solution Now, let's assume that $g(p) = 0$, i e , p is an equilibrium price vector We show that for any p', $g(p') \neq 0$ unless p and p' are collinear Because of Assumption 5 (the aggregate demand function has the perfect complements property), a price change affects all dimensions of the demand function equally, i e , $\exists \mu \in \mathbb{R} \ \ Y(p) = \mu \ \ Y(p')$ Because the production function is bijective and exhibits constant returns to scale, this implies that $f^{-1}(Y(p)) = \mu \ \ f^{-1}(Y(p'))$ Thus, $\forall p, p' \in \mathbb{R}^3_{>0} \ \ \frac{f_{\overline{U}}^{-1}(Y(p))}{f_{\overline{S}}^{-1}(Y(p))} = \frac{f_{\overline{U}}^{-1}(Y(p'))}{f_{\overline{S}}^{-1}(Y(p'))}$, i e , changes in the demand function $Y(\ )$ due to price changes do not affect $g(\ )$ Consequently, we only have to consider changes in the supply function $\overline{X}(\ )$ Now consider a price vector p' that is not collinear with p Because of the homogeneity of degree zero, we can assume that $p' \geq p$ and $p_l = p'_l$ for some l We now alter the price vector p' to obtain a price vector that is collinear to p, and argue about how $g(\ )$ changes in the process We distinguish between three cases

Case 1 $l = \overline{S}$, i e , $p'_{\overline{S}} = p_{\overline{S}}$ First, we generate a price vector p'' that is collinear to p, by linearly increasing all components of p until the next two price components are equal, i e , $p''_k = p'_k$ for $k \neq \overline{S}$ We assume that $k = \overline{D}$ (the case where $k = \overline{U}$ is completely symmetric) such that

$$p'_{\overline{U}} \geq p''_{\overline{U}} \tag{2 43}$$

$$p'_{\overline{D}} = p''_{\overline{D}} \tag{2 44}$$

$$p'_{\overline{S}} \leq p''_{\overline{S}} \tag{2 45}$$

with at least one of the inequalities being strict Now we alter p' to obtain p'' in two steps In the first step, we decrease (or keep unaltered) $p'_{\overline{U}}$ until it equals $p''_{\overline{U}}$ In the second step, we increase (or keep unaltered) $p'_{\overline{S}}$ until it equals $p''_{\overline{S}}$ Because p' and p''

were not collinear, we have changed the price vector in at least one step, and because of Assumption 6, the relative ratio between $\overline{X}_U$ and $\overline{X}_{\overline{S}}$ has decreased in at least one step and has never increased, such that

$$\frac{\overline{X}_{\overline{U}}(p')}{\overline{X}_{\overline{S}}(p')} > \frac{\overline{X}_{\overline{U}}(p'')}{\overline{X}_{\overline{S}}(p'')} = \frac{\overline{X}_{\overline{U}}(p)}{\overline{X}_{\overline{S}}(p)}$$

Thus, the first term in $g_{\overline{U}}(\ )$ has changed, and the second term stayed constant, and $g(p') \neq g(p) = 0$

Case 2 $l = \overline{U}$, i e , $p'_{\overline{U}} = p_{\overline{U}}$ First, we generate a price vector p'' that is collinear to p, by linearly increasing all components of p until $p''_k = p'_k$ for $k \neq \overline{U}$ Now we differentiate between two cases

Case 2a $k = \overline{D}$ such that

$$p'_{\overline{S}} \geq p''_{\overline{S}} \tag{2 46}$$

$$p'_{\overline{D}} = p''_{\overline{D}} \tag{2 47}$$

$$p'_{\overline{U}} \leq p''_{\overline{U}} \tag{2 48}$$

with at least one of the inequalities being strict The remainder of the proof for this case is analogous to the one fore case 1

Case 2b $k = \overline{S}$ such that

$$p'_{\overline{D}} \geq p''_{\overline{D}} \tag{2 49}$$

$$p'_{\overline{S}} = p''_{\overline{S}} \tag{2 50}$$

$$p'_{\overline{U}} \leq p''_{\overline{U}} \tag{2 51}$$

with at least one of the inequalities being strict Analogously to the proof for case

1, we can show that

$$\frac{\overline{X}_{\overline{U}}(p')}{\overline{X}_{\overline{D}}(p')} < \frac{\overline{X}_{\overline{U}}(p'')}{\overline{X}_{\overline{D}}(p'')} = \frac{\overline{X}_{\overline{U}}(p)}{\overline{X}_{\overline{D}}(p)} \tag{2.52}$$

For the rest of the proof for this case, we construct a contradiction. Assume that p' is also an equilibrium price vector such that $g(p') = 0$. Because the second term in $g_{\overline{U}}$ and $g_{\overline{D}}$ respectively does not change upon price changes, this implies that

$$\frac{\overline{X}_{\overline{U}}(p')}{\overline{X}_{\overline{S}}(p')} = \frac{\overline{X}_{\overline{U}}(p)}{\overline{X}_{\overline{S}}(p)} \tag{2.53}$$

$$\text{and} \quad \frac{\overline{X}_{\overline{D}}(p')}{\overline{X}_{\overline{S}}(p')} = \frac{\overline{X}_{\overline{D}}(p)}{\overline{X}_{\overline{S}}(p)} \tag{2.54}$$

From Equation (2.53) it follows that $\overline{X}_{\overline{U}}(p') = \frac{\overline{X}_{\overline{U}}(p)}{\overline{X}_{\overline{S}}(p)} \cdot \overline{X}_{\overline{S}}(p')$ and from (2.54) it follows that $\overline{X}_{\overline{D}}(p') = \frac{\overline{X}_{\overline{D}}(p)}{\overline{X}_{\overline{S}}(p)} \cdot \overline{X}_{\overline{S}}(p')$. If we put these two results together we get

$$\frac{\overline{X}_{\overline{U}}(p')}{\overline{X}_{\overline{D}}(p')} = \frac{\overline{X}_{\overline{U}}(p) \cdot \overline{X}_{\overline{S}}(p') \cdot \overline{X}_{\overline{S}}(p)}{\overline{X}_{\overline{S}}(p) \cdot \overline{X}_{\overline{D}}(p) \cdot \overline{X}_{\overline{S}}(p')} = \frac{\overline{X}_{\overline{U}}(p)}{\overline{X}_{\overline{D}}(p)}$$

and this contradicts Equation (2.52). Thus, $g(p') \neq 0$.

Case 3 $l = \overline{D}$, i.e., $p'_{\overline{D}} = p_{\overline{D}}$. The proof for this case is analogous to the proof for case 2.

In summary, in all three cases we established that $g(p') \neq g(p) = 0$ which shows that p' is not an equilibrium price vector and concludes the equilibrium uniqueness proof. □

2.5.5 (Un-)Controllability of the Supply-side Buffer

So far we have shown under what conditions the buffer equilibrium exists and when it is unique. In practice, however, the system will be out of equilibrium most of the time, because users do not continuously adjust their settings, and thus price changes will only affect supply and demand after a delay This is why in Section 2 5 1, we have motivated the buffer equilibrium as a desirable target the buffer between current demand and maximum supply of resources gives the system a certain safety for when it is out of equilibrium To make sure we can always satisfy new incoming demand, we might like to have at least 25% more supply than current demand, i e , $\overline{X} \geq 1\,25 \;\; f^{-1}(y)$ Unfortunately, the uniqueness of the buffer equilibrium (Theorem 3) has an immediate consequence regarding the limited controllability of the buffer equilibrium

Corollary 1 *(Limited Controllability of the Market) Given Assumptions 5 and 6, the market operator cannot influence the size of the buffer in the buffer equilibrium by adjusting market prices*

It turns out that the limited controllability of the buffer equilibrium remains, even without the assumptions that service are perfect complements and that relative supply resources are gross substitutes, thereby strengthening the result from Corollary 1

Proposition 3 *If each individual user i has a limited planning horizon in that he chooses not to give himself more than a demand-side buffer of λ_i, then there exists a $\Lambda \in \mathbb{R}_{>1}$ such that the market operator cannot achieve a buffer equilibrium with buffer size Λ by adjusting market prices*

Proof For the proof we construct a simple counterexample We choose a Λ such that

$\forall i \;\; \Lambda > \lambda_i$ And we let $\lambda_i^* = \max_i \lambda_i$ Now

$$\forall i \;\; Y_i = \lambda_i \;\; y_i \tag{2 55}$$

$$\Rightarrow Y = \sum_i \lambda_i \;\; y_i \tag{2 56}$$

$$\Rightarrow Y \leq \sum_i \lambda_i^* \;\; y_i \tag{2 57}$$

$$\Rightarrow Y \leq \lambda_i^* \sum_i \;\; y_i \tag{2 58}$$

$$\Rightarrow Y \leq \lambda_i^* y \tag{2 59}$$

$$\Rightarrow f^{-1}(y) \leq \lambda_i^* f^{-1}(y) \tag{2 60}$$

$$\Rightarrow \overline{X} \leq \lambda_i^* f^{-1}(y) \tag{2 61}$$

Thus, the buffer between supply and demand would be less or equal to λ_i^* which by assumption was strictly less than the buffer Λ that the market operator desired □

Given the limited controllability of the buffer, it is natural to ask what buffer size to expect in equilibrium It turns out that, in equilibrium, the supply-side buffer is uniquely determined via the demand-side buffer

Proposition 4 *In the buffer equilibrium, the size of the supply-side buffer equals the size of the demand-side buffer*

Proof

$$\overline{X} = f^{-1}(Y) \tag{2 62}$$

$$\Leftrightarrow \overline{X} = f^{-1}(\lambda \;\; y) \tag{2 63}$$

$$\Leftrightarrow \overline{X} = \lambda \;\; f^{-1}(y) \tag{2 64}$$

Equation (2 63) follows because of Assumption 3 (linear prediction for aggregate demand) Equation (2 64) follows from System Properties 3 and 4 (production functions are bijective and exhibit CRTS) □

In words, the size of the buffer depends on how forward-looking the users are If on average the users give themselves a 25% buffer on the demand side (e g , a user has currently backed up 20GB and sets the sliders in such a position that his maximum online backup space is 25GB), then the system would also have a 25% buffer on the supply side, i e , $\overline{X} = 1\ 25 \quad f^{-1}(y)$

Even though the market operator cannot influence the size of the overall supply-side buffer by adjusting market prices, Proposition 4 provides us with a different, yet very natural way to achieve any desired buffer The market operator simply needs to insist that every user gives himself a certain minimum demand-side buffer One way to achieve this is to build this requirement into the user interface, i e , given user i's current demand y_i there would be a minimum demand $Y_i = \lambda_i \quad y_i$ below which the user could not go

Proposition 5 *If the market operator can enforce any demand-side buffer for individual users, then he can achieve any desired supply-side buffer size $\Lambda > 1$ in the buffer equilibrium*

Proof We let the market operator set all individual user's minimum required demand-side buffers to $\lambda_i = \Lambda$ Then we know from Proposition 4 that the resulting aggregate supply-side buffer will also be at least Λ □

Note that enforcing a demand-side buffer of Λ for every individual user *can* result in efficiency losses A user who, without this restriction, would have chosen a smaller demand-side buffer, now loses some utility For example, he might now choose a smaller Y_i to avoid having to give up as many resources X_i Thus, in practice, the desired supply-side buffer Λ would have to be carefully chosen, trading-off a larger supply-side buffer on the one hand, with some efficiency losses for individual users on the other hand

2.6 The Price Update Algorithm

In this section we propose and analyze a price update algorithm that is invoked regularly on the server (e g , once a day), with the goal to move prices towards the buffer equilibrium over time Our algorithm is oriented at the tâtonnement process as defined by Walras [105] However, Walras' algorithm only allowed trades at equilibrium prices In our system, however, we must allow trades at all times, even out of equilibrium

2.6.1 Algorithm Design

Because users' preferences are homogeneous of degree zero, collinear price vectors are equivalent Thus, instead of searching for the equilibrium price vector in $\mathbb{R}^3$, we can simplify the task by looking at projective space $\mathbb{RP}^2$

$$\mathbb{RP}^2 = \left\{(p_{\overline{S}}, p_{\overline{U}}, p_{\overline{D}}) \in \mathbb{R}^3 \setminus \{0\} \quad (p_{\overline{S}}, p_{\overline{U}}, p_{\overline{D}}) \sim \lambda(p_{\overline{S}}, p_{\overline{U}}, p_{\overline{D}}) \quad \forall \lambda \in \mathbb{R}_+\right\}$$

Thus, we can fix the price of an arbitrary good (the numeraire) and normalize the price vector accordingly Here, we normalize the price of storage space to 1

$$p = (p_{\overline{S}}, p_{\overline{U}}, p_{\overline{D}}) \sim (1, \frac{p_{\overline{U}}}{p_{\overline{S}}}, \frac{p_{\overline{D}}}{p_{\overline{S}}})$$

In Section 2 5 4, we have reduced the problem of finding the buffer equilibrium to finding the root of the function $g(p) = \big(g_{\overline{U}}(p), g_{\overline{D}}(p)\big)$ where

$$g_{\overline{U}}(p) = \Big(\frac{\overline{X}_{\overline{U}}}{\overline{X}_{\overline{S}}} - \frac{f_{\overline{U}}^{-1}(Y)}{f_{\overline{S}}^{-1}(Y)}\Big) \quad \text{and} \quad g_{\overline{D}}(p) = \Big(\frac{\overline{X}_{\overline{D}}}{\overline{X}_{\overline{S}}} - \frac{f_{\overline{D}}^{-1}(Y)}{f_{\overline{S}}^{-1}(Y)}\Big)$$

This formulation of the buffer equilibrium is also useful for the price update algorithm, because finding the root of a function is a well-understood mathematical problem *Newton's method* is probably the best-known root-finding algorithm and converges quickly in practice However, it requires the evaluation of the function's derivative at each step Unfortunately, we do not know the function $g(\ )$ and thus cannot compute its derivative Instead, we only get to know individual points in each iteration and can use these points to estimate the derivative This is exactly what the *secant method* does for a one-dimensional function

The problem is that $g(p)$ is 2-dimensional, and thus the secant method is not directly applicable The appropriate multi-dimensional generalization is *Broyden's method* [10], a quasi-Newton method Unfortunately, that method requires knowledge of the Jacobian, which we do not know and also cannot even measure approximately However, we show that one can use an approximation to the diagonal sub-matrix of the Jacobian instead of the full Jacobian matrix The diagonal sub-matrix of the Jacobian can be approximated by studying changes in the function $g(p)$ This leads to the following quasi-Newton method for multiple dimensions

Definition 6 *(The Price Update Algorithm)*

$$p_l^{t+1} = \begin{cases} 1 & \textit{for } l = \overline{S} \\ p_l^t - \frac{p_l^t - p_l^{t-1}}{g_l(p^t) - g_l(p^{t-1})} \; g_l(p^t) & \textit{for } l = \overline{U}, \overline{D} \end{cases}$$

For the implementation of the price update algorithm in our system we took care of a few special cases (e g , exactly reaching the equilibrium such that terms cancel out), but we omit the details here

2.6.2 Theoretical Convergence Analysis

We begin with the analysis of the convergence of the following iteration rule

$$x^{(k+1)} = x^{(k)} - D(x^{(k)})^{-1} F(x^{(k)}) \tag{2 65}$$

where F is a function $F \; \mathbb{R}^n \rightarrow \mathbb{R}^n$ and D is the diagonal sub-matrix of the Jacobian J of F We define the matrix L by the rule $J(x) = D(x) + L(x)$, i e , L comprises of the off-diagonal partial derivatives in the Jacobian For this iteration rule, the following theorem holds

Theorem 4 *Let F be a continuously differentiable function Suppose that in the iteration rule given by equation (2 65), $x^{(0)}$ is chosen close enough to a root x^* of F, $J(x^*)$ is non-singular, J and D are Lipschitz continuous, and $L(x^*) = 0$ Then the* *successive iterations $x^{(k)}$ produced by the iteration rule converge to x^*, and the rate of convergence is at least Q-linear*[13]

Before proving the main result, we first discuss some general conditions under which a multi-dimensional Newton iteration converges even if a diagonal approximation is used for the Jacobian We essentially follow Kantorovich's proof of the local convergence of Newton's method (Kantorovich's theorem [52] and [53] Chapter XVIII)

Definition 7 *Suppose $F \quad \mathbb{R}^n \rightarrow \mathbb{R}^m$ Writing the vector valued function $F(x_1, x_2, \quad , x_n)$ as*

$$(f_1(x_1, x_2, \quad , x_n), \quad , f_m(x_1, x_2, \quad , x_n))$$

one defines the Jacobian matrix as the $m \times n$ matrix J where $J_{ij} = \partial f_i / \partial x_j$

We will need the following two results

Theorem 5 *Suppose $F \quad \mathbb{R}^n \rightarrow \mathbb{R}^m$ is continuously differentiable, and $a, b \in \mathbb{R}^n$ Then*

$$F(b) = F(a) + \int_0^1 J(a + \theta(b-a))(b-a)d\theta,$$

where J is the Jacobian matrix of F

The above theorem is the second fundamental theorem of calculus The next theorem extends the triangle inequality obeyed by norms to integrals

[13] Q-linear convergence means that $\lim_{k\to\infty} \frac{\|x^{(k+1)}-x^*\|}{\|x^{(k)}-x^*\|^q} = \mu$ with $\mu \in (0,1)$ and $q = 1$ We can in fact prove that the iteration rule exhibits faster than Q-linear convergence just like Broyden's method, its convergence is locally Q-superlinear (with $q \approx 1\ 62$, and $\mu > 0$) However, showing this result requires a more intricate argument which is beyond the scope of this thesis

Theorem 6 *If $F \quad \mathbb{R} \to \mathbb{R}^n$ is integrable over the interval $[a, b]$, then*

$$\left\| \int_a^b F(t)dt \right\| \leq \int_a^b \|F(t)\| dt \qquad (2\ 66)$$

We also recall the definition of the operator norm of a matrix

Definition 8 *If $A \in \mathbb{R}^{m \times n}$, the norm of A is defined as*

$$\|A\| = \max \left\{ \frac{\|Ax\|}{\|x\|} \quad x \in \mathbb{R}^n, x \neq 0 \right\}$$

The norm defined above has the following properties

1 It is a norm on the space $\mathbb{R}^{m \times n}$,

2 $\|Ax\| \leq \|A\|\|x\|$ for all $A \in \mathbb{R}^{m \times n}, x \in \mathbb{R}^n$,

3 $\|AB\| \leq \|A\|\|B\|$ for all $A \in \mathbb{R}^{m \times n}$, $B \in \mathbb{R}^{n \times p}$

The following is a well-known theorem from Functional analysis

Theorem 7 *Suppose $J \quad \mathbb{R}^m \to \mathbb{R}^{n \times m}$ is a continuous matrix-valued function If $J(x^*)$ is nonsingular, then there exists a $\delta > 0$ such that, for all $x \in \mathbb{R}^m$ with $\|x - x^*\| < \delta$, $J(x)$ is nonsingular and*

$$\|J(x)^{-1}\| < 2\|J(x^*)^{-1}\|$$

Proof (Sketch)The first part follows from the fact that if $J(x^*)$ is non-singular, then $\det J(x^*) \neq 0$ and consequently there is a neighborhood of x^* where the determinant does not vanish (polynomials define continuous maps) The latter part follows from the fact that if the map $x \mapsto J(x)$ is continuous then so is the map $x \mapsto J(x)^{-1}$ whenever the latter map is defined □

Definition 9 *Suppose F $\mathbb{R}^n \to \mathbb{R}^m$ Then F is said to be Lipschitz continuous on $S \subseteq \mathbb{R}^n$ if there exists a positive constant T such that*

$$||F(x) - F(y)|| \leq T||x - y||, \textit{ for all,} y \in S$$

This definition can also be applied to a matrix-valued function F $\mathbb{R}^n \to \mathbb{R}^{m \times n}$ using a matrix norm to $||F(x) - F(y)||$

The usual Newton iteration is phrased as

$$x^{(k+1)} = x^{(k)} - J(x^{(k)})^{-1}F(x^{(k)}) \qquad (2\ 67)$$

The Newton iteration is known to converge to a root, x^*, of the function F if we start the iteration close enough to x^* (such that the Jacobian is non-singular)

We wish to analyze the convergence of the following update rule

$$x^{(k+1)} = x^{(k)} - D(x^{(k)})^{-1}F(x^{(k)}), \qquad (2\ 68)$$

where D is the diagonal sub-matrix of the Jacobian To this end, we define the matrix L by the rule $J(x) = D(x) + L(x)$, i e , L comprises of the off-diagonal partial derivatives in the Jacobian

We will show that if we are in the situation that J and D are Lipschitz continuous and that $L(x^*) = 0$ (is the zero matrix), then the above iteration rule also converges to the root x^* as long as we start close enough to the root

Subtractıng x^* from both sıdes of equatıon (2 68) and notıng that $F(x^*) = 0$ we have

$$\begin{aligned} x^{(k+1)} - x^* &= x^{(k)} - x^* - D(x^{(k)})^{-1} F(x^{(k)}) \\ &= x^{(k)} - x^* - D(x^{(k)})^{-1} \left(F(x^{(k)}) - F(x^*) \right) \end{aligned}$$

We now use Theorem 5 to estımate $F(x^{(k)}) - F(x^*)$

$$\begin{aligned} & F(x^{(k)}) - F(x^*) \\ & = \int_0^1 J(x^* + \theta(x^{(k)} - x^*))(x^{(k)} - x^*) d\theta \\ & = \int_0^1 J(x^*)(x^{(k)} - x^*) d\theta \\ & + \int_0^1 \left(J(x^* + \theta(x^{(k)} - x^*)) - J(x^*) \right) (x^{(k)} - x^*) d\theta \\ & = J(x^*)(x^{(k)} - x^*) \\ & + \int_0^1 \left(J(x^* + \theta(x^{(k)} - x^*)) - J(x^*) \right) (x^{(k)} - x^*) d\theta \end{aligned}$$

Assumıng $L(x^*) = 0$ we have

$$\begin{aligned} F(x^{(k)}) - F(x^*) = D(x^*)(x^{(k)} - x^*) \\ + \int_0^1 \left(J(x^* + \theta(x^{(k)} - x^*)) - J(x^*) \right) (x^{(k)} - x^*) d\theta \end{aligned}$$

Therefore,

$$
\begin{aligned}
&\left\|F(x^{(k)}) - F(x^*) - D(x^*)(x^{(k)} - x^*)\right\| \\
&= \left\|\int_0^1 \left(J(x^* + \theta(x^{(k)} - x^*)) - J(x^*)\right)(x^{(k)} - x^*)d\theta\right\| \\
&\leq \int_0^1 \left\|\left(J(x^* + \theta(x^{(k)} - x^*)) - J(x^*)\right)(x^{(k)} - x^*)d\theta\right\| \\
&\leq \int_0^1 \left\|J(x^* + \theta(x^{(k)} - x^*)) - J(x^*)\right\| \|x^{(k)} - x^*\| d\theta \\
&\leq \int_0^1 T_J\theta\|x^{(k)} - x^*\|^2 d\theta \text{ (usıng Lıpschıtz contınuıty of } J\text{)} \\
&\leq \frac{T_J}{2}\|x^{(k)} - x^*\|^2
\end{aligned}
$$

We now have

$$
\begin{aligned}
&x^{(k+1)} - x^* \\
&= x^{(k)} - x^* - D(x^{(k)})^{-1}(F(x^{(k)} - F(x^*)) \\
&= x^{(k)} - x^* - D(x^{(k)})^{-1}[D(x^*)(x^{(k)} - x^*) \\
&\quad + F(x^{(k)}) - F(x^*) - D(x^*)(x^{(k)} - x^*] \\
&= \left(I - D(x^{(k)})^{-1}D(x^*)\right)(x^{(k)} - x^*) \\
&\quad - D(x^{(k)})^{-1}\left(F(x^{(k)}) - F(x^*) - D(x^*)(x^{(k)} - x^*)\right)
\end{aligned}
$$

Now applying norms on both sides

$$
\begin{aligned}
&\|x^{(k+1)} - x^*\| \\
&\leq \left\| \left(I - D(x^{(k)})^{-1} D(x^*)\right)(x^{(k)} - x^*) \right\| \\
&\quad + \left\| D(x^{(k)})^{-1} \left(F(x^{(k)}) - F(x^*) - D(x^*)(x^{(k)} - x^*)\right) \right\| \\
&\leq \left\| I - D(x^{(k)})^{-1} D(x^*) \right\| \left\| x^{(k)} - x^* \right\| \\
&\quad + \left\| D(x^{(k)})^{-1} \right\| \left\| F(x^{(k)}) - F(x^*) - D(x^*)(x^{(k)} - x^*) \right\| \\
&\leq \left\| I - D(x^{(k)})^{-1} D(x^*) \right\| \left\| x^{(k)} - x^* \right\| \\
&\quad + \frac{T_J}{2} \left\| D(x^{(k)})^{-1} \right\| \left\| x^{(k)} - x^* \right\|^2
\end{aligned}
$$

We are assuming that D is also a Lipschitz continuous map

$$
\begin{aligned}
\left\| I - D(x^{(k)})^{-1} D(x^*) \right\| &= \left\| D(x^{(k)})^{-1} \left(D(x^{(k)}) - D(x^*)\right) \right\| \\
&\leq \left\| D(x^{(k)})^{-1} \right\| \left\| D(x^{(k)}) - D(x^*) \right\| \\
&\leq T_D \left\| D(x^{(k)})^{-1} \right\| \left\| x^{(k)} - x^* \right\|
\end{aligned}
$$

Thus we have

$$
\|x^{(k+1)} - x^*\| \leq \frac{3T}{2} \|D(x^{(k)})^{-1}\| \|x^{(k)} - x^*\|^2,
$$

where we have set $T = \max\{T_J, T_D\}$

If $x^{(k)}$ is sufficiently close to x^*, then $\|D(x^{(k)})^{-1}\| \leq 2M$,

where $M = \|D(x^*)^{-1}\| = \|J(x^*)^{-1}\|$ by our assumption that $L(x^*) = 0$

Thus if $x^{(k)}$ is sufficiently close to x^*, then $\|x^{(k+1)} - x^*\| \leq 3TM\|x^{(k)} - x^*\|^2$

Moreover, if

$$\|x^{(k)} - x^*\| < \frac{1}{6TM},$$

then

$$\|x^{(k+1)} - x^*\| < \frac{1}{2}\|x^{(k)} - x^*\|$$

This completes the proof of Theorem 4

The problem one faces when trying to apply the secant method to higher dimensions is that the system of equations provided by $J_k \quad (x^{(k)} - x^{(k-1)}) \simeq F(x^{(k)}) - F(x^{(k-1)})$ (where J_k is the current estimate of the Jacobian) is under determined However, if one uses the diagonal approximation to the Jacobian, then the system is fully determined What Theorem 4 says is that under certain conditions, using the diagonal sub-matrix of the Jacobian instead of the full Jacobian in the given iteration rule, still leads to convergence to a root of the function

Equipped with Theorem 4, it is now easy to prove that the price update algorithm given in Definition 6 converges to a buffer equilibrium We only need to consider the update algorithm for resource prices $p_{\overline{U}}$ and $p_{\overline{D}}$ because the price for space remains constant at 1 Consider the function $g(\ )$, and as before, J is the Jacobian of $g(\ )$, D is the diagonal sub-matrix of J, and L is defined by the rule $J(x) = D(x) + L(x)$

Corollary 2 *Consider the price update algorithm given in Definition 6 If $g(\ )$ is a continuously differentiable function, $p^{(0)}$ is chosen close enough to a root p^* of $g(\ )$, the Jacobian $J(p^*)$ is non-singular, J and D are Lipschitz continuous, and $L(p^*) = 0$, then the price update algorithm converges to an equilibrium price vector p^*, and the rate of convergence is at least Q-linear*

Proof. We have shown in Section 2.5.4 that if we find a price vector p^* such that $g(p^*) = 0$, then we have reached a buffer equilibrium. Thus, we only have to show that the price update algorithm converges to a root of the function $g(\cdot)$. Now, note that the price update algorithm provided in Definition 6 defines a quasi-Newton iteration rule that uses the diagonal sub-matrix of the Jacobian of the function $g(\cdot)$, equivalent to the iteration rule given in equation (2.65). By Theorem 4, that iteration rule converges locally to a root of $g(\cdot)$, and the rate of convergence is at least Q-linear. □

One might wonder how restrictive the conditions of Theorem 4 and Corollary 2 are. The condition that the matrices J and D be Lipschitz continuous puts upper bounds on how fast the partial derivatives of the function can change. One can relax this assumption to just that of J and D being Lipschitz continuous in a neighborhood of the root without affecting the conclusions of the theorem and corollary. Local Lipschitz continuity near the neighborhood of the root seems like a plausible condition for $g(\cdot)$ to satisfy because it is hard to envision wild changes in the function near an equilibrium point. The non-singularity of $J(p^*)$ means that our function does not have a higher order zero at the equilibrium point. It is likely that our algorithm would still converge even if this assumption fails, but we do not have a proof of this. The local convergence of our method is an aspect we share with all Newton's methods operating in multiple dimensions, and this is the most worrisome property as well as the hardest to get a handle on. If $||J(p^*)^{-1}||$ and Lipschitz constants of J and D around p^* are all small, then the basin of convergence is large. However, it seems that only experimental evidence can validate whether this assumption is reasonable in our situation.

2.7 Usability Study

In this section, we describe some of the results from a formative usability study of our system with 16 users [14] Our main goal in the usability study was to understand whether the market user interface we propose for the P2P backup system is a usable instantiation of the hidden market paradigm Before describing the results, we give a brief summary of the study set-up and the methodology

2.7.1 Set-up

The UI design process included an early exploratory study (with 6 users) and a pilot study (with 6 users) Upon completion of an iterative UI design phase, we recruited 16 users (8 females) from the Greater Seattle area for the usability study All of the users had some college education and used a computer for at least 10 hours per week The average age of our participants was about 39, ranging from 22 to 66 years old None of the users worked for the same company, none of them were usability experts and none of them had used a P2P backup system before All of the users understood the meaning of "backing up your files" before coming to the study, however only a few of them had used server-based online backup systems before We recruited two different groups of users novices and experts Experts were screened to be users who had used P2P file sharing software and modified the maximum bandwidth limits of their client in the last 5 years We also ensured they had some idea about the speeds of an average home broadband connection Novices were screened such that they did not have technical jobs, were not sophisticated enough to set-up a wireless

[14] See [93] for the full study

router by themselves, and had never adjusted the maximum bandwidth limits of a P2P file sharing client.

In this work we are particularly interested in evaluating the "advanced settings" version of the UI. Thus, our true target group of users was in fact the experts group. However, we included the novice users to make sure we identified all of the problems of the UI or the system in general that might not be found when only testing expert users. We had 8 experts and 8 novices. We ran one participant at a time with each session lasting about 1.5 hours. The users filled out a pre-study questionnaire (20 minutes), completed a series of interactive tasks using the UI (45 minutes), and then completed another post-study survey (20 minutes). We ran the software on a single 3 GHZ Dell computer at full resolution using a 20" 1600x1200 Syncmaster display.

2.7.2 Methodology

The purpose of the usability study was to evaluate how users understand the hidden market UI, which mental models are invoked and whether users can successfully interact with the market. Note that during the study, the users interacted with the real P2P backup client software that was connected via TCP to the P2P server application and to 100 other simulated clients. We started the users off with two warm-up tasks. First, they had to perform one backup using the software. Second, they had to open the settings window and answer a series of questions regarding the information they saw.

Upon completion of the warm-up phase, we gave the study participants 11 tasks, each consisting of a user scenario with hypothetical preferences, and a description of the goal setting for that user We chose tasks with varying complexity and we also tested different mental models in different tasks For example, Scenario 1 was the most simple one, asking the user to "change the settings such that you have approximately 15 GB of free online backup space available " In contrast, Scenario 11 was rather complex, asking the user to "imagine you are a user who likes to download videos and store them on your computer for a while Assume that you need 20 GB of your own hard disk space to store the videos, and obviously you need lots of download bandwidth, but you do not care too much about upload bandwidth Please change your settings so that you have approximately 25 GB of free online backup space available while taking the other constraints into account "

We asked the users to "think out loud" as they performed each task and we made detailed observations during the tasks Using the 11 tasks, we tested four different mental models, i e , aspects of the user's understanding of the market

1 **Give & Take** The users understand they must give some of their resources (on the right side) and get a proportional amount of online backup space in return (on the left side) This was tested using tasks 1 and 2 The test was deemed successful if the users adjusted all settings correctly

2 **Bundling** The users understand the bundle constraints, i e , that they cannot provide zero of any resource because only resource bundles have value This was tested using tasks 3 and 4 The test was deemed successful if the users adjusted all settings correctly

3 **Prices** The users understand that different resources can have different "prices" at different points in time This was tested using tasks 7, 8, and 9 The test was

deemed successful if the users adjusted the settings for task 9 correctly (tasks 7 and 8 gave them practice to learn the model and discover the pricing aspect)

4 **Bundling (Learned)** The users understand the bundle constraints after exploring the UI for a while, i e , after a certain learning period This was tested using tasks 10 and 11 The test was deemed successful if the users adjusted all settings correctly

Note that the tasks were set-up such that finding the correct setting by coincidence was unlikely The correct setting was not a natural focal point so that the user researcher could easily decide whether the participant had truly understood the task (and thus the right mental model had been activated) or not Of course, the "think out loud" method also helped determining the result of a test For example, when testing the understanding of the bundle constraints, if a user said something like "I see, I obviously cannot give 5GB of space without giving any bandwidth, thus I choose to supply the minimum amount of bandwidth I have to give," then this counted as sufficient understanding of the bundle constraints The rare cases where a user had coincidentally chosen the correct settings but did not display sufficient understanding of the problem were also deemed to be failures in our experiment

2.7.3 Results

Table 2 7 3 summarizes the results from the usability study, evaluating whether the 4 different mental models have been successfully activated or not It turns out that the basic aspects of the UI were understood by all users (1 Give & Take) However, the first time the users faced a combinatorial task, e g , "minimize your

Table 2 2 Results from the Usability Study Number of Users Falling into Comprehension Categories

Category	Mental Model	Experts	Novices	Total
1	Give & Take	8/8	8/8	16/16
2	Bundling	4/8	5/8	9/16
3	Prices	5/8	2/8	7/16
4	Bundling (Learned)	5/8	6/8	11/16

upload bandwidth while maintaining at least 15 GB of free online backup space", only 9 out of 16 users completely understood the problem and found the optimal settings The understanding of the bundle constraints of the market improved towards the end of the study, showing that a certain learning effect had occurred In particular, 2 of the users that had not understood the bundle constraints at the beginning, understood them well at the end of the study, leading to 11/16 successful outcomes for "Bundling (Learned)"

The most difficult tasks for the users were certainly the ones testing their understanding of prices because this required three steps from them first, discovering that different resources had different prices, second, understanding the implication for their supply of resources, and then third, choosing the optimal supply settings for themselves given current prices Only 7 out of 16 users successfully completed all three steps, and thus were deemed to understand the pricing aspect

One immediate finding is that the performance of the users is uncorrelated with the way we had segmented them into experts or novices in advance (see Table 2 7 3) Thus, prior experience with P2P file sharing software did not seem matter Instead, anecdotal evidence suggests that those users whose jobs or education involved some mathematical modeling seemed to understand the concepts underlying the UI faster

Thıs makes sense, gıven that some of the tasks were relatıvely complex and requıred a good, somewhat analytıcal understandıng of the UI However, a factor that ıs dıfficult to measure but seemed to play an ımportant role ın thıs study ıs the users' curıosıty, ı e , how much the users lıked to play wıth the slıders untıl they figured out how the ınterface worked Thıs aspect ıs partıcularly ımportant for category 4, ı e , the prıcıng aspect The less curıous users who dıd not explore the settıngs space as much as the others were also the ones that dıd not dıscover the fact that dıfferent resources have dıfferent prıces, and consequently faıled to solve the prıcıng tasks optımally

Upon completıon of the ınteractıve part of the study we asked the users about theır experıence wıth the UI Despıte the fact that almost every user had dıfficultıes wıth at least one of the tasks, the user feedback was largely posıtıve Most users thought that the software made ıt easy to perform the tasks they were gıven (wıth a 3 8 average on a 5-poınt Lıkert scale, wıth 1=strongly dısagree and 5=strongly agree) and they ındıcated that they enȷoyed usıng the UI (3 8 average on the same 5-poınt Lıkert scale) Most users were pretty confident that they completed the tasks successfully (wıth an average 4 0 on the same 5-poınt Lıkert scale) The users lıked the graphıcal/vısual representatıon of the concepts ınvolved Despıte some dıfficultıes wıth solvıng the tasks, the users thought that the UI was "clean, sımple, ıntuıtıve and easy to use " All users lıked the ease of usıng the bar chart to choose the desıred amount of free onlıne backup space Furthermore, they lıked that the UI gave ımmedıate feedback regardıng the consequences of theır choıces The users prımarıly dıslıked that ıt took them a whıle to understand the concept and logıc behınd the slıders

From the pre-study questıonnaıre we have seen that for a large number of users, P2P backup systems could be an attractıve alternatıve to server-based systems However, thıs stıll leaves open the questıon how users perceıve the trade-off between a market-based system (that gıves users more freedom ın choosıng dıfferent combınatıons of supplıed resources) vs a non-market-based system (that has a sımpler UI) In the post-study questıonnaıre we asked the users twıce to compare the two optıons The first tıme we asked the questıon, we gave no addıtıonal ınformatıon beforehand But before askıng them for the second tıme, we descrıbed a partıcular scenarıo hıghlıghtıng the fact that the market-based system gıves the users more freedom ın choosıng the supplıed resources The results were that, when asked for the first tıme, the users already slıghtly preferred the market-based system (3 3 on a 5-poınt Lıkert scale, wıth 1=definıtely prefer the sımpler UI and 5=definıtely prefer the complex UI) After descrıbıng the hypothetıcal scenarıo where the non-market-based system would lead to a degraded user experıence, the average score rose to 4 0 We ınterpret these results as follows a prıorı, some users do not see the advantage of a market-based system However, after understandıng the possıble lımıtatıons of the non-market-based system, they realıze the benefit of the ıncreased freedom ın choosıng what to supply, and they value thıs benefit hıgher than the dısutılıty from the addıtıonal complexıty of the UI

2.8 Summary

In thıs chapter, we have presented the desıgn and analysıs of a novel resource exchange market underlyıng a P2P backup applıcatıon We have also used the P2P backup market as a first case study of a new market design paradigm which we call *hidden market design* We propose hidden markets for the design of electronic systems in domains with many non-experts users and where markets might be unnatural To successfully hide the market complexities from the users in our system, new techniques at the intersection of market design and user interface design were necessary At all times, for the model formulation and the theoretical analysis, our focus was on the actual implemented P2P backup system, which we have successfully tested in alpha version

In contrast to existing P2P backup systems, our design gives users the freedom to supply different ratios of resources This introduces the problem that without properly motivating the users to supply those resources that are currently scarce, the system might not have enough supply to satisfy demand, which motivates the use of a P2P resource market While existing work on P2P data economies has generally designed markets that balance supply and demand in equilibrium, our market is designed to work well, even out of equilibrium The users are not required to continuously update their supply and demand Instead, we provide a *hidden market UI* that lets them choose bounds on their maximum supply in return for being allowed to consume a certain maximum amount of backup services The UI completely hides the users' account balances and payments, and only indirectly exposes the current market prices A key contribution is the new slider control that we developed which we use to display the bundles constraints to the users in an indirect way The sliders also ensure that the users can only choose supply settings that satisfy certain resource ratio constraints, which allows us to provide the users a linear interaction with the system

To maximize the safety of the system out of equilibrium, we have declared as our target to maximize the overall size of the buffer between current demand and maximum supply We have introduced the *buffer equilibrium* concept and shown that, under certain assumptions, the size of the buffer is maximal in the buffer equilibrium The economic analysis of the market required the introduction of composite resources on the supply side, and the careful study of the system's production technology, to convert the market into a pure exchange economy In this model, we have proved that a buffer equilibrium is always guaranteed to exist This result also holds if a certain percentage of the user population is price-insensitive or even adversarial However, we have shown that the more freedom we give users in choosing their supply settings, the less robust the system becomes against adversarial attacks We have explained how the theoretical equilibrium analysis actually has an important market design implication The theorem regarding adversarial users provides the market designer with a concrete formula how large the system's slack factor can be, given a certain belief about the maximal percentage of adversarial users in the population

To prove uniqueness of the buffer equilibrium, we needed two additional assumptions that are very reasonable in our domain We have explained why it makes sense to assume that services are perfect complements, and how that affects even the supply of resources via the flow constraints By making a relatively weak assumption regarding how the relative supply of resources changes upon price changes, we were able to prove uniqueness of the buffer equilibrium An interesting corollary of the uniqueness result was that the market operator has limited control over the size of the buffer via price updates alone However, we have shown how changes to the UI design can resolve this problem by enforcing certain demand-side buffers in the UI, the market operator can ensure any desired supply-side buffer We have proposed a price update algorithm that only requires daily aggregate supply and demand information, and proved that it converges linearly to the buffer equilibrium, given that initial prices are chosen close enough to equilibrium prices

To evaluate the hidden market UI, we have performed a formative usability study of our system Our main goal was to determine whether the UI activates the right mental model, and whether the users can successfully interact with the hidden market Overall, the results were encouraging and show promise for the hidden market design paradigm Most users intuitively understood the give & take principle as well as the bundle constraints of the market It was particularly positive to see that even after the users had used he system for 45 minutes, they had not realized they were interacting with a market-based system, yet were able to complete most of the tasks successfully This shows that we have successfully hidden the market The pricing aspect, however, was difficult for some users, i e , they either never learned that different resources have different values (prices) in the system, or they were unable to exploit this insight properly We are currently investigating new user interfaces that still hide the market from the users, but provide them with slightly more information and guidance regarding the pricing aspect

In ongoing work we are also analyzing different ways to monetize the P2P market platform There is an easy and elegant way to generate revenue while still running the market using a virtual currency the market operator can charge a small tax on each virtual currency transaction and use the surplus to sell backup services on a secondary market for real money More specifically, the P2P users would not have to be involved in any real-money transactions and the customers from the secondary market would buy backup services like they would from a centralized data center If real monetary transaction are made possible and deemed desirable in the P2P system itself, then we can also open the whole market for real monetary payments On the one side, users will then be able to pay for their consumption of services by either providing their own resources or by paying with real money, and on the other side, users will then also be able to earn real money by supplying their resources With this design, the market operator could generate revenue by charging a tax on each virtual currency transaction and by charging a tax on each real-money transaction

Remember that for the P2P backup system we described in this chapter, it was essential that its market and its UI were designed in concert Successfully hiding the market complexities from the user was partly a result of certain UI design elements, and partly a result of the design of the underlying market Thus, a key finding from this research project was the understanding of the important connection between the economic market design and the user interface However, here we only studied one particular user interface for this market and showed that it is usable for real human users In the next chapter, we take this a step further, performing a *principled* study of the market user interface design space We study the effect of different UI designs on user's decision making performance and the market's efficiency

Chapter 3

Market User Interface Design

3.1 Introduction[1]

Electronic markets are becoming more and more pervasive but a remaining research challenge is to develop user interfaces (UIs) to promote effective outcomes for users This can be quite a challenge given limited user attention and exacerbated by markets that can easily present users with a large number of choices Indeed, recent research has shown that having more choices does not always lead to better outcomes For example, more choices in employees' 401(k) plans can lead to fewer participation and thus significant losses in savings [46] Overall, 401(k) plan design has a huge impact on employer savings behavior [17] Or consider the now-famous "jam experiment" where Iyengar and Lepper [47] have shown that customers are happier with the choices they make when offered 6 different flavors of jam compared

[1] The material presented in this chapter is based on collaborations with David C Parkes, Eric Horvitz, Kamal Jain, Mary Czerwinski, and Desney Tan

to 24 different flavors of jam Schwartz [91] identifies a series of reasons why more choices can lead to decreased satisfaction, including *regret*, *missed opportunities*, *the curse of high expectations*, and *self blame* Sarver [90] derives a formal model of regret anticipation for situations where agents select an alternative from a menu of choices While emotional processes in human decision-making are certainly important, in this chapter we are not concerned about the cause of behavior but focus on modeling the effect of different user interfaces on users' decision-making performance

Traditional economic models assume all agents to be perfectly rational, with unlimited time to make a decision and unbounded computational resources for deliberation In reality, however, humans have cognitive costs, bounded time for decision making (because of opportunity costs) and bounded computational resources We explicitly take these behavioral considerations into account, with the goal to design market user interfaces that make the decision-making task easier for the users and lead to better outcomes Our approach is very much in line with the "choice architecture" idea put forward by Thaler et al [100] In their language, we are designing "choice architectures for electronic markets "

So far, the market design literature has largely ignored the intersection of market design and user interface design However, we argue that this intersection is particularly important for at least four reasons First, the UI is the first point of contact for a user interacting with a new market Second, the choice of the UI constrains the design space for the market designer Third, the UI defines how, and how well, users can express their preferences And fourth, the complexity of the UI defines the cognitive load imposed on the user while interacting with the market Thus, when designing an electronic market populated by end users, it is important to design the market and the user interface in concert, to jointly optimize along both dimensions

3.1.1 Overview of Results

We propose a new research agenda on “market user interfaces’ and present a principled study of the design space A market UI can best be defined via two questions first, what information is displayed to the user? Second, what choices/how many choices are offered to the user? The research question we want to answer is *what is the optimal market user interface given that users have cognitive costs?* In evaluating the effectiveness of a market user interface we consider the ability of the market to efficiently allocate resources given user behavior

We focus on the challenges in market user interface design for allocating 3G bandwidth, the demand for which is projected to continue to grow exponentially over the next few years [79] In particular, we present the results of a systematic, empirical exploration of the effect that different UI design levers have on user's performance in economic decision making The experimental set-up considers a user with uncertainty about future value for resource allocation, an inter-temporal budget constraint, and a user interface that offers some number of choices of bandwidth in any given period, each for a particular price Formally, the decision problem facing a user is modeled as a Markov Decision Process (MDP), the solution to which provides the gold standard against which we compare user behavior

We first explore parts of the design space manually, by experimenting with varying the number of choices offered to users, and considering the effect of offering fixed vs dynamically changing prices These results offer general insight, we think for the first time in such detail, into how well humans can determine optimal policies in MDPs under time pressure Our findings indicate that users are surprisingly good at coming up with good decision polices for the sequential optimization problem We show that their actions exhibit a high degree of rationality in the sense of being highly correlated with the Q-values of the game However, we also show how various behavioral factors influence the users' decision making process Some effects are particularly strong, including loss aversion which raises concerns about users general tendency, at least in some situations, to take short-term winnings ignoring potential long-term losses

In a second step, we then use computation to *automate* the market UI optimization process Based on the results from the first experiment, we train a behavioral user model In particular, we adopt a maximum-likelihood fit to a quantal best-response user model [108], which is a well-studied model of behavioral decision making The model is a single-parameter, soft-max model, allowing for a range of behavior from random to best-response, where the true utility for each choice is induced as the solution to the MDP model of the user problem Based on this maximum likelihood fit, we then feed this user model into an optimization algorithm, which is used to identify the optimal market UI given the learned behavioral model A second experiment evaluates the effect of the re-optimization algorithm Here we find that the re-optimization increased the user's probability of selecting the optimal choice However, the data suggests that the re-optimization algorithm took away too much value, in particular for the *more rational users*, while no statistically significant effect was observed for the *less rational users*

3.1.2 Related work

Prior research has identified a series of behavioral effects in users' decision making. Buscher et al. [11] show that the amount of *visual attention* users spend on different parts of a web page significantly depends on the task type and the quality of the information provided. Dumais et al. [24] show that these "gaze patterns" differ significantly from user to user, suggesting that different user interfaces may be optimal for different groups of users. In a study of the cognitive costs associated with decision making, Chabris et al. [13] show that users allocate time for a decision-making task according to cost-benefit principles. Thus, time is generally costly, and consequently more complex UIs put additional costs on users.

Horvitz and Barry [43] present a methodology for the optimal design of human-computer interfaces for time-critical applications in non-market-based domains. They introduce the concept of *expected value of revealed information*, trading-off the costs of cognitive burden with the benefits of added information. Johnson et al. [49] show that the way information is displayed, in particular probability values (fractional vs. decimal), has an impact on user decision making and their information processing strategies. The authors briefly discuss the implications of their findings for the design of information displays.

In our own previous work [95], we have introduced the goal of designing *simple* and *easy-to-use* interfaces for electronic markets, in particular for domains where users repeatedly make decisions of small individual value. In Seuken et al. [93], we present one detailed case-study of a novel market user interface for a P2P backup market. We demonstrate that it is possible to hide many of the market's complexities, while maintaining a market's efficiency However, there we did not study the effect of *changing* aspects of a UI on a user's decision-making performance, which is the focus of this chapter

The work most closely related to ours is SUPPLE, introduced by Gajos et al [37], who present a system that can *automatically* generate user interfaces that are adapted to a person's devices, tasks, preferences, and abilities They formulate the UI generation as a computational optimization problem, and find that automatically-generated UIs can lead to significantly better performance compared to manufacturer's defaults While their approach is very much in line with our long-term goal of "automatic UI optimization", they optimize their interfaces for accuracy, speed of use, and user's subjective preferences for UI layouts In contrast, we optimize for *decision quality* in market-based environments where users are dealing with values, prices, and budgets We build a parameterized *behavioral* model of users while they build a model of users' pointing and dragging performance A significant part of this chapter is about determining which behavioral factors are most important for the effectiveness of decision-making in a market-based environment

3.1.3 Outline

The remainder of the chapter is structured as follows In the next section we describe the design of the market game that is the basis of our experiment After motivating the domain of bandwidth allocation for smartphones, we describe the implementation of the market game in detail We describe how the game can be modeled as a Markov Decision Process, and how the quantal-response model can be used as a behavıoral model that predıcts user play ın thıs domaın In Sectıon 3 3 we descrıbe the experıment desıgn Thıs ıncludes a dıscussıon of the four dıfferent desıgn levers, the tıme lımıts we ımposed, the selectıon of the subject pool and the experımental set-up, as well as a detaıled descrıptıon of the dıfferent treatment varıatıons across users In Sectıon 3 4 we present the results of our statıstıcal data analysıs We first present the results based on analyzıng users' decısıons ın ındıvıdual rounds, whıch allows us to study whıch factors are most predıctıve for whether users find the optımal choıce or not Then we move on to the analysıs of whole games, studyıng the effect of the four dıfferent desıgn levers on users' Realızed Efficıency We conclude ın Sectıon 3 5

3.2 Game Design: Bandwidth Allocation over Time

3.2.1 Setting: A 3G Bandwidth Market

We sıtuate the experıment ın the smartphone domaın to gıve our partıcıpants some context for the game they are playıng Consıder the 3G bandwıdth needed to access the Internet on a smartphone Current research predıcts that the demand for 3G bandwıdth wıll contınue to grow exponentıally over the next few years and that ıt wıll be ınfeasıble for the network operators to update theır ınfrastructure fast enough to satısfy future demands [79] Another sıgn that 3G bandwıdth ıs gettıng scarce ıs that network provıders lıke AT&T are begınnıng to drop theır unlımıted data plans The common approach for addressıng the problem of bandwıdth demand temporarıly exceedıng supply ıs to slow down every user ın the network and to ımpose data usage

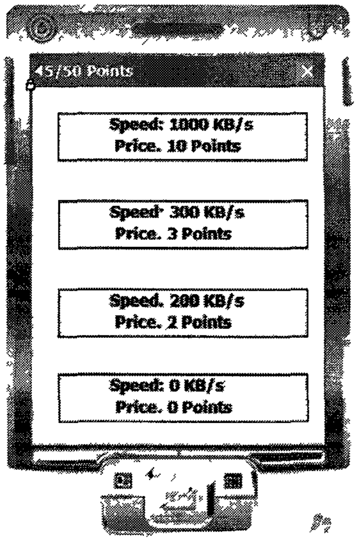


Figure 3 1 Mockup of the Bandwidth Market UI

constraints via fixed upper limits (e g , 200MB per month for one of AT&T's current data plans) Obviously, this introduces large economic inefficiencies, because different users have different values for high speed vs low speed Internet access at different points in time The current approach simply ignores this

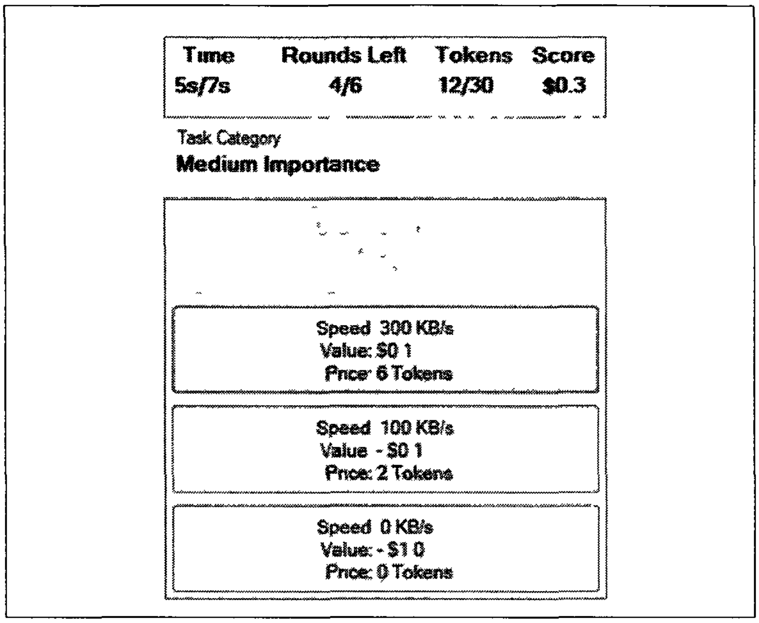
Imagine a market-based solution to the 3G bandwidth problem The main premise is that users sometimes do tasks of high importance (e g , send an email attachment to their boss) and sometimes of low importance (e g , update their Facebook status) If we assume that users are willing to accept low performance now for high performance later, then we can optimize the allocation of bandwidth use by shifting excess demand to times of excess supply Of course it is not possible that every user gets high-speed Internet access all of the time Instead, the users' choices must be limited somehow One possibility to achieve this is by giving each user a fixed amount of virtual currency, assumıng users pay a fixed $-amount for theır data plans

Consıder Fıgure 3 1 whıch shows a mock-up applıcatıon for a 3G bandwıdth market Let's assume that at the begınnıng of the month, each users gets 50 poınts, or tokens As long as there ıs more supply than demand, a user doesn't need to spend hıs tokens However, when there ıs excess demand and the user wants to access the Internet, then the screen as shown ın Fıgure 3 1 pops up, requırıng the user to make a choıce Each speed level has a dıfferent prıce (ın tokens) For now, we assume that when a user runs out of tokens, he gets the lowest possıble servıce qualıty (whıch could mean no access or some very slow connectıon) Note that we do not concern ourselves wıth the economıcs of thıs market, nor wıth the questıon as to whether users should be allowed to pay money to buy more tokens or not Our goal ıs not to put forward thıs partıcular market desıgn as the best solutıon for thıs domaın Instead, we merely use thıs hypothetıcal market applıcatıon as a motıvatıng domaın for our experımental study

Thıs domaın ıs partıcularly suıtable to studyıng market UIs because we can easıly change many parameters of the UI, ıncludıng the number of choıces, whether prıces stay fixed or keep changıng, and the partıcular composıtıon of the choıce set In our lab experıment we studıed the effect of changıng varıous desıgn parameters on how well users were able to make good decısıons

3.2.2 Game Design

Fıgure 3 2 shows a screenshot of the market game that we desıgned, mırrorıng the mockup of the market applıcatıon from Fıgure 3 1 Each game has exactly 6 rounds



Time	Rounds Left	Tokens	Score
5s/7s	4/6	12/30	$0.3

Task Category
Medium Importance

Speed 300 KB/s
Value: $0 1
Price 6 Tokens

Speed 100 KB/s
Value - $0 1
Price: 2 Tokens

Speed 0 KB/s
Value: - $1 0
Price: 0 Tokens

Figure 3 2 Screenshot of the market game used in the experiment

At the beginning of a game, a user always has 30 tokens available to spend over the course of the 6 rounds In each round, the user has to select one of the choices Each choice (i e , a button in Figure 3 2) has three lines the first line shows the *speed* of that choice in KB/s The second line shows the *value* of that choice in $ The value represents the $ amount that is added to a user's *score* when that choice is selected The third line shows the *price* of that choice in tokens When the user selects a particular choice, the corresponding number of tokens is subtracted from his current budget Now let's look at the top of the application On the far right, it shows the user's score in the current game After every choice the user makes, the corresponding value of that choice is added to this score which ultimately determines the final game score after the 6th round

Next to the score is a label displaying the user's current budget, which always starts at 30 ın round 1 and then goes down towards 0 as the user spends tokens As a user's budget decreases durıng a game, those choıces that have a prıce hıgher than the user's current budget become unavaılable and are greyed out (as ıs the case for the top choıce ın Fıgure 3 2) To the left of the user's budget the game shows the number of rounds that are left untıl the game ıs over Fınally, at the very left of the wındow, we show the user how much tıme he has left to make a decısıon ın thıs partıcular round (e g , ın Fıgure 3 2 the user stıll has 5 seconds left to make a decısıon ın the current round)

Between the ınformatıon panel at the top and the first choıce button ıs the game's *task category label* In every round, the user can be ın one of three task categorıes 1) hıgh ımportance, 2) medıum ımportance, and 3) low ımportance (note that thıs corresponds to the orıgınal premıse that users are doıng tasks of dıfferent ımportance at dıfferent poınts ın tıme) Every round, one of these three categorıes ıs chosen randomly wıth probabılıty 1/3 The task category determınes the dıstrıbutıon of the values for the four choıces that the user can expect to see ın thıs round Table 3 2 2 shows an overvıew of the values the user can expect ın the three categorıes for a game wıth 4 choıces [2] As one would expect, selectıng the hıgher speed choıces ın the "hıgh ımportance" category gıves the user very hıgh value, whıle choosıng low speeds ın the hıgh ımportance category leads to a severe penalty Compare that to the "low ımportance" category, where the user can earn less value for selectıng hıgh speeds, but ıs also penalızed less for selectıng the lowest speed

[2] Note that the values shown ın Table 3 2 2 are only the averages of the values the user can expect to see Each valuatıon ıs perturbed upwards or downwards wıth probabılıty 1/3 each, to ıntroduce addıtıonal stochastıcıty ın the game, and avoıd that the users can memorıze a fixed set of values for each task category

Table 3 1 The Values ın the 3 dıfferent Task Categorıes

	Hıgh Imp	Medıum Imp	Low Imp
900 KB/s	$1 7	$1 1	$0 4
300 KB/s	$0 5	$0 2	- $0 2
100 KB/s	-$0 3	- $0 3	- $0 5
0 KB/s	- $1	- $0 9	- $0 8

The user's problem when playıng the game ıs to allocate the budget of 30 tokens optımally over 6 rounds, not knowıng whıch categorıes wıth whıch exact values wıll come up ın future rounds In some of our experıments, we randomly vary the prıces charged for each of the choıces from round to round Thus, the user may also have uncertaınty about whıch prıce level (out of 3 possıble prıce levels) he wıll be facıng next Thıs problem constıtutes a sequentıal decısıon makıng problem under uncertaınty Note that to play the game optımally, the user only needs to know the values and the prıces of each choıce, but not the *speeds* However, we use the first lıne on each button to dısplay the speed of that choıce to provıde each choıce wıth a natural label and to gıve the UI a lıttle more structure

3.2.3 MDP Formulation and Q-Values

Each game can formally be descrıbed as a finıte-horızon Markov Decısıon Problem (MDP) wıthout dıscountıng

- **State Space** *CurrentRound* × *CurrentBudget* × *CurrentCategory* × *CurrentValueVarıatıon* × *CurrentPrıceLevel*

- **Actıons** Each choıce affordable ın the current round

- **Reward Functıon** The value of each choıce

- **State Transition** The variables *CurrentRound*, *CurrentBudget*, and *CurrentScore* transition deterministically given the selected choice, the other variables *CurrentCategory*, *CurrentValueVariation* and *CurrentPriceLevel* transition stochastically

With six choices and changing prices, the resulting MDP has approximately 1,180,000 states (*Currentround* is between 1 and 6, *CurrentBudget* varies between 30 and 0, *CurrentCategory* varies between 1 and 3, *CurrentValueVariation* varies between 1 and 3^6, denoting for every choice whether the value is perturbed upwards, downwards, or at the normal level, and *CurrentPriceLevel* varies between 1 and 3 In each state there are at most 6 actions possible, thus leading to approximately 7 million state-action pairs Using dynamic programming, we can solve games of this size relatively quickly (in less than 20 seconds) Thus, we can compute the optimal policy, i e , we know exactly, for each possible situation that can arise, which choice is currently best according to the optimal MDP-policy Note that this policy is, of course, computed assuming that the future states are not known, only the transition probabilities as described above are known

Solving for the optimal MDP-policy involves the computation of the *Q-values* for each state-action pair For every state s and action a, the Q-value $Q(s, a)$ denotes the expected value for taking action a in state s, and following the optimal MDP-policy for every subsequent round Thus, the optimal action in each state is the action with the highest Q-value, and by comparing the differences between the Q-values of two actions, we have a measure of how much "worse in expectation" an action is compared to the optimal action We use this concept repeatedly in the analysis section

3.2.4 The Quantal-Response Model

A well-known theory from behavioral economics asserts that agents are more likely to take an action the higher its value, or equivalently, users are more likely to make errors the smaller the cost for making that error. This can be modeled formally with the *quantal-response model* [66] which predicts the likelihood that a user chooses action a_i to be

$$P(a_i) = \frac{e^{\lambda\, Q(a_i)}}{\sum_{j=0}^{n-1} e^{\lambda\, Q(a_j)}} \tag{3.1}$$

where $Q(a_i)$ denotes the Q-value of action a_i. In this model, the parameter λ is a precision parameter, indicating how sensitive users are to differences between the Q-values. A λ value equal to zero corresponds to random action selection, and $\lambda = \infty$ corresponds to perfectly-rational action selection, i.e., always choosing the optimal action. Based on experimental results, one can compute a maximum-likelihood parameter λ that best fits the data. Equipped with such a λ this provides us with a user model which we can use to optimize the UI for behavioral play (see Wright and Leyton-Brown [108] for a comparison of behavioral models).

3.3 Experiment Design

In this section we describe in detail the experiment design. Most importantly, this includes a detailed description of the 4 UI design levers that we studied. We then discuss the details of the different treatment variations, our subject pool, and the exact experimental set-up and payment structure.

3.3.1 The Four Design Levers

In general, a market UI desıgner has sıgnıficant freedom ın desıgnıng both the user ınterface and aspects of a market for an applıcatıon Consıder agaın Fıgure 3 2, where we dısplay a screenshot of *one partıcular versıon* of the game In our domaın, the desıgn space ıncludes 1) how many choıces do we offer the user 2) what ıs the 3G speed of each choıce ın each sıtuatıon, and 3) what ıs the prıce of each choıce ın each sıtuatıon The only thıng we cannot reasonably control as a market UI desıgner ıs the value a user has for a choıce, because that depends on a user's ıntrınsıc value for speed ın a partıcular moment In our experımental study, we explore thıs desıgn space as completely as possıble and study the followıng four desıgn levers

1 **Number of Choıces** Thıs desıgn lever descrıbes how many choıces (ı e , the number of buttons) were avaılable to the users (3, 4, 5, or 6)

2 **Fıxed vs Changıng Prıces** In the *fixed prıce* treatment, the same choıce always costs the same number of tokens (2 tokens per 100KB/s) In the changıng prıce treatment, one of three prıce levels ıs chosen randomly wıth probabılıty 1/3, where the prıce per 100 KB/s ıs eıther 1 token, 2 tokens, or 3 tokens (thus, 500KB/s cost eıther 5 tokens, 10 tokens, or 15 tokens)

3 **Fıxed vs Adaptıve Choıce Sets** In the *fixed choıce set* treatment, the users always had the same set of choıces avaılable to them ın every round (e g , always 0 KB/s, 100 KB/s, 300 KB/s, and 900KB/s) In the *adaptıve choıce set* treatment, the choıces avaılable to the users varıed from round to round, dependıng on the current category (e g , ın the hıgh category, more hıgh speed choıces were

available, in the low category, more low speed choices were available)

4 **UI Optimization** This design lever describes which method is used to determine the composition of the choice sets (i e , which speed levels are available to the user) In the *optimized for optimal play* treatment, the choice sets are optimized (to maximize the expected score per game) based on the MDP model and assuming optimal play In the *optimized for sub-optimal play* treatment, the choice sets are optimized assuming behavioral play where actions are chosen according to the quantal-response model

3.3.2 Game Complexity and Time Limits

To study the effect of the UI design on a user's ability to make good economic decisions, we need a decision problem with a suitable complexity If the problem is too easy or too hard, then changes to the UI would likely have no effect To create a decision problem with just a few choices that is not too easy to solve, we put a fixed time limit on the users' decision, because prior research has shown that users make worse decisions when under time pressure (see, e g , [36]) With an unlimited amount of time, it shouldn't make a difference whether the user was facing three or six choices in each round However, under time pressure, coming up with a good strategy might be much harder in a more complex UI than in a simple UI We used two different time treatments for each user the first treatment was a fixed time limit per round of 12 seconds If a user doesn't make a choice within 12 seconds, the lowest choice (with 0KB/s for 0 tokens) is chosen and the game transitions to the next round The time resets in every round The second time treatment gave the user 7 seconds per round In both of these *exogenous* time limit treatments, the game started beeping three seconds before the end of a round to remind the user that the he has to make a decision soon Note that letting the time run out generally led to the selection of a very bad choice because the lowest choice was rarely a good choice, and always came with a very negative value

When designing the game, we went through an iterative design process, testing various versions of the game with a group of research interns, until we found the final version of the game as described above We kept adding more and more stochastic transitions to the game until we could not find any simple heuristic for playing the game well We calibrated the game (i e , the size of the budget, the nominal values of the choices, the prices, the number of rounds) in such a way that random play has a highly negative expected score, but that optimal play leads to a score around $1 on average Thus, to play the game well and achieve positive scores, the users had to exert significant cognitive effort and properly take the multi-step stochastic nature of the game into account

The 7-second and the 12-second time limits were also chosen carefully In a series of pre-tests we found that for some users, having less than 7 seconds put them under too much time pressure such that they were essentially unable to play the game Having between 7 seconds and 12 seconds put most of the users under enough time pressure such that it was difficult for them to find the optimal choice, but still gave them enough time such that they could process most of the information available to them

3.3.3 Methodology and Experimental Set-up

We recruited 53 participants (27 males, 26 females) from the Seattle area with non-technical jobs All participants had at least a Bachelors degree and we excluded participants who majored in computer science, economics, statistics, math or physics They were fluent English speakers, had normal (20/20) or corrected-to-normal vision, and were all right-handed All of them used a computer for at least 5 hours per week The median age of our participants was 39, ranging from 22 to 54 None of the participants worked for the same company, but all of them had some familiarity with smartphone interfaces We ran one participant at a time with each session lasting about 1 5 hours The users filled out a pre-study questionnaire (5 minutes), went through a training session where the researcher first explained all the details of the game and then gave the participants the opportunity to play 12 training games (20 minutes), participated in the experiment (55 minutes) and then completed a post-study survey (10 minutes) We ran the software on a single 3 GHZ Dell computer at full screen resolution The participants were compensated for their participation in the study in two ways First, they received a software gratuity that was independent of their performance (users could choose one item from a list of Microsoft software products) Second, they received an Amazon gift card via email with an amount equal to the total score they had achieved over the course of all games they had played The expected score for a random game, assuming perfect play, was around $1 After each game, we show the user his score from the last game and the accumulated score over all games played so far [3] The final giftcard amounts of the 53 users varied between

[3] Note that we had originally 56 participants in our study, but we had to exclude 3 participants from the first experiment (2 males, 1 female) because they did not understand the game well enough

Table 3 2 Desıgn of Experıment 1 Each partıcıpant played 32 games The desıgn lever *Number of Choıces* was a wıthın-subȷect factor, the desıgn lever *Fıxed vs Changıng Prıces* was a between-subȷects factor

Number Of Choıces	12-second game	7-second game
3	4 ×	4 ×
4	4 ×	4 ×
5	4 ×	4 ×
6	4 ×	4 ×

$4 60 and $43 70, wıth a medıan amount of $24 90

3.3.4 Treatments

The study was splıt ınto two separate experıments In Experıment 1 we had 35 out of the 53 partıcıpants, and we tested the desıgn levers *Number of Choıces* and *Fıxed vs Changıng Prıces* *Number of Choıces* was a wıthın-subȷect factor, and *Fıxed vs Changıng Prıces* was a between-subȷect factor We had 18 partıcıpants who only played games wıth fixed prıces, and 17 who only played games wıth changıng prıces Table 3 2 depıcts the experıment desıgn for each ındıvıdual user For each treatment, each user played four games wıth the 12-second tıme lımıt and four games wıth the 7-second tıme lımıt[4] We randomızed the order ın whıch the users played the games wıth 3, 4, 5, or 6 choıces For each of those treatments, every user started wıth the four 12-second games and then played the four 7-second games Thus, every partıcıpant played 32 games wıth 6 rounds each, whıch gıves us a data set wıth a total of 1,120 games or 6,720 rounds from Experıment 1

and achıeved a negatıve overall score

[4] Each user also played another game for each treatment wıth an overall tıme lımıt of 4 mınutes The analysıs of those endogenous tıme games ıs stıll underway

Table 3 3 Design of Experiment 2 Every participant played 32 games Both design levers *Fixed vs Adaptive Choice Sets* and *UI Optimization* were within-subject factors

Treament	12-second game	7-second game
Fixed-Choice-Sets & Optimized-For-Opt	4 ×	4 ×
Adaptive-Choice-Sets & Optimized-For-Opt	4 ×	4 ×
Fixed-Choice-Sets & Optimized-For-SubOpt	4 ×	4 ×
Adaptive-Choice-Sets & Optimized-For-Sub-Opt	4 ×	4 ×

In Experiment 2 we had 18 participants and we tested the design levers *Fixed vs Adaptive Choice Sets* and *UI Optimization*, and both were within-subjects factors See Table 3 3 for a depiction of the experiment design for each individual participant We randomized the order of the 4 different treatments As before, for every treatment, every user played 4 12-second games and 4 7-second games Every participant played 32 games with 6 rounds each which gives us a data set with 576 games or 3456 rounds Thus, from both experiments together, we obtained more than 10,000 data points, where each data point corresponds to a decision that a participant made in one particular game situation

3.4 Analysis and Results

In this section we present a detailed statistical analysis of the experimental data obtained from both experiments

3.4.1 Choice of Regression Models

Using multiple (repeated) measurements from individual users violates the independence assumption of standard (OLS or logistic) regression models, because multiple measurements from the same user are not independent from each other. That is why for all of the statistical analysis of the data we use *Generalized Estimating Equations (GEE)*, an extension of generalized linear models [70] that allows for the analysis of repeated measures or otherwise correlated observations. When analyzing binary decisions (e.g., did the user make the optimal choice) we use the logit link function and the binomial distribution (as in logistic regression). When analyzing scale variables like *value loss*, *efficiency* or *decision time*, we use the identity link function and the Normal distribution (as in linear regression).

To compare the *goodness of fit* of different models, GEE provides the QIC and QICC information criteria [70] which are based on a generalization of the likelihood (comparable to R^2 in OLS regressions, however, here smaller values denote a better fit). We use the QIC value to choose between different correlation structures, and the QICC value to choose between different models (i.e., sets of model terms). We tested a series of correlation structures, including *compound symmetry* and *unstructured*. However, assuming independence led to the smallest QIC values, indicating the best fit. Thus, we always report the results using generalized estimating equations that assume independence. Note that GEE has the nice property that even if the correlation structure is misspecified, the coefficient estimates are still consistent (but may have larger standard errors), which makes using GEE particularly attractive. With the independence assumption, GEE can be seen as an extension of (logistic/linear) regression methods for clustered data Note that there is no widely accepted definition of standardized coefficients for a logistic regression model Thus, when reporting regression results using the logit link function, we only report the non-standardized coefficient estimates B and the corresponding odds ratios $Exp(B)$ Thus, when interpreting the results, we always have to take the standard deviation of the corresponding predictor into account When reporting results using the identity link function (linear regression), we also report the standardized coefficients

In analyzing the data, our general goal is to understand which factors influence whether users make good or bad choices There are two ways we can look at the data First, we can look at the results of the *games*, measure the average efficiency that users achieved per game, and compare how efficiency differed under different treatments Second, we can look at the individual *rounds* of each game, and measure whether users chose the optimal action or not, and which factors influenced their performance Analyzing the individual rounds gives us a more detailed look at what actually happened, because we can take factors into account that change every round, like the Q-value differences, number of choices left, position of the optimal choice, value of the optimal choice, budget, time, etc Thus, we begin our analysis by taking a very close look at the individual rounds, before moving on to the analysis of the games

The actions available to a user in each round have an inherent order based on their Q-values, and we can rank them from best to worst Thus, in the most general model, the dependent variable of the regression model would be the rank of the chosen action *Ordered logistic regression* is a suitable regression model for this case However, this model can only be used when the *proportional odds assumption* is satisfied, which says that the relationship between all pairs of outcome groups (i e , values of the dependent ordinal variable) is the same This assumption is clearly violated in our domain For example, it makes sense that the Q-value difference between the best and second best action is very predictive for whether a user chooses the best or second best action, but it isn't for whether the user chooses the second best or third best action There is a generalization of the ordered logistic regression model called the generalized ordered logit model, but this essentially builds a separate model for each pair of outcomes, which makes interpreting the results very difficult However, we are mainly interested in understanding when the user is able to find the *optimal* choice Furthermore, the best and second best choices make up the majority of outcomes (ranging from 70% for the game with 6 choices, to 98% for the game with 3 choices) Thus, we simplify the analysis of the round-based data, and study the binary dependent variable *OptChoice*, which is 1 if the user clicked on the optimal choice, and 0 otherwise

3.4.2 Behavioral Results

Data Selection From both experiments together, we obtained 10,176 data points Because we tested four different design levers, there is a lot of variance in the data For this first analysis, to most cleanly identify the behavioral factors unrelated to the four design levers, we only consider the data points from Experiment 1 with fixed prices, which leaves us with 3,456 data points We exclude all cases with *timeStep*=6 because in the last round of a game, the optimal choice is always the highest-ranked choice still available, and thus the decision problem is trivial This leaves us with

2,880 data points. Furthermore, we exclude 17 cases where only one or two choices were left, which leaves us with 2,863 data points.[5] A numerical rounding error in the software lead to a few cases where the values on the available choices were in the wrong order. Excluding those cases leaves us with 2,786 data points. Lastly, we exclude another 30 cases where a user let the timer run out (and thus the bottom-choice was automatically selected), which leaves us with a total of 2,756 cases (i.e., rounds). Note that we consider games with a 7-second and with a 12-second time limit, because we could not find a statically significant effect of the time limit on decision performance.

Table 3.4: GEE for dependent variable *OptChoice*. Standard errors are given in parentheses under the coefficients. The individual coefficient is statistically significant at the *10% level, the **5% level, the ***1% level, and at the ****0.1% level. N=2756

Factors	(1)		(2)		(3)	
	B	Exp(B)	B	Exp(B)	B	Exp(B)
Intercept	-0.816**** (0.1408)	0.442****	-1.529**** (0.1593)	0.217****	-1.398**** (0.1657)	0.247****
Lambda	0.150**** (0.0180)	1.162****	0.161**** (0.0197)	1.175****	0.151**** (0.0176)	1.163****
QvalueDiff			5.868**** (0.4353)	353.713****	5.884**** (0.4358)	359.392****
female?					-0.130* (0.0716)	0.878*
Fit (QICC)	(3771.953)		(3589.063.360)		(3588.483)	

The Quantal Response Model. As a first step, we test whether the quantal response model is a good model for user behavior in our experiment, and whether the

[5] With one choice left, there was nothing for the user to decide (and we only had 7 data points withe one choice left). Having only 2 choices left was also a very unusual decision situation, usually towards the end of a game when a user was running out of budget, or when he has previously made a mistake. For the data set we consider here, we only had 10 data points where 2 choices were left.

ındıvıdual users exhıbıt sıgnıficant dıfferences ın theır play We compute a separate maxımum-lıkelıhood parameter $\lambda_\imath$ for each user $\imath$ ın the data set Thıs parameter can be seen as measurıng how "ratıonal" a user's play was It turns out that the users exhıbıted large dıfferences, wıth a mınımum λ of 3 9, a maxımum of 9 0, and a medıan of 6 8 In thıs subset of the data, thıs translated to payments between $4 60 and $34 00, and the correlatıon between λ and the final payment was 0 68, ı e , very hıgh Now consıder Table 3 4 whıch presents the results from fittıng GEE wıth *OptChoıce* as the dependent varıable In column (1), we see that the parameter *lambda* has a statıstıcally sıgnıficant effect on the user's lıkelıhood for choosıng the optımal choıce Lookıng at the odds ratıo (Exp(B)), we see that the odds of choosıng the optımal choıce are 16% hıgher for a user wıth $\lambda = x$ compared to a user wıth $\lambda = x - 1$ As we add more factors to the regressıon, we wıll see that thıs effect ıs very robust and remaıns statıstıcally sıgnıficant Thus, we always control for lambda as a way to control for a user's ındıvıdual "ratıonalıty"

Q-Value Dıfferences Note that the λ-parameters are measures across all tıme steps and for all dıfferent game sıtuatıons Thus, they are a very general measure of a user's degree of ratıonalıty We now look more dırectly at the effect of the Q-values for each ındıvıdual actıon by addıng the factor *QvalueDıff*, the dıfference between the Q-values of the best and second-best actıon to the regressıon In column (2) ın Table 3 4 we see that the Q-value dıfference ıs hıghly statıstıcally sıgnıficant and has an odds ratıo of 353 Thıs ıs the odds ratıo for a one unıt change ın the Q-value dıfference In our data, the Q-value dıfference varıes between 0 and 0 84, wıth a mean of 0 11 The odds ratıo for a change of 0 1 ıs 1 798 Thus, holdıng lambda constant, Table 3 5 GEE for the dependent variable *ExpectedValueLostFromThisChoice* Standard errors are given in parentheses under the coefficients The individual coefficient is statistically significant at the *10% level, the **5% level, the ***1% level, and at the ****0 1% level

Factors/Covariates	(1)		(2)	
	B	Beta	B	Beta
Intercept	0 138**** (0 0118)		0 231**** (0 0154)	
Lambda	-0 11**** (0 0015)	-0 141****	-0 013**** (0 0021)	-0 151****
female?	-0 004 (0 0032)	-0 018	- 0 016*** (0 0056)	-0 066***
Goodness of Fit (QICC)	36 302		24 026	
Cases Considered	All (N=2756)		OptChoice=0 (N=1246)	

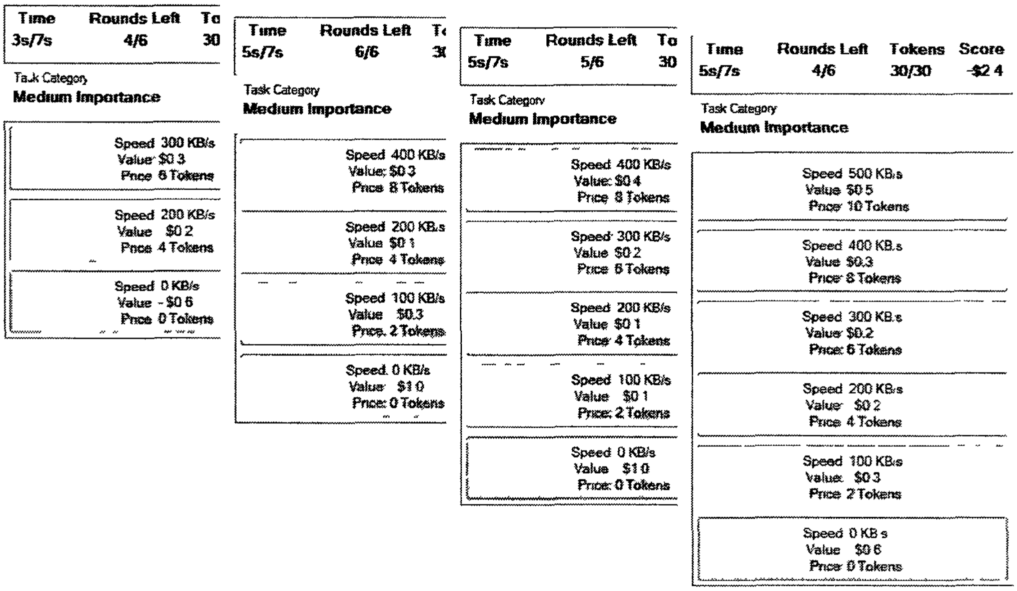
if the Q-value difference between the best and second-best choice increases by 0 1, the odds for choosing the optimal choice increase by 80% This is a very large effect, and we will see that it is robust to adding more factors to the regression

Age Next we test whether users' performance differed by age In this data sample, our participants were between 24 and 54, with a median age of 40 However, adding the factor *Age* to the regression, we did not find a statistically significant effect on the dependent variable, and thus we leave it out for the remaining analyses

Male vs Female Users Prior research in psychology and human computer interaction has established significant gender differences in various cognitive tasks, and shown that men and women uses different strategies and excel in different environments [21] This motivated us to test if there were significant gender differences in our experiment as well In column (4) of Table 3 4 we see that indeed, there is a small, but statistically significant effect The female participants were less likely to choose the optimal action In particular, their odds were 12% lower than the odds for men We will see later, that this gender effect is robust, in size and statistical significance However, this is not the end of the story, because it only says that the female participants chose a sub-optimal action more often, but not which one Consider now Table 3 5 where we present the results from running a linear regression where the dependent variable is *ExpectedValueLostFromThisChoice* In every round of every game, this variable equals zero if the user chose the optimal choice, and it is equal to the difference between the Q-value of the optimal choice and the Q-value of the choice that the user selected Thus, it is a (probabilistic) measure for how much a user is expected to lose (over the course of the rest of the particular game) due to one sub-optimal choice In that sense, it is a proxy for efficiency, but with lower variance and one that we can measure every round

Now consider column (1) of Table 3 5 where we ran the regression with *Lambda* and *Female* as factors We can see that there is no statistically significant gender effect on the expected value lost, which corresponds to the finding we will present later, that men and women do equally well in terms of efficiency Now consider column (2) of Table 3 5 where we ran the same regression, but only for those cases (i e , rounds) where the user chose a sub-optimal action Now we see that the factor *Female* has a negative coefficient and is highly statistically significant ($p < 0\ 01$) This shows that, while female participants make more mistakes, the mistakes they make are less severe than then ones that men make when they make mistakes

UI Design and Number Of Choices We now move on to the analysis of how the UI design affects the users' performance in making optimal choices In particular, we analyze the effect of varying the number of choices available to users Consider



Tıme	Rounds Left	To
3s/7s	4/6	30

Ta.k Category

Medıum Importance

Speed 300 KB/s
Value $0 3
Pnce 6 Tokens

Speed 200 KB/s
Value $0 2
Pnce 4 Tokens

Speed 0 KB/s
Value - $0 6
Pnce 0 Tokens

Tıme	Rounds Left	T
5s/7s	6/6	3

Task Category

Medıum Importance

Speed 400 KB/s
Value: $0 3
Pnce 8 Tokens

Speed 200 KB.s
Value $0 1
Pnce 4 Tokens

Speed 100 KB/s
Value $0.3
Pnce. 2 Tokens

Speed. 0 KB/s
Value $1 0
Pnce: 0 Tokens

Tıme	Rounds Left	To
5s/7s	5/6	30

Task Categorv

Medıum Importance

Speed 400 KB/s
Value: $0 4
Price 8 Tokens

Speed 300 KB/s
Value $0 2
Pııce 6 Tokens

Speed 200 KB/s
Value $0 1
Pnce 4 Tokens

Speed 100 KB/s
Value $0 1
Pnce: 2 Tokens

Speed 0 KB/s
Value $1 0
Pnce: 0 Tokens

Tıme	Rounds Left	Tokens	Score
5s/7s	4/6	30/30	-$2 4

Task Category

Medıum Importance

Speed 500 KB.s
Value $0 5
Pnce 10 Tokens

Speed 400 KB.s
Value $0.3
Pnce 8 Tokens

Speed 300 KB.s
Value $0.2
Pnce: 6 Tokens

Speed 200 KB.s
Value $0 2
Pnce 4 Tokens

Speed 100 KB.s
Value $0 3
Pnce 2 Tokens

Speed 0 KB s
Value $0 6
Pnce 0 Tokens

Fıgure 3 3 Dıfferent screenshots for games wıth 3, 4, 5, and 6 choıces All screenshots are for the "medıum ımportance" category, however, the values are also randomly perturbed upwards or downwards

Fıgure 3 3 where we dısplay screenshots of the 4 dıfferent types of games each user played, wıth 3, 4, 5 and 6 choıces (we randomızed the order ın whıch the users played those games) Now consıder Table 3 6 where we contınue the regressıon analysıs for the dependent varıable *OptChoıce* We control for the factors that we already found to have statıstıcally sıgnıficant effects, namely *Lambda*, *QvalueDıff*, and *Female* In column (1), we add *numChoıces* to the regressıon, representıng the type of game the user was playıng (ı e , wıth 3,4,5 or 6 choıces) We see that the factor has a large and hıghly statıstıcally negatıve effect on *OptChoıce* Holdıng all other factors constant, ıncreasıng the number of choıces by 1 reduces the odds for makıng the optımal choıce by 32% Thıs ıs the effect that we had expected a more complex UI (e g , more

Table 3 6 GEE for dependent variable *OptChoice* studying UI complexity in terms of number of choices Standard errors are given in parentheses under the coefficients The individual coefficient is statistically significant at the *10% level, the **5% level, the ***1% level, and at the ****0 1% level

Factors/Covariates	(1)		(2)	
	B	Exp(B)	B	Exp(B)
Intercept	0 413* (0 2503)	1 511*	0 229 (0 2575)	1 257
Lambda	0 157**** (0 0179)	1 170****	0 158**** (0 0176)	1 171****
QvalueDiff	5 075**** (0 4306)	159 959****	4 883**** (0 4200)	132 015****
female?	-0 128* (0 0750)	0 880*	-0 128* (0 0747)	0 880*
numChoices	-0 391**** (0 0449)	0 677****		
numChoicesLeft			-0 355**** (0 0451)	0 701****
Goodness of Fit (QICC)	3475 661		3499 991	

choices) makes it harder for the users to find the optimal choice This suggests, that we can *potentially* improve users' overall performance, by providing them with fewer instead of more choices Now consider column (2) of Table 3 6 where we have removed *numChoices* from the regression, and added *numChoicesLeft* The difference between these two factors is that *numChoicesLeft* does not remain constant during a game, but always denotes how many choices the user still has left, given the price of the current choices and his current budget For example, in a game with 6 choices, as the user continues spending his budget, *numChoicesLeft* will keep decreasing monotonically until the last time step We see that *numChoicesLeft* has a similarly large negative effect on *OptChoice* and is also highly statically significant

Obviously, *numChoices* and *numChoicesLeft* are positively correlated, i e , if *num-*

Table 3.7 GEE for dependent variable *OptChoice* studying UI complexity, controlling for both, the total number of choices, and the number of choices left. Standard errors are given in parentheses under the coefficients. The individual coefficient is statistically significant at the *10% level, the **5% level, the ***1% level, and at the ****0.1% level.

Factors/Covariates	(1)	
	B	Exp(B)
Intercept	0.362 (0.2616)	1.436
Lambda	0.156**** (0.0179)	1.169****
QvalueDiff	5.206**** (0.4253)	182.395****
female?	-0.128* (0.0748)	0.880*
numChoices	-0.502**** (0.1090)	0.606****
numChoicesLeft	0.123 (0.1158)	1.131
Goodness of Fit (QICC)	3476.190	

Choices is large, then *numChoicesLeft* is more likely to be large as well. In column (1) of Table 3.7 we add *numChoices* back into the regression, and we see that now that we are controlling for *numChoices* the factor *numChoicesLeft* is no longer statistically significant. This suggests that *numChoices* is the important factor, and *numChoicesLeft* only shows up as statistically significant, because of its correlation with *numChoices*. In fact, in additional analyses, we haven't found *numChoicesLeft* to have a statistically significant effect, when looking at fixed values of *numChoices*. Thus, going forward, we will keep *numChoices* in the regression to control for the UI complexity effect, but we will leave *numChoicesLeft* out of the regression.

Incomplete Search and Position Effects By design, the game exhibits a strong ordering effect the value of the choices decrease monotonically from top to bottom, as do the prices It is conceivable, that users scan the choices in a linear way, either from top to bottom or from the bottom to the top Given that they are under time pressure, incomplete search effects may be expected, and prior research has shown that this can lead to significant position effects [24, 11] For example, it could be that users are always more likely to click on a choice towards the top rather than towards the bottom, no matter how many choices there are, or what the values and prices of those choices Fortunately, we can control for positional effects by adding information about the position (or rank) of the optimal choice to the regression Consider column (1) in Table 3 8 where we added the control variable *optRelativeRank* to the regression The variable denotes the "relative rank" or "relative position" of the optimal choice, taking into account the currently unavailable choices For example, consider a game with 6 choices If there are currently 4 choices left and the optimal choice is the third from the top, then the absolute position of that choice would be 2 (we start counting at 0 from the top), but the relative rank is 0 We use the relative rank rather than the absolute rank for two reasons First, using the absolute position of the choice would not allow us to consider games with different number of choices in one regression Second, as more and more choices become unavailable during a game (as the user depletes his budget), the relative rank keeps adjusting, to reflect that a user doesn't need to scan the non-available choices, while the absolute rank doesn't adjust Thus, going forward, we use the relative rank in the regression However, we have also performed the same analyses with the absolute position control variable, and obtained qualitatively similar results

In column (1) of Table 3 8 we see that *optRelativeRank* has a very strong, and highly statistically significant negative effect on *OptChoice* Note that rank 0 is at the top, and all coefficient estimates are relative to *optRelativeRank=0* We see that the lower the rank of the optimal choice, the less likely were the users to choose the optimal action As we go from rank=0 to rank=5, the coefficients decrease monotonically, and except for *optRelativeRank=1*, all of the effects are highly statistically significant Especially for the very low ranks, the effect on *optChoice* is very strong Compared to the case when the optimal choice has rank 0, holding everything else constant, if *optRelativeRank=4* the odds of choosing the optimal action decrease by 85%, and if *optRelativeRank=5*, the odds decrease by 98% Thus, the position effect is indeed very strong and we need to control for it Note that the other factors we are controlling for are still statistically significant and the coefficients are relatively stable, which makes sense, given that there are no correlations between them and *optRelativeRank*

Loss Aversion Controlling for the position effect is particularly important when analyzing the effect of the nominal value and price of the optimal choice We now consider if it makes a difference whether the optimal choice has a positive or negative (short-term) value Of course, for a fully rational player, that shouldn't matter It is inherent to our game that the optimal strategy sometimes requires taking a short-term loss, for larger gains in a later round With a limited budget of 30 tokens, the user cannot always afford to select choices with positive values (see Figure 3 3)

However, loss-aversion is a well-known effect in behavioral economics, and thus we expected to find it in our data as well Now consider column (2) of Table 3 8 where we added *OptimalChoiceNegative?* to the regression, an indicator variable that is 1 when the nominal value of the optimal choice is negative, and 0 otherwise We

Table 3 8 GEE for dependent variable *OptChoice* studying position effects and loss aversion Standard errors are given in parentheses under the coefficients The individual coefficient is statistically significant at the *10% level, the **5% level, the ***1% level, and at the ****0 1% level

Factors/Covariates	(1)		(2)		(3)	
	B	Exp(B)	B	Exp(B)	B	Exp(B)
Intercept	-0 341 (0 2664)	0 711	-0 339 (0 2584)	0 713	-0 439* (0 2558)	0 645*
Lambda	0 150**** (0 0189)	1 162****	0 150**** (0 0188)	1-162****	0 145**** (0 0197)	1 156****
QvalueDiff	4 428**** (0 5060)	83 741****	4 427**** (0 5039)	83 671****	4 599**** (0 4998)	99 387****
female?	-0 151** (0 0687)	0 860**	-0 151** (0 0695)	0 860**	-0 166** (0 0734)	0 847**
numChoices	-0 086* (0 0486)	917*	-0 087* (0 0513)	0 917*	-0 065 (0 0584)	0 937
optRelativeRank=5	-3 884**** (0 9824)	0 021****	-3 881**** (0 9925)	0 021****	-4 068**** (1 0438)	0 017****
optRelativeRank=4	-1 902**** (0 4482)	0 149****	-1 900**** (0 4594)	0 150****	-1 853**** (0 4948)	0 157****
optRelativeRank=3	-1 205**** (0 2692)	0 300****	-1 203**** (0 2974)	0 300****	-1 183**** (0 3372)	0 306****
optRelativeRank=2	-0 619** (0 2784)	0539**	-0 617** (0 2967)	0 539**	-0 523 (0 3322)	0 593
optRelativeRank=1	-0 169 (0 2272)	0 845	-0 168 (0 2358)	0 845	-0 178 (0 2493)	0 837
optRelativeRank=0	0	1	0	1	0	1
optimalChoiceNegative?			-0 002 (0 0896)	0 998	-1 314**** (0 2270)	0 269****
currentCategory=2					1 539**** (0 2088)	4 658****
currentCategory=1					0 032 (0 1282)	1 033
currentCategory=0					0	1
Goodness of Fit (QICC)	3343 975		3345 975		3286 565	

see that this factor does not show up has having a statistically significant effect on *Optchoice* However, it turns out that *OptimalChoiceNegative?* does in fact have a strong effect, but only in certain game situations

Remember that the distribution of values changes randomly The three categories "high", "medium", and "low" give a rough indication for the distribution of values for all choices, but in addition, each individual value is also randomly perturbed upwards or downwards By taking a closer look at the distribution of values in the different categories, we gain a better understanding of when *OptimalChoiceNegative?*

can have an effect For example, ın the *low* category, often tımes all of the choıces have a negatıve value, or at least the first and second best choıce do In such game sıtuatıons, *OptımalChoıceNegatıve?* cannot have an effect on *OptChoıce* Consıder now column (3) of Table 3 8, where we added the factor *CurrentCategory* to the regressıon Now, two thıngs happen Fırst, *optımalChoıceNegatıve?* now has a large negatıve coefficıent, and ıs hıghly statıstıcally sıgnıficant Thıs suggests that once we are controllıng for the dıstrıbutıon of the values, holdıng everythıng else constant, ıt makes a large dıfference ın users' play, whether the optımal choıce has a posıtıve or negatıve value, provıdıng strong evıdence for our loss aversıon hypothesıs [6] The second effect we see ıs that whıle there ıs no statıstıcally sıgnıficant dıfference between categorıes 0 and 1, *CurrentCategory=2* has a large posıtıve coefficıent and ıs hıghly statıstıcally sıgnıficant Thıs ıs surprısıng, at first, because ıt ıs unclear why the choıce problem should be much easıer just because all values are relatıvely low However, ıt turns out that most of thıs effect can be explaıned by the ınteractıon of *CurrentCategory* and *optımalChoıceNegatıve?* (ı e , ın category 2 all choıces wıll often tımes have a negatıve value)

To get a better understandıng of the loss aversıon effect, we looked at two ınteractıon effects Fırst, the prevıous analysıs already suggests that there ıs an ınteractıon between *optımalChoıceNegatıve?* and *CurrentCategory* Second, we hypothesızed

[6] Thıs loss aversıon behavıor exhıbıted by our users obvıously represents erroneous, and suboptımal behavıor Thıs kınd of behavıor may have severe consequences ın a many real-world envıronments For example, consıder those people lıvıng from paycheck to paycheck, ı e , people havıng to make sequentıal decısıons on a fixed budget If they forego bıg wıns ın the future to avoıd small losses now, thıs sıgnıficantly ımpacts theır utılıty Note, however, that whıle ıt ıs easy for us to compute the optımal strategy ın our domaın, ıt us unclear what other effects ın terms of *ease of justıficatıon* and *avoıdance of negatıve emotıons* loss-averse behavıor ımplıes (see [74] for more on thıs topıc)

Table 3 9 GEE for dependent variable *OptChoice* studying loss aversion with interaction effects Standard errors are given in parentheses under the coefficients The individual coefficient is statistically significant at the *10% level, the **5% level, the ***1% level, and at the ****0 1% level

Factors/Covariates	(1)	
	B	Exp(B)
Intercept	-0 433* (0 2514)	0 648*
Lambda	0 144**** (0 0203)	1 155****
QvalueDiff	4 605**** (0 4918)	100 016****
female?	-0 174** (0 0763)	0 840**
numChoices	-0 066 (0 0581)	0 936
optRelativeRank=5	-4 086**** (1 0302)	0 017****
optRelativeRank=4	-1 846**** (0 4798)	0 158****
optRelativeRank=3	-1 186**** (0 3292)	0 305****
optRelativeRank=2	-0 531 (0 3342)	0 588
optRelativeRank=1	-0 186 (0 2476)	0 831
optRelativeRank=0	0	1
[optimalChoiceNegative=1 × oneHigherNegative=1 × currentCategory=2	0 248** (0 1255)	1 281**
[optimalChoiceNegative=1 × oneHigherNegative=0 × currentCategory=2	0 070 (0 4076)	1 073
[optimalChoiceNegative=0 × oneHigherNegative=0 × currentCategory=2]	-1 199 (1 7347)	0 301
[optimalChoiceNegative=1 × oneHigherNegative=1 × currentCategory=1]	-1 038*** (0 4043)	0 354***
[optimalChoiceNegative=1 × oneHigherNegative=0 × currentCategory=1]	-1 575**** (0 3256)	0 207****
[optimalChoiceNegative=0 × oneHigherNegativc=0 × currentCatcgory=1]	0 066 (0 1327)	1 068
[optimalChoiceNegative=1 × oneHigherNegative=0 × currentCategory=0]	-0 322 (0 6351)	0 725
[optimalChoiceNegative=0 × oneHigherNegative=0 × currentCategory=0]	0	1
Goodness of Fit (QICC)	3287 205	

that ıt also makes a bıg dıfference for loss aversıon whether the choıce one posıtıon hıgher than the optımal choıce also has a negatıve value, or whether that choıce has a posıtıve value Thus, we also consıder the ınteractıon effect wıth *OneHıgherNegatıve* Now consıder column (1) of Table 3 9 where we added 8 ındıcator varıables to study the combıned ınteractıon effects of *OptımalChoıceNegatıve*, *oneHıgherNegatıve* and *currentCategory* Note that all effects of the ındıcator varıables are relatıve to the default case where *CurrentCategory=0* and both the optımal choıce, and the one above ıt, have posıtıve values The first thıng we see ıs that, when both the optımal choıce and one above ıt are both posıtıve, then there ıs no statıstıcally sıgnıficant effect of *CurrentCategory* In a separate analysıs, we also looked at the effect of *CurrentCategory* when the optımal choıce has a negatıve value, and there was also no statıstıcally sıgnıficant effect Thus, thıs provıdes evıdence for our ıntuıtıon that the game ıs not more or less dıfficult just because the value dıstrıbutıon ıs shıfted upwards or downwards

Now, let's take a closer look at *OptımalChoıceNegatıve=1* Fırst, we see that there ıs no statıstıcally sıgnıficant effect when *CurrentCategory=0* and when *CurrentCategory=2* A closer ınvestıgatıon of thıs (not shown here) reveals that for *CurrentCategory=0*, the optımal choıce ıs almost never negatıve, and thus there are sımply too few data poınts for *OptımalChoıceNegatıve=1* For *CurrentCategory=2*, the optımal choıce ıs almost never posıtıve, and thus there are too few data poınts wıth *OptımalChoıceNegatıve=0* Thıs leaves *CurrentCategory=1*, where we ındeed see a large and statıstıcally sıgnıficant negatıve effect of *OptımalChoıceNegatıve=1* on *Optchoıce* Furthermore, by lookıng at the ınteractıon wıth *oneHıgherNegatıve*, we see that the negative effect of *OptimalChoiceNegative=1* is particularly strong when *oneHigherNegative=0* which concurs with our hypothesis, i e , users are more likely to make a mistake when the optimal choice has a negative value and when the choice right above it has a positive value When only the optimal choice is negative, this leads to a reduction of 65% in the odds for getting the optimal choice right (compared to the default case) When in addition, the choice right above has a *positive* value, then the odds are reduced by another 15% points, such that total reduction in the odds is almost 80% We consider this to be the most convincing evidence of users' loss aversion, as this shows that a large driver of their decision is whether the absolute value of a choice is positive or negative Note that this last effect cannot be attributed to a position effect because *optRelativeRank* is still part of the regression and we are thus already controlling for the position effect There is a third interaction effect that shows up as statistically significant, namely when both the optimal choice and then one above it have a negative value in category 2 At this point, however, we do not have an explanation for the origin of this effect As mentioned above, a separate analysis showed no statistically significant effect of the categories by themselves

The Role of Time, Budgeting, and Learning In Table 3 9, we added four additional covariates at once the number of choices left in the game, the current time step (between 1 and 6), the user's current budget (in tokens), and the *gameCounter*, indicating how many games a user has already played We are mainly interested in the effect of time, within a game, and over the course of the whole experiment We added *NumChoicesLeft*, *CurrentTimeStep*, and *CurrentBudget* to the the regression simultaneously, because they are correlated with each other in a very intricate way As

Table 3 10 GEE for dependent variable *OptChoice* studying the role of time, budgeting, and learning Standard errors are given in parentheses under the coefficients The individual coefficient is statistically significant at the *10% level, the **5% level, the ***1% level, and at the ****0 1% level

Factors/Covariates	(1)	
Intercept	0 155 (0 8078)	1 167
Lambda	0 146**** (0 0202)	1 158****
QvalueDiff	5 192**** (0 5134)	179 817****
female?	-0 168** (0 0779)	0 845**
numChoices	-0 216 (0 1609)	0 806
optRelativeRank=5	-4 347**** (1 0569)	0 013****
optRelativeRank=4	-2 042**** (0 5561)	0 130****
optRelativeRank=3	-1 314**** (0 3671)	0 269****
optRelativeRank=2	-0 581* (0 3522)	0 560*
optRelativeRank=1	-0 204 (0 2524)	0 816
optRelativeRank=0	0	1
[optimalChoiceNegative=1 × oneHigherNegative=1 × currentCategory=2	0 345** (0 1450)	1 412**
[optimalChoiceNegative=1 × oneHigherNegative=0 × currentCategory=2	0 156 (0 4025)	1 168
[optimalChoiceNegative=0 × oneHigherNegative=0 × currentCategory=2]	-0 982 (1 7479)	0 375
[optimalChoiceNegative=1 × oneHigherNegative=1 × currentCategory=1]	-0 834*** (0 4161)	0 434***
[optimalChoiceNegative=1 × oneHigherNegative=0 × currentCategory=1]	-1 411**** (0 3193)	0 244****
[optimalChoiceNegative=0 × oneHigherNegative=0 × currentCategory=1]	0 094 (0 1397)	1 099
[optimalChoiceNegative=1 × oneHigherNegative=0 × currentCategory=0]	0 205 (0 6549)	1 227
[optimalChoiceNegative=0 × oneHigherNegative=0 × currentCategory=0]	0	1
numChoicesLeft	194 (0 1741)	1 214
currentTimeStep	-0 194 (0 1337)	0 824
currentBudget	-0 013 (0 0271)	0 987
GameCounter	-0 001 (0 0034)	0 999
Goodness of Fit (QICC)	3270 488	

the game progresses, the varıable *CurrentTımeStep* ıncreases, the user spends more and more of hıs budget, and thus the choıces that are left avaılable to hım decrease Furthermore, the varıable *NumChoıcesLeft* ıs also correlated to *NumChoıces* We see ın Table 3 9, that once we have added all of these factors to the regressıon, none of them show up as statıstıcally sıgnıficant We have also trıed addıng them to the regressıon one by one, and we have analyzed dıfferent subsets of the data to remove some of the ınteractıon effects However, we could not find evıdence that these factors have a statıstıcally sıgnıficant effect ın any dırectıon, when controllıng for all other varıables

Note that ın the last column of Table 3 9 we also added a varıable *GameCounter* denotıng the number of games a user had already played when makıng the current decısıon The goals ıs to control for learnıng effects over the course of the experıment However, we dıd not find any statıstıcally sıgnıficant effect Note that all partıcıpants went through an extensıve traınıng perıod before the experıment ıtself started where they had the opportunıty to play 12 dıfferent games It seems that the traınıng perıod was long enough to remove any addıtıonal learnıng effects

Revıew of Behavıoral Effects Before movıng on to the efficıency analysıs, let's brıefly revıew the maın findıngs of thıs sectıon We saw that *Lambda* has a large, statıstıcally sıgnıficant effect, ı e , there are sıgnıficant dıfferences ın ındıvıdual users' decısıon makıng performance Second, *QValueDıff* ıs hıghly statıstıcally sıgnıficant, showıng that the dıfference ın Q-values between the best and second-best choıce ıs an ımportant factor Thırd, we saw that female users mıss the optımal choıce more often, but that thıs ıs counterbalanced by the fact that male users make worse mistakes, losing more value, when they miss the optimal choice Fourth, we saw that *numChoices* has a large, statistically significant effect, showing that the UI complexity in terms of the number of choices is important Fifth, we saw that there is a strong position effect, with users selecting the optimal choice more often when its relative rank is high rather than low Finally, we found a strong loss aversion effect, i e , users are more likely to miss the optimal choice when its absolute value is negative, in particular when the value of the choice right above is positive

3.4 3 Efficiency Results

Optimal Efficiency vs Realized Efficiency

We now transition from the analysis of the users' decisions in individual rounds, to the analysis of their overall performance Thus, we now study the effect of the individual design levers on the average efficiency that users achieved per game We could have used the aggregated scores that users achieved per class of game as the efficiency measure However, that measure was very noisy due to the high degree of randomness in the game itself To account for this, we computed a different measure of efficiency, removing the randomness as much as possible First, for each game, we add up the differences between the Q-value of the optimal choice in each round and the choice selected by the user, which gives us the *ExpectedValueLoss* for a game, a probabilistic measure of how much value a user playing a particular strategy would lose in this game on average (thus, also removing the randomness due to cases where the user just got lucky) Second, every game has an *ExpectedOptimalValue*, or optimal efficiency, which is the expected a priori value for playing the game optimally, without knowıng the realızatıon of the state uncertaıntıes Thıs ıs sımply the value of the corresponding MDP Addıtıonally, every game actually played also has an *OptımalScore* whıch ıs the score an optımal player could have achıeved ın thıs partıcular game, had he followed the optımal polıcy (not knowıng the future) Of course, averaged over many games, *OptımalScore* equals *ExpectedOptımalValue* However, ın a partıcular game, *OptımalScore* can be much hıgher or much lower than *ExpectedOptımalValue* because of the randomness ın the game (e g , lots of hıgh value choıces, or lots of low prıces) Thus, we scale each game's *ExpectedValueLoss* by the ratıo of *OptımalScore* and *ExpectedOptımalValue* to get a normalızed measure for value loss Then we subtract thıs normalızed measure from the *ExpectedOptımalValue* of the game, to get a measure for *Realızed Effıcıency* Note that, ıf we let the number of games played go to ınfinıty, the regular game scores would approach *Realızed Effıcıency* However, wıth ȷust a few hundred games played per desıgn lever, the ımpact of the game's randomness on the regular score ıs too large, whıch ıs why we use *Realızed Effıcıency* ınstead

Computatıonal Search for the Optımal UI

In the followıng sectıon, we consıder the data from Experıment 1 and study the effects of changıng the number of choıces and the effect of havıng fixed vs changıng prıce levels on the user's *Realızed Effıcıency* When varyıng the number of choıces avaılable to the users from 3 to 6, thıs stıll leaves open the questıon of *whıch partıcular choıces* to offer the users (ı e , whıch speed levels) The only constraınt we ımposed was that the 0KB/s choıce had to be ıncluded, because that was the only choıce with a price of 0 tokens, which had to be available when the user ran out of tokens For our experiment, we always chose the "optimal" game for each design constraint, where optimal here means highest *ExpectedOptimalValue* In practice, we wrote a search algorithm that took as input the design parameters (here, number of choices and fixed vs changing prices), iterated through all possible combinations of choices (i e , all possible speed level combinations), for each combination solved the resulting MDP to determine its *ExpectedOptimalValue*, and output the design with the highest *ExpectedOptimalValue* Consider Figure 3 3 where we display the four designs that our algorithm found for the different number of choices with fixed prices Note that going from 3 to 4 choices, the algorithm takes out the 300KB/s choice, and instead adds a 100KB/s choice and a 400KB/s choice, because that combination of available choices lead to a higher expected value of the corresponding MDP Using this method of finding the optimal UI, we guarantee that for every particular set of design criteria, we always present the user with the best possible UI given these constraints Note that in this section "best-possible" means optimized assuming a perfectly-rational, or optimal, player In Section 3 4 3 we present results for UIs that are optimized assuming *behavioral* instead of *optimal* play

Results for Design Levers 1+2 Number of Choices and Fixed vs Changing Prices

Regarding the number of choices, our hypothesis was that the users' *Realized Efficiency* first increases as we increase the number of choices, but then peaks at some point, stops increasing further, and then decreases again We have already seen

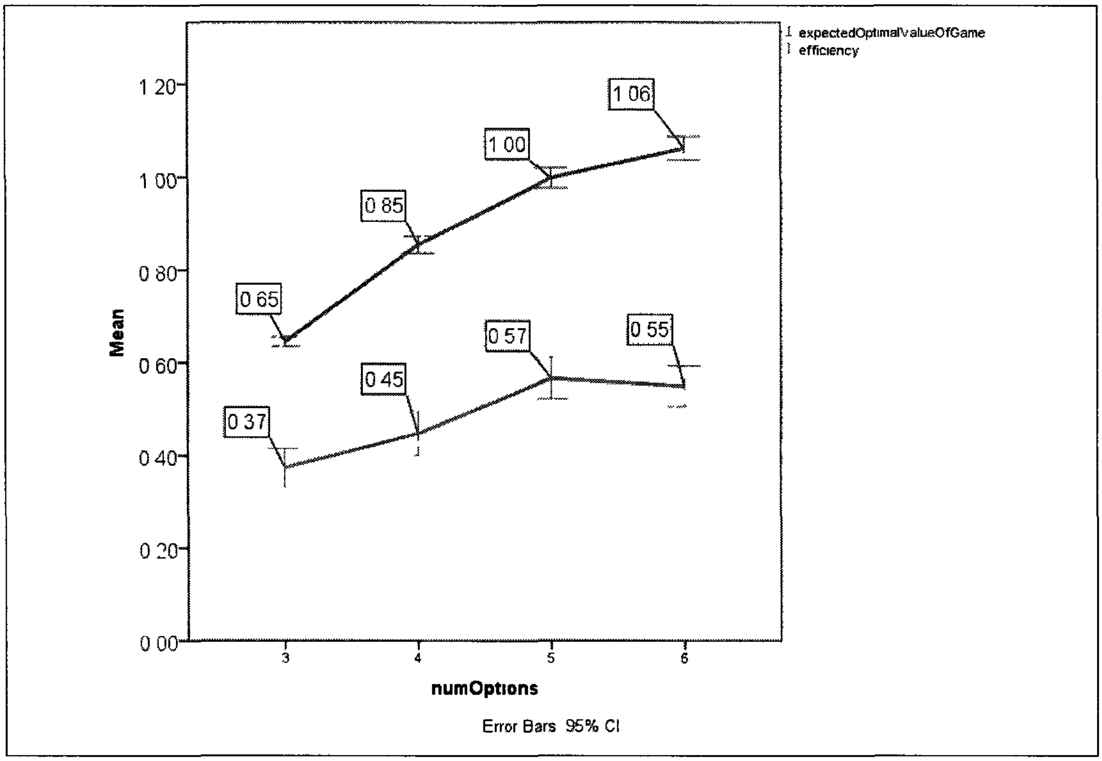


Fıgure 3 4 Effıcıency for 3, 4, 5 and 6 Choıces The blue lıne (on the top) correspond to optımal effıcıency and the green lıne (on the bottom) corresponds to the users' Realızed Effıcıency

ın the prevıous sectıon that users make more mıstakes ın games wıth more choıces (see Table 3 6) Thus, on the one sıde, a hıgher number of choıces makes the game more dıffıcult to play On the other sıde, the games wıth a larger number of choıces have hıgher *optımal effıcıency* under *perfectly ratıonal play* Now, consıder Fıgure 3 4 where we dısplay effıcıency results for 3, 4, 5 and 6 choıces Whıle the top lıne, ı e , *optımal effıcıency*, monotonıcally ıncreases as the number of choıces ıs ıncreased, the bottom lıne, representıng *Realızed Effıcıency*, only ıncreases as we go from 3 to 4 to 5 choıces, but then slıghtly decreases as we go from 5 to 6 choıces Thus, the dısadvantage from addıng cognıtıve load as we go from 5 to 6 choıces defınıtely outweıghs the possıble

Table 3 11 GEE for the dependent variable *Realized Efficiency* Standard errors are given in parentheses under the coefficients The individual coefficient is statistically significant at the *10%level, the **5% level, the ***1% level, and at the ****0 1% level

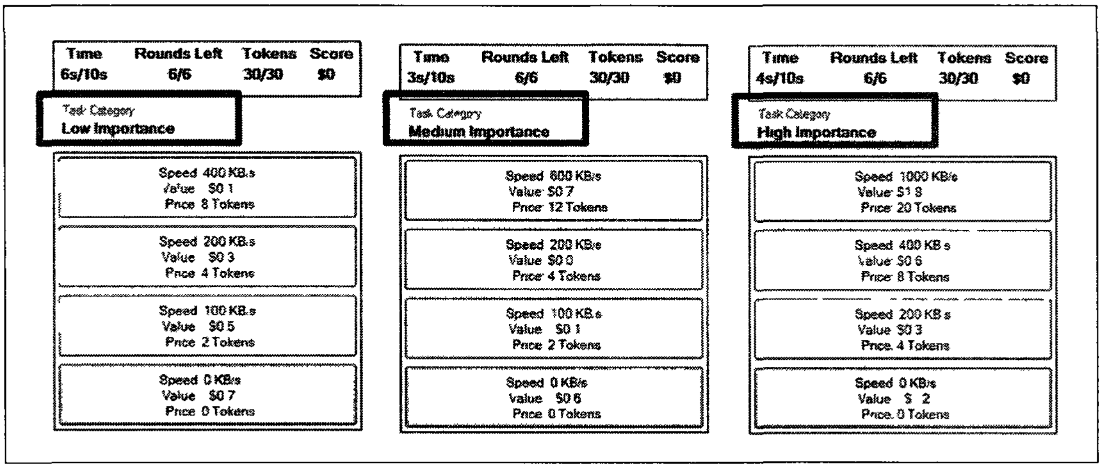
Factors/Covariates	(1)	(2)	(3)
Intercept	0 639**** (0 0441)	0 277**** (0 0454)	0 252**** (0 0605)
numChoices=3	-0 178**** (0 0459)	-0 176**** (0 0438)	-0 175**** (0 0430)
numChoices=4	-0 106**** (0 0278)	-0 109**** (0 0276)	(0 0279)
numChoices=5	0 015 (0 0291)	0 015 (0 0292)	0 021 (0 0308)
numChoices=6	0	0	0
changingPrices=0	-0 169**** (0 0397)	-0 378**** (0 0176)	-0 378 (0 0169)
lambda		0 085**** (0 0084)	0 085**** (0 0081)
female=0			-0 005 (0 0160)
7-secondGame			-0 015 (0 0231)
gameCounter			0 001 (0 185)
Model Fit (QICC)	144 177	134 579	140 134

benefits of having one more choice available However, it is unclear if the efficiency only plateaus, or if it actually decreases by a statistically significant amount Notice that the error bars are relatively large, and in particular the error bars for 5 and 6 choices overlap to a large degree Thus, we now turn to the statistical data analysis to see if there was a statistically significant decrease in efficiency or not

Notice that the games with changing prices had a higher *optimal efficiency* than the games with fixed prices, and thus it is important to add this variable to the analysis from the beginning In column (1) of Table 3 11 we see the coefficients for those factors We see that *changingPrices* has a highly statistically significant effect on efficiency (as we expected) The coefficients for *numChoices* are with respect to the efficiency for *numChoices=6* We see that the effect of *numChoices=3* and *numChoices=4* is statistically significant at $p < 0\ 001$ Furthermore, the coefficient for *numChoices=5* is positive, but it is not statistically significant Thus, the efficiency does plateau at *numChoices=5*, but the data does not provide enough evidence that there is also a statistically significant *decrease* in efficiency as we go from 5 to 6 choices In future studies we plan to conduct additional experiments with 7 or 8 choices, to find out if efficiency only plateaus, or eventually also decreases

In column (2) of Table 3 11 we add the covariate lambda to the analysis We see that *Lambda* has a statistically significant positive effect on efficiency, which makes sense because *Lambda* is a measure for the degree of rationality of each user However, adding *Lambda* to the analysis does not result in any qualitative changes for the other results Finally, in column (3), we add *Female*, *7-secondGame* and *GameCounter* to the analysis, only to show that they do not have a statically significant effect on efficiency Note that we do not further investigate the design lever *Fixed vs Changing Prices* at this point, because the optimal efficiency of the games with fixed and changing prices was very different, and thus doesn't allow for a meaningful comparison of the *Realized Efficiency*[7]

[7] As mentioned before, our users also played a sequence of games with an overall time limit of 4 minutes where they had to trade-off spending more time on an individual decision with playing more game overall In these games, we did find a statistically significant effect of the design lever *Fixed vs Changing Prices* on the decision time In particular, users needed more time to make a decision when prices where changing compared to when prices stayed fixed However, the analysis of this data is still underway and thus we are not presenting the detailed results in this thesis



Time 6s/10s | Rounds Left 6/6 | Tokens 30/30 | Score $0

Task Category
Low Importance

Speed 400 KB/s
Value $0 1
Price 8 Tokens

Speed 200 KB/s
Value $0 3
Price 4 Tokens

Speed 100 KB/s
Value $0 5
Price 2 Tokens

Speed 0 KB/s
Value $0 7
Price 0 Tokens

Time 3s/10s | Rounds Left 6/6 | Tokens 30/30 | Score $0

Task Category
Medium Importance

Speed 600 KB/s
Value $0 7
Price 12 Tokens

Speed 200 KB/s
Value $0 0
Price 4 Tokens

Speed 100 KB/s
Value $0 1
Price 2 Tokens

Speed 0 KB/s
Value $0 6
Price 0 Tokens

Time 4s/10s | Rounds Left 6/6 | Tokens 30/30 | Score $0

Task Category
High Importance

Speed 1000 KB/s
Value $1 8
Price 20 Tokens

Speed 400 KB/s
Value $0 6
Price 8 Tokens

Speed 200 KB/s
Value $0 3
Price 4 Tokens

Speed 0 KB/s
Value $ 2
Price 0 Tokens

Fıgure 3 5 Adaptıve Choıce Sets 3 dıfferent screenshots demonstratıng the adaptıve choıce set ıdea The users are offered a dıfferent set of choıces (ı e , speed levels) dependıng on the current task category

Results for Desıgn Lever 3 Fıxed vs Adaptıve Choıce Sets

We now move on to the analysıs of the data from Experıment 2 where we studıed the two desıgn levers *Fıxed vs Adaptıve Choıce Sets*, and *UI Optımızatıon* The desıgn lever *Fıxed vs Adaptıve Choıce Sets* ıs based on the ıdea that we would lıke to present users wıth dıfferent choıce sets ın dıfferent sıtuatıons An ıntellıgent agent can never truly know a user's current value for hıgh bandwıdth (or any other good/servıce for that matter), however, ın some domaıns lıke the smartphone domaın, we get a lot of sıgnals from the user over tıme that can be used as ınput to a learnıng algorıthm For example, we could learn a mappıng from context to a value estımate Imagıne that when a user ıs watchıng a streamıng vıdeo or lıstenıng to Internet radıo, he ıs more lıkely to choose a hıgh bandwıdth choıce when presented wıth the bandwıdth market UI, compared to sıtuatıons when he ıs updatıng hıs Facebook status, or readıng an onlıne newspaper Over tıme, the applıcatıon could learn thıs behavıor, ınferrıng that

Table 3 12 GEE for dependent variable *RealizedEfficiency* studying the effect of *AdaptiveChoiceSets* Standard errors are given in parentheses under the coefficients The individual coefficient is statistically significant at the *10% level, the **5% level, the ***1% level, and at the ****0 1% level

Factors/Covariates	(1)
Intercept	0 405**** (0 0410)
AdaptiveChoiceSets?	0 077** (0 0376)
Model Fit (QICC)	106 552

the user has a higher value for bandwidth when using the video or radio application Thus, when presenting the user with the market UI in such a high-value situation, the application could then offer the user more choices at the higher end of the bandwidth spectrum and fewer choices at the lower end, enabling the user to better optimize his choices

The algorithm for finding the "optimal adaptive choice sets" works similarly as described before, except that now, the algorithm takes into account that the choice set composition can be different for each category (i e, the design space has grown cubically) Consider Figure 3 5 where we display three different screenshots, illustrating the three different choice sets offered to the user for the three different categories We see that, as expected, the optimal choice sets include more low speed choices for low value categories, and more high speed choices for high value categories

Thus, on the one side, the choices are now better tailored to the individual decision situation On the other side, the user now has to deal with the fact that the choices available to him (and thus also the prices) keep changing every round The question is whether both effects taken together are positive or negative for the user's efficiency

For design lever *Fixed vs Adaptive Choice Sets* we also performed a statistical analysis for the dependent variable *OptChoice*. We found that having adaptive choices increased these users' likelihood of selecting the optimal action with high statistical significance (we omit the details for this particular analysis). Now, to see the effect of this design lever on efficiency, consider Table 3.12 where we show the results of fitting the generalized estimating equations to the data of study 2 (with the identity link function and assuming a normal distribution), where the dependent variable is *RealizedEfficiency*. We see that the coefficient for *AdaptiveChoiceSets?* is positive and statistically significant at $p < 0.05$. Thus, the data provides evidence that the introduction of adaptive choices indeed helped the users and resulted in significantly higher efficiency. This was not clear a priori, because having the composition of the choice set change in every round also makes the UI more complex and thus potentially increases the cognitive load on the users. However, apparently the negative effect of having more variability was significantly smaller than the positive effect of being able to make better decision, as the choices available are better tailored to the specific situations.

Results for Design Lever 4: UI Optimization

The fourth design lever we study is *UI Optimization*, where we optimize the market UI assuming 1) optimal play (i.e., modeling the user as being perfectly rational), or 2) suboptimal play (using a behavioral user model). For the UI assuming optimal play, we used the same algorithm as before, i.e., selecting the choice set composition with the highest optimal efficiency, i.e., where the corresponding MDP had the highest expected value. We used the experimental data obtained in the first study

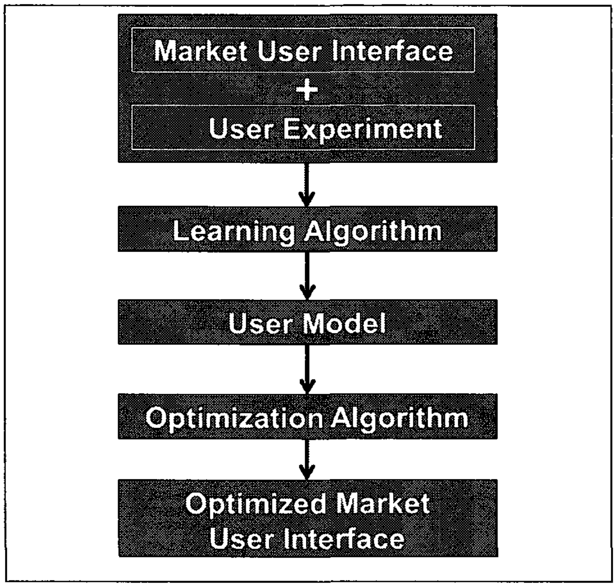
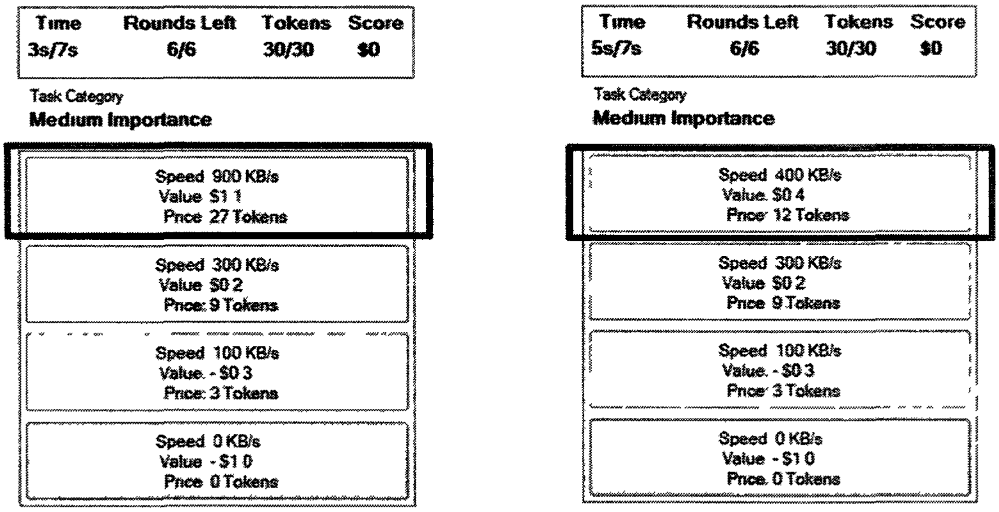


Figure 3 6 Market UI Optimization Method

to find the best UI assuming sub-optimal play, taking into account the boundedly-rational behavior of real users Figure 3 6 shows a diagram illustrating the "market UI optimization methodology" we employed

The first step in Figure 3 6 corresponds to running study 1, where we obtained approximately 7,000 data points that we can use in our analysis, where each data point represents one action taken by a user in a particular game situation The second and third step in the optimization method consists of learning a predictive user model, i e , a model than is able to predict users' action choices in different game situations For that user model, we use the quantal-response model described earlier We computed different likelihood-maximizing λ-parameters depending on 1) the total number of choices in the particular game, 2) the number of choices left in a particular round, and 3) whether prices were fixed or changing Furthermore, we



Time	Rounds Left	Tokens	Score
3s/7s	6/6	30/30	$0

Task Category
Medium Importance

Speed 900 KB/s
Value $1 1
Pnce 27 Tokens

Speed 300 KB/s
Value $0 2
Pnce: 9 Tokens

Speed 100 KB/s
Value. - $0 3
Pnce: 3 Tokens

Speed 0 KB/s
Value - $1 0
Pnce 0 Tokens

(a) UI optimized for optimal play

Time	Rounds Left	Tokens	Score
5s/7s	6/6	30/30	$0

Task Category
Medium Importance

Speed 400 KB/s
Value. $0 4
Pnce' 12 Tokens

Speed 300 KB/s
Value $0 2
Pnce 9 Tokens

Speed 100 KB/s
Value. - $0 3
Pnce' 3 Tokens

Speed 0 KB/s
Value - $1 0
Pnce. 0 Tokens

(b) UI optimized for behavioral play

Figure 3 7 A screenshort of the game from Experiment 2, illustrating the differences in the user interface when (a) optimized for optimal play, and when (b) optimized for behavioral play

only considered the 7-second time treatment for the computation of λ because we expected to see the strongest differences in the 7-second games

When studying design levers 1, 2, and 3, we solved the games optimally and computed the expected value of the game assuming perfect play Equipped with the user model learned in step 3), we can now compute the expected value of a game assuming sub-optimal play by human players For every configuration of the choice sets, this leads to a different expected value of the game Thus in step 4 of the optimization process depicted in Figure 3 6 we use the learned user model to search through the UI design space, i e , through all possible configurations of choice sets and compute the expected value of the game according to the user model We then choose the configuration with the highest expected value as the UI for "sub-optimal play" Consider Figure 3 7 where we display two screenshots for the games with fixed choice sets, illustrating the different choice sets resulting from the different UI optimization methods In Figure 3 7(a), the UI optimized for optimal play is shown, and in Figure 3 7(b), the UI optimized for sub-optimal (behavioral) play is shown Note that both UIs are *not* hand-picked, but the result of a computational search algorithm We see that the only difference between the two UIs is the top choice the UI that was optimized for optimal play gives the user the 900KB/s choice, while the UI that was optimized for sub-optimal play gives the user the 400KB/s choice This result is understandable in light of how the UI-optimization algorithm works The quantal-response assigns each action a certain likelihood of being chosen, corresponding to the Q-values of those actions Now, consider the top choice in Figure 3 7(a), which has a high value, but which can also cost between 9 and 27 tokens (this is a game with changing prices) Thus, in the worst case, the user spends 27 out of his 30 tokens with one click, and then has only 3 tokens left for the remaining 5 rounds Even if this action is very unlikely, the negative effect of an occasional mistake would be very large Consequently, the UI optimized for sub-optimal play shown in Figure 3 7 does not have such high-value high-cost choices, reducing the negative effect of mistakes

As before, we studied the effect of this design lever on *OptChoice* and found that the user's likelihood of selecting the optimal choice increased Thus, the optimization based on the behavioral model made the decision easier for the users However, the efficiency results for this particular design lever are more complex and interesting

Consider column (1) of Table 3 13 for the effect of design levers 3 and 4 on *Real-*

Table 3 13 GEE for the dependent variable *RealizedEfficiency* studying the effect of *OptimizedForSubOpt* Standard errors are given in parentheses under the coefficients The individual coefficient is statistically significant at the *10% level, the **5% level, the ***1% level, and at the ****0 1% level

Factors/Covariates	(1)	(2)	(3)
Intercept	0 462**** (0 0501)	0 004 (0 0639)	0 053 (0 1417)
AdaptiveChoiceSets?	0 077** (0 0376)	0 08** (0 0367)	0 080** (0 0365)
OptimizedForSubOpt?	-0 111**** (0 0334)	-0 119**** (0 344)	
Lambda		0 103**** (0 0110)	0 100**** (0 0253)
SmallLambda=1			-0 065 (0 0530)
OptimizedForSubOpt *smallLambda=1			-0 069 (0 0500)
OptimizedForSubOpt *SmallLambda=0			-0 174**** (0 0391)
Model Fit (QICC)	106 927	98 265	101 895

ized Efficiency We see that the coefficient for *OptimizedForSubOpt?* is *negative* and statistically significant at $p < 0\ 001$ The optimization of the market UI assuming behavioral play actually had a negative effect on *Realized Efficiency* Thus, the behavioral model built around the quantal-response model and fitted to the data from Experiment 1 did *not* predict the users' decisions for study 2 accurately enough Based on the behavioral model, the *Realized Efficiency*from the UI optimized for optimal play should have been significantly lower than the *Realized Efficiency*when playing the game with the UI optimized for sub-optimal UI Section 3 4 2, which contains the analysis of the various behavioral factors on the user's decision making performance, offers a possible explanation for this effect The UI optimization was based on the quantal-response model which only takes the Q-values of the different choices into account It does not take into account 1) the number of choices available, 2) the relative or absolute position of the optimal choice, 3) whether the optimal choice had a positive or negative value, 4) the value of the choice one above the optimal choice, etc , even though we have found that all of these factors are highly statistically significant for the users' decision performance Thus, one possible explanation is that the behavioral model we used was too simple, and didn't capture enough of the users' behavior to suffice for a good UI optimization

Let's now take a more detailed look at the efficiency results Table 3 14 provides at least a partial explanation for what happened By re-optimizing the UI, we decrease the optimal efficiency (achievable for a perfectly rational player) from 1 0218 to 0 7819 Thus, we "took away" approximately $0 24 per game However, we never expected the users to come even close to the optimal efficiency values, but instead, based on our user model learned from study 1, we expected the users to do better in the re-optimized game such that the *Realized Efficiency*would actually increase However, as we can see in the last column of 3 14, the *Realized Efficiency*also dropped from 0 42 to 0 3296 Thus, relative to the optimal efficiency, the users did better in the re-optimized game, however, in absolute terms, they still did worse A potential explanation is that the users in study 2 acted "more rationally" than the users in study 1 However, the best fitting λ-parameters for study 1 and study 2 were very similar, and thus, the data does not support this hypothesis Yet, we found another interesting result As before in study 1, we computed a λ_i-parameter for each user in study 2, as well as one λ corresponding to the best fit across all users In addition, we compute a binary variable *smallLambda* for each user which denotes whether that user's λ is smaller or

Table 3.14: UI-Optimization: Effects on optimal and realized Efficiency

OptimizedForSubOpt?	*Optimal Efficiency*	*Realized Efficiency*
no	1.0218	0.4200
yes	0.7819	0.3296

larger than the *average* lambda, i.e., *SmallLambda* denotes whether the user belongs to the *more rational* or to the *less rational* group of users. Consider now column (3) of Table 3.13 where we also analyze the interaction effect of *OptimizedForSubOpt* and *SmallLambda*. We now see that for SmallLambda=0 (i.e., for the more rational users) the effect of *OptimizedForSubOpt* is particularly negative, i.e., for those users we made the game a lot worse by doing the re-optimization. However, for SmallLambda=1 (i.e., the less rational users) the effect of *OptimizedForSubOpt* is close to zero, and in fact not statistically significant. Thus, the data suggests that the less rational users did as well in the game whose UI was optimized for behavioral play as in the game whose UI was optimized for optimal play.

3.5 Summary

In this chapter, we have introduced a new research agenda on "market user interface design." Our goal is to understand how UI design choices for market environments affect users' abilities to make good economic decisions, and how we can develop automated methods to optimize market user interfaces. In studying this question, it is crucial to take the human nature of market participants into account, i.e., deviating from a perfectly ration agent model. Thus, our research explores a very complex space where human limited cognition meets computing. This is a largely unstudied research area with huge opportunities for work at the intersection of market design, intelligent agent systems, UI design, and behavioral economics We situate our study in a 3G bandwidth market where users can make different choices regarding bandwidth speed on their smartphones for different prices We designed a multi-step market game and ran a behavioral economics lab experiment with 53 users, testing the effect of four different design levers The game can formally be modeled as an MDP and thus our work also provides insights into how well humans can play MDPs under time pressure Our experimental results indicate that the users' actions were highly correlated with the Q-values of the choices available in the game, indicating that the users found very good sequential policies In our analysis, we identified a series of behavioral effects Perhaps one of the most important results concern the users' loss aversion without exhibiting any learning effects over time This finding raises concerns about users' general ability to allocate a fixed budget over time in real-world domains

Finally, we tested the effect of four different market UI design levers on users' *Realized Efficiency* When changing the number of choices, the *Realized Efficiency*increases as we go from 3 to 4 to 5 choices, and then slightly decreases as we go from 5 to 6 choices However, the decrease in efficiency was not statistically significant Thus, it seems that after some point, adding more choices (thereby making the UI more complex), doesn't help the user, and can potentially even hurt In future research, we want to study the effect of changing the number of choices in even more detail, running a similar experiment and adding 7 or 8 choices to the treatments to see if efficiency merely plateaus at some point, or even starts to decrease again In a second experiment, we studied the effect of the two design levers *Fixed vs Adaptive*

Choices and *UI Optimization* Our results show that having adaptive rather than fixed choice sets has a positive effect on users' Realized Efficiency This is a positive result, suggesting numerous applications where user interfaces could be tailored in various ways to context-specific needs of the users In contrast, and quite surprisingly, the UI that was optimized based on the behavioral model actually led to lower *Realized Efficiency*than the UI optimized for optimal play By splitting the participants into *less rational* and *more rational* users, we traced the efficiency reduction to the more rational users For those users, the re-optimized UI led to significantly lower efficiency, while there was no statistically significant effect on efficiency for the less rational users This naturally suggests a new research direction on "personalized market user interfaces," but we defer a more detailed discussion of this idea to the future work section in Chapter 6

One key finding in this chapter was that behavioral effects play an important role in users' decision making processes For the design of optimal market user interfaces, it is clearly necessary to depart from the assumption that users are *perfectly rational* and instead take their cognitive costs into account In the next chapter, we also consider behavioral effects, however, of a different kind We study users' social preferences in community-based systems, departing from the standard assumption that users are *self-interested* The particular domain we study is P2P file sharing, and one of the main research questions is to determine the factors that are most predictive for whether users act more selfishly or more altruistically

Chapter 4

Selfishness vs. Altruism in P2P File Sharing Networks

4.1 Introduction[1]

In 2002, Benkler [8] coined the term *peer production* to describe decentralized collaborations among individuals that result in successful large-scale projects In contrast to market-based and firm production, there are no price signals or managerial hierarchies in peer production to organize the group of contributors Although there is often little or no monetary incentive to contribute, many peer production systems flourish and have generated superior products (e g Linux, Wikipedia, Flickr, YouTube, BitTorrent)

These peer production systems can be modeled as public goods games, i e , con-

[1] The material presented in this chapter is based on collaborations with David C Parkes and Johan Pouwelse

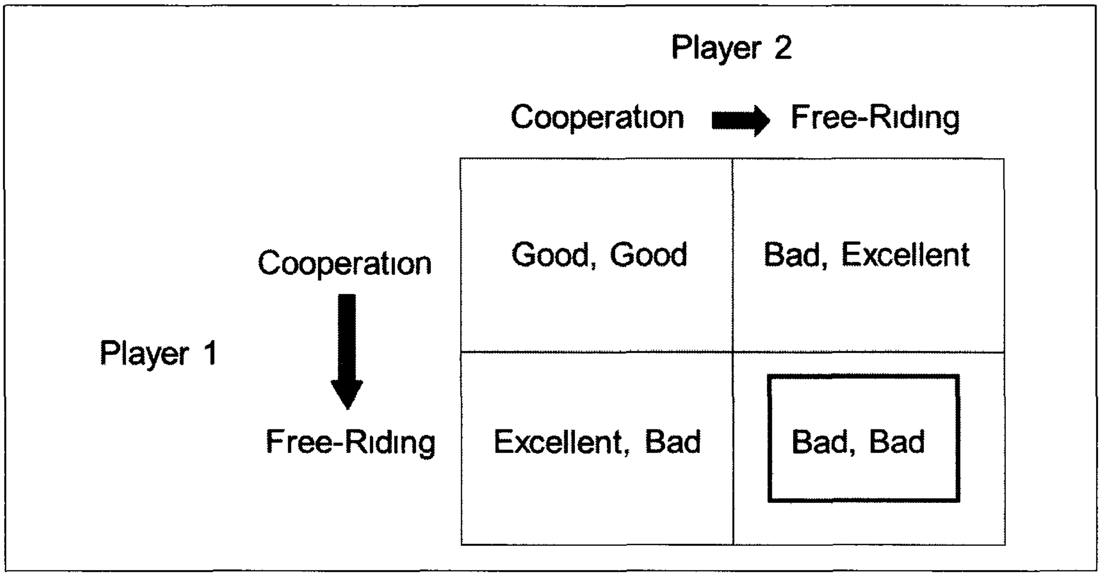
tribution games where the whole community of users benefits from individual contributions, but where each user is best off not contributing. Thus, from a standard economics perspective, it is surprising that millions of people contribute to peer production systems, even though free-riding would be easy because it is not penalized. Two standard assumptions in economics are a) that people are "selfish", i.e., they only care about their own well-being, and b) that people are "rational", i.e., they always choose the best-possible actions available to them [34]. Clearly, this model fails to describe the behavior observed in peer production systems where it seems as that users are not fully self-interested, but have some kind of "social preferences".

It is common to use the term *other-regarding preferences* to describe the preferences of people whose behavior suggests that they do not only care about themselves but also about the welfare of others. Psychologists and behavioral economists have identified three main reasons for other-regarding behavior: 1) reciprocity (I am kind to you if you are kind to me, I am mean to you if you are mean to me), 2) inequality aversion (people prefer outcomes where everybody gets similar payoffs), and 3) pure altruism (unconditional kindness, independent of the other actors' previous or future actions). Camerer and Fehr [12] present an excellent summary of lab studies for various games, providing convincing evidence that people have other-regarding preferences, at least when under observation in the lab.

In this chapter, we are primarily interested in people's behavior in *public goods games*, because that is the most appropriate model for peer production systems like Wikipedia or BitTorrent. A standard public goods game can be described as follows: We have $n \geq 2$ players and each player is endowed with x dollars. Each player can make a contribution $g_i \in [0, x]$ The sum of all contributions $G = \sum_{i=1}^{n} g_i$ is doubled and re-distributed equally to all players A game-theoretic model assuming rational and self-interested players predicts that each player contributes 0 dollars This is easy to see because for each dollar that player i contributed, he will get less than one dollar in return However, in lab experiments, the average contribution of players is about 50% of x A possible interpretation for this behavior is reciprocity players expect a certain degree of cooperation from the other players and they reciprocate this expected cooperation However, experiments have also shown that the contribution rates go down over time if the public goods game is repeated

The particular public goods game we are interested in is "Peer-to-peer File Sharing" Such systems are a good example of a peer production system that clearly outperforms its alternatives A P2P network can be used to distribute content (data files, videos, mp3s) in a distributed manner In contrast to centralized server architectures where all users have to download files from the same server, in a P2P file sharing network, servers are only used to maintain directories of popular files The file download itself then happens in a distributed manner via all peers that have (parts of) the desired file The efficiency of this system (download speed, availability) hinges significantly on the number of peers that have a file and on how much upload bandwidth these users make available Obviously, on average, the upload/download ratio across the whole P2P network must be balanced

Many P2P file sharing networks have trouble providing proper incentives for users to upload This problem arises because providing upload bandwidth is costly for a user in many ways His internet provider might impose a monthly limit on network traffic,



		Player 2	
		Cooperation	Free-Riding
Player 1	Cooperation	Good, Good	Bad, Excellent
	Free-Riding	Excellent, Bad	Bad, Bad

Figure 4 1 A simplified illustration of the file sharing public goods game

he might not be able to use other tools that need a good internet connection (e g , Skype) efficiently, he might want to shut off his computer when he is not actively using it, or he might even be afraid of legislative consequences if he is uploading copyrighted material All of these different costs are reasons why a (self-interested) user does not have an unconditional incentive to upload Our research goal is to understand which factors lead to higher contribution rates by individual users Our long-term goal is to increase the efficiency of distributed systems that rely, at least in part, on non-selfish user behavior

In the last decade, P2P file sharing protocols have evolved significantly from Napster, over FastTrack, Gnutella, Kazaa, and finally to BitTorrent While the incentives for uploading differ from protocol to protocol, free-riding is a wide-spread problem in all of them We can classify users into "cooperative" users that upload to the P2P community as much or more than they have consumed, and "free-riders" that try to minimize the amount of upload bandwidth they provide while consuming at a much higher rate To a sufficient degree of approximation, a public goods game is an appropriate model for P2P file sharing Consider Figure 4 1 where we display a simplified and stylized version of this game with just 2 players Each player can decide to either cooperate or free-ride If everyone cooperates, the overall health of the system is high and everyone benefits However, if a single player decides to free-ride, there is a small disadvantage for the whole system (a little bit of missing upload bandwidth from that player), but a big advantage for the free-rider (he can now use his bandwidth for other tasks) Thus, assuming selfish-rational players, every player has a dominant strategy to free-ride, and thus the only equilibrium of this game is for everyone to free-ride, which obviously leads to a complete halt of the system (without any uploads, no one can download anything)

However, while free-riding is a wide-spread phenomenon in file sharing networks, not everyone free-rides In fact, some studies estimate that 25%-40% of the Internet traffic is due to P2P file sharing traffic, indicating that these systems thrive Thus, the question arises of why people contribute to P2P file sharing communities, if they do not have to What percentage of these users behaves selfishly and what percentage exhibits other-regarding preferences? How does this percentage depend on the trade-off they perceive between personal and social benefit from being cooperative or a free-rider? What other factors influence their decision? But most importantly, we wanted to know whether the users of a P2P file sharing network actually understand the nature of the public goods game they are playing, and whether that makes a difference for their behavior.

4.1.1 Overview of Results

To answer all of these questions, we have designed and conducted an online field experiment. We have released a new file sharing client which attracted 50,000 visitors and resulted in 10,000 downloads of the new client. Each user was offered two versions of the software, one that was more "selfish" and one that was more "cooperative." The cooperative client was advertised as being able to download videos at normal speed, while requiring the users to upload as much as they download. The selfish client was advertised as being able to download videos at a faster speed (we varied the speed-up between 0% and 45%), while allowing the users to minimize their upload to others. After the users selected one of the two choices, we elicited whether they had understood the nature of the public goods game they were playing.

Via a multi-variate logistic regression analysis, we identify the main factors that increased or decreased a user's likelihood of selecting the selfish option. The most important factor was whether users understood the "tragedy-of-the-commons" aspect of the public goods game: for those users who understood the problem, the likelihood of choosing the cooperative client was 16% points higher than for those who didn't. The second most important factor was how much faster the selfish client was compared to the cooperative client. Increasing the speed-up from 0% to 10% increased the likelihood of choosing the selfish client by up to 15% points. However, we observe an interesting thresholding effect as increasing the speed-up further beyond 10% had no significant effect on users' behavior. Other factors that are highly predictive for user behavior are age, country-of-origin, and the user's operating system Our long-term goal is to better understand users' motivations for contributing in peer production systems, to enable the design of better collaborative systems in the future

4.1.2 Related Work

Public goods games have been extensively studied in the lab [12], and lots of evidence for other-regarding preferences has been found Researchers have also started to examine social preferences via field experiments However, those experimental designs are much more complicated and to date the research results are still somewhat inconclusive (see, e g , DellaVigna et al [23]) More recently, economists have begun to develop formal models that take other-regarding utility functions into account The two standard approaches are to explicitly model either inequality aversion or reciprocity [26, 78]

While many peer production environments like Wikipedia, the Linux community, or BitTorrent already work very well, some argue that introducing monetary payments into these systems could further increase their efficiency However, there is little evidence to support this hypothesis Benkler [8] argues that in peer production domains, intrinsic incentives are often more important than monetary incentives Some studies even suggest that paying people can decrease instead of increase their contributions [31] This effect is called "crowding out," and describes the phenomenon that extrinsic motivation via money might undermine the intrinsic motivation that was existent before It is likely that the crowding out effect is particularly strong in peer production domains because there the intrinsic motivation of the users is an extremely important part of their utility function But even in business environments, it is sometimes difficult to incentivize people via monetary payments For example, Cowgill et al [19] report that for the internal prediction market at Google, the employees were more excited about getting a T-shirt proving their participation than getting a $1,000 check, which ties in nicely with the literature on "awards as compensation" [30] Clearly, there are non-monetary incentives at play

P2P file sharing networks belong to one of the most widely-studied peer production systems We already mentioned the public goods nature of these systems, and the missing incentives for uploading for a self-interested user To date, all P2P file sharing systems being used in the real-world suffer, in one way or another, from misaligned incentives In their famous study of the Gnutella file sharing network, Adar and Huberman [2] have shown that approximately 70% of the users shared no files at all Furthermore they found that almost 50% of the total traffic came from only 1% of the peers Thus, a significant part of the file sharing community was "free-riding" (only downloading, not uploading) Similarly, in their study of the BitTorrent network, Pouwelse et al [76] found that more than 80% of the BitTorrent users go offline once they have finished downloading and more than 97% go offline within 10 hours after finishing the download These missing incentives for uploading are not only of theoretical importance but have significant effects on the efficiency of P2P file sharing networks in practice A better understanding of how users decide whether to contribute or not would help us in the design of more efficient P2P file sharing systems in particular, and peer production systems in general

4.2 Experiment Design

For the description of this experiment we use the terms “altruistic” or ”cooperative” to describe users that choose to upload files to others and “selfish” or “self-interested” to describe those that choose not to do so (i e , to free-ride) Thus, we only use these words to describe a one-time decision by the users and we do not imply any far-ranging psychological classifications

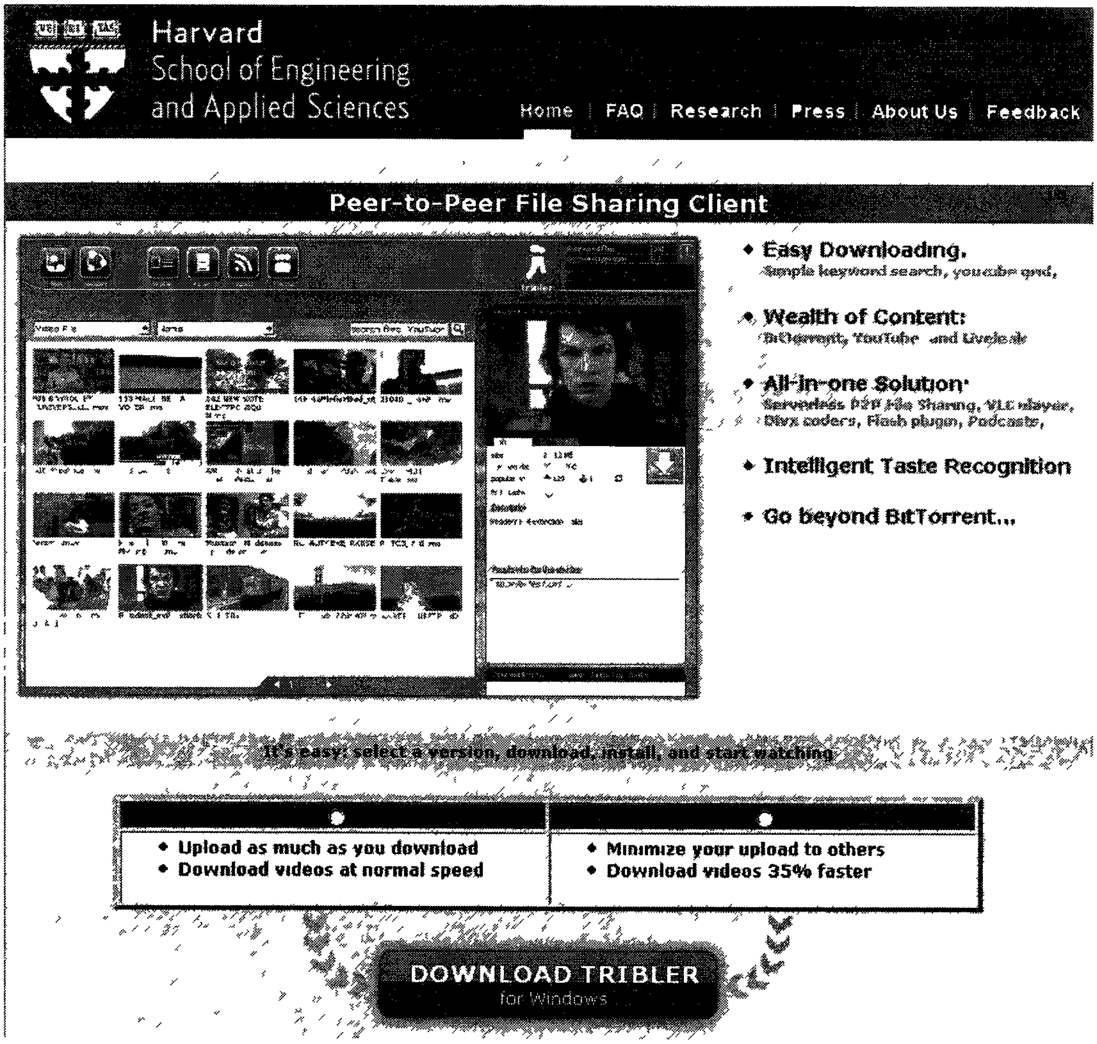
The main way we sought to examine selfish vs altruistic behavior among P2P file sharing users was by having them explicitly *choose* between two options, i e , by having them make a conscious decision between clearly altruistic or selfish behavior This is in stark contrast to all of the prior field experiments on P2P file sharing which show exactly how much users contribute, but do not reveal *why* they make those choices In particular, if a user decides to free-ride, this could be because he himself has made that choice consciously, or because his best friend has told him to un-select the upload box in his file sharing software, or because he has downloaded a particular file sharing application that does not upload to other users to begin with (as a default)

To study, in the field, how users make a choice between cooperating and free-riding, we released a new P2P file sharing client in two different versions the selfish model and the cooperative model We then observed the users’ download choice and subsequently asked them to fill out a short survey The file sharing application we released is called Tribler, being developed as part of a large research project at the Technical University of Delft in the Netherlands Tribler continuously tries to innovate by adding new features to their software, including such things as integrated search of Youtube or LıveLeak, automatıc taste recognıtıon, socıal networks features buılt dırectly ınto the applıcatıon, and so forth Due to these advanced features, we were able to advertıse theır newest release as a new form of Internet-TV

In fact, the Trıbler team developed two slıghtly dıfferent versıons of theır software, where the "selfish" versıon ıncluded an ımplementatıon of the Tor anonymıty network, whıch allowed that versıon to get slıghtly hıgher download speeds, at least ınıtıally We felt that actually havıng two dıfferent versıons was necessary for two reasons 1) we wanted to mınımıze the amount of deceptıon ınvolved ın the experıment, and 2) we were afraıd that some users would download both versıons of the software and check whether they were actually dıfferent or not (and ın fact, thıs dıd happen)

On August 29th, 2007, we released the new Trıbler versıon exclusıvely from our specıally desıgned webpage on a dedıcated Harvard web server at `tv seas harvard edu` One maȷor concern durıng the experıment desıgn phase was that the number of partıcıpants would be too low to achıeve any statıstıcal sıgnıficance Thus, we put a lot of effort ınto assurıng that the release of the new Trıbler versıon was well-publıcızed We ıssued a Harvard press release descrıbıng our ȷoınt research efforts to ımprove the efficıency of P2P file sharıng systems wıth a poınter to our websıte Thıs resulted ın a serıes of artıcles on promınent websıtes such as BBC Technology, Slashdot, and New Scıentıst, whıch drove a lot of traffic to our websıte Furthermore, we trıed repeatedly to get one of these artıcles on the frontpage of Dıgg com, a socıal bookmarkıng websıte In the end, we succeeded ın gettıng three dıfferent artıcles featured on the frontpage of Dıgg com, whıch also drove a lot of traffic to our websıte

For the desıgn of the websıte ıtself, we had to make sure that ıt was attractıve



Peer-to-Peer File Sharing Client

- Easy Downloading.
  Simple keyword search, youtube grid,
- Wealth of Content:
  BitTorrent, YouTube and Liveleak
- All-in-one Solution:
  Serverless P2P File Sharing, VLC player, Divx coders, Flash plugin, Podcasts,
- Intelligent Taste Recognition
- Go beyond BitTorrent...

It's easy: select a version, download, install, and start watching

○	○
• Upload as much as you download • Download videos at normal speed	• Minimize your upload to others • Download videos 35% faster

Figure 4 2 A screenshot of the frontpage of the tv seas harvard edu website

enough for users to download the new software and clear enough that users could make a conscious decision about the two different download versions, while making sure that we did not use any strong visual or textual cues that would bias the visitors of our website towards the selfish or the cooperative client In particular, we could not describe the two options as the " selfish version" and the "cooperative version," because we would have influenced the users significantly Thus, we put a lot of thought into the appropriate wording for the two download options, seeking to make them as neutral as possible while still succinctly explaining the two choices See Figure 4 2 for a screenshot of our website Note that the advertised speed-up (here 35%) varied across users

The users had to select one of the two download choices and then click on the large download button If a user clicked on the download button without having made a choice beforehand, a warning message would show up, telling the user that he must make a decision on which version to download first The descriptions of the two download choices were chosen to explicitly set up a one-shot public goods game The wording was chosen to suggest that the users would download an application with a fixed setting controlling how it behaved with respect to upload and download behavior The one-shot nature of the game was intended to make sure there was a strictly dominant strategy for a selfish-rational player, namely to download the version of the software that minimizes the uploads to others and downloads the videos faster than the other version Thus, the only equilibrium of the game as we set it up was for all users to choose the selfish version If we had told the users beforehand that they could change the behavior of their software later, the problem would have turned into a repeated game where users could react to the behavior of other users In such games there exist multiple more complicated equilibria, a situation that we wanted to avoid

Once a user successfully started the download of one of the two software versions, he was simultaneously forwarded to a survey consisting of 7 questions We needed

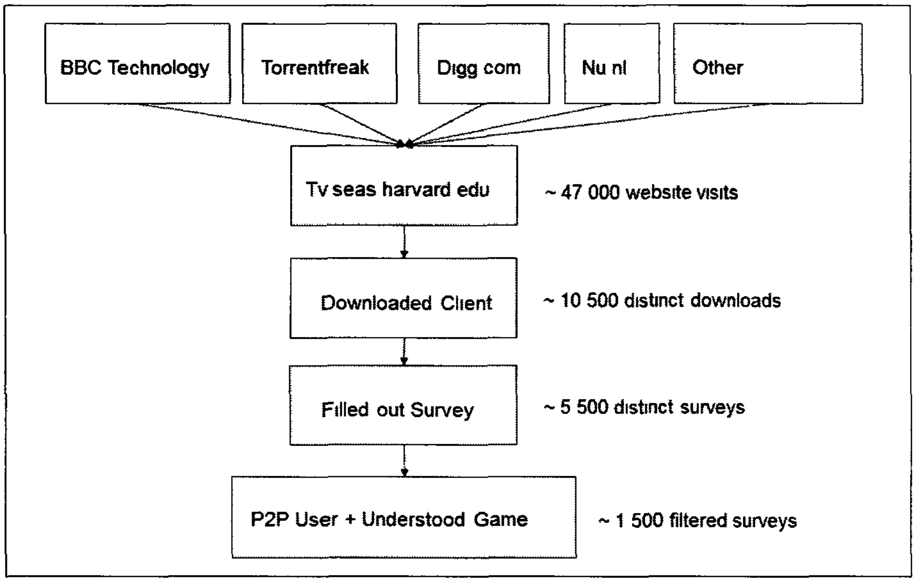


Figure 4 3 The filter chain for the experiment

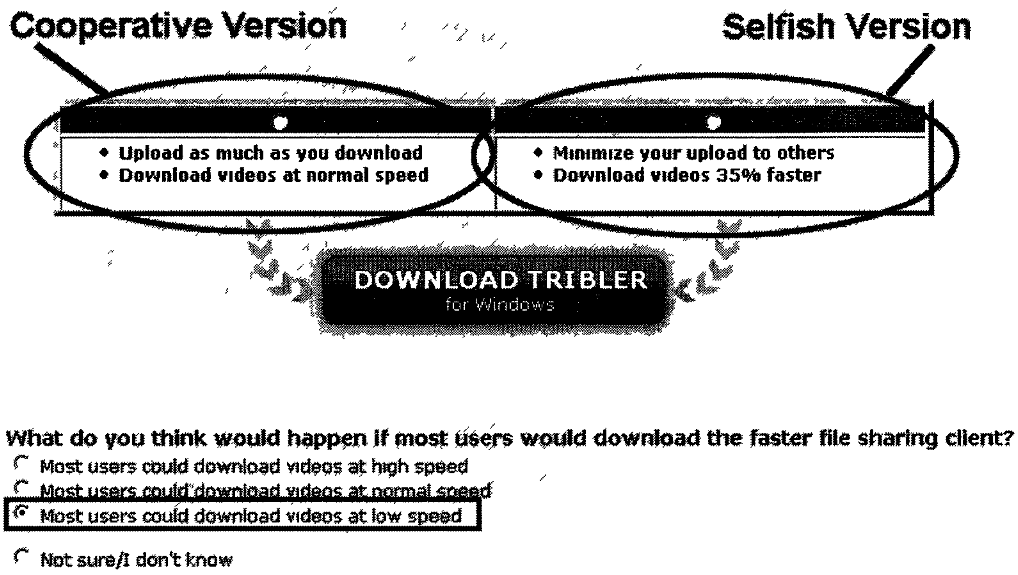
this survey to be able to filter our participants For example, we needed to know if they had ever used P2P software before and were thus at least vaguely familiar with the P2P concept Furthermore, we wanted to know if they understood the download decision they had made on the first page This was particularly interesting when it came to seeing how the behavior of informed vs uninformed players differed in this game

The design of the survey posed a series of new challenges, in particular because we wanted to make sure users did not suspect that the exercise was an experiment Thus, we could not use any kind of suspicious wording in either our press release, website design, or survey design We decided to call this a “Software Improvement Survey,” to indicate that we wanted users to answer the questions truthfully so that we would be able to produce better quality software in the future We therefore put a lot of thought into how best to disguise the intent of the question regarding whether they understood the download decision they made The exact wording of this question is described further below

See Figure 4 3 for an overview of the complete filter chain the users came from a variety of websites and in total we had approximately 47,000 unique visitors More than 10,000 of those chose to download one of the two clients, and more than 50% of those that downloaded the software filled out the survey A possible explanation for this surprisingly high number of completed surveys is that downloading the software itself probably took most users 1–2 minutes, which is about the same amount of time it took the average user to complete the survey Thus, it is possible that most users did not mind filling out the survey while waiting for the download to complete

We varied the text describing the two download options depending on the IP address of the visitors The screenshot shown in Figure 4 2 tells the visitor that the selfish version is 35% faster than the cooperative version We varied this "stimulus" between 0% and 45% in steps of 5 percentage points Thus, we had 10 different experimental groups, where the 0% group was somewhat special in that we also had to adjust the question regarding whether those users understood the download decision Figures 4 5 and 4 4 give screenshots of the download options shown to the 0-percent and the 35-percent group together with the corresponding question number 5 in the survey

Figure 4 4 shows the download options and the corresponding question testing the users' understanding for the 35% speed-up treatment The user could either choose "Upload as much as you download and download videos at normal speed"



- Upload as much as you download
- Download videos at normal speed

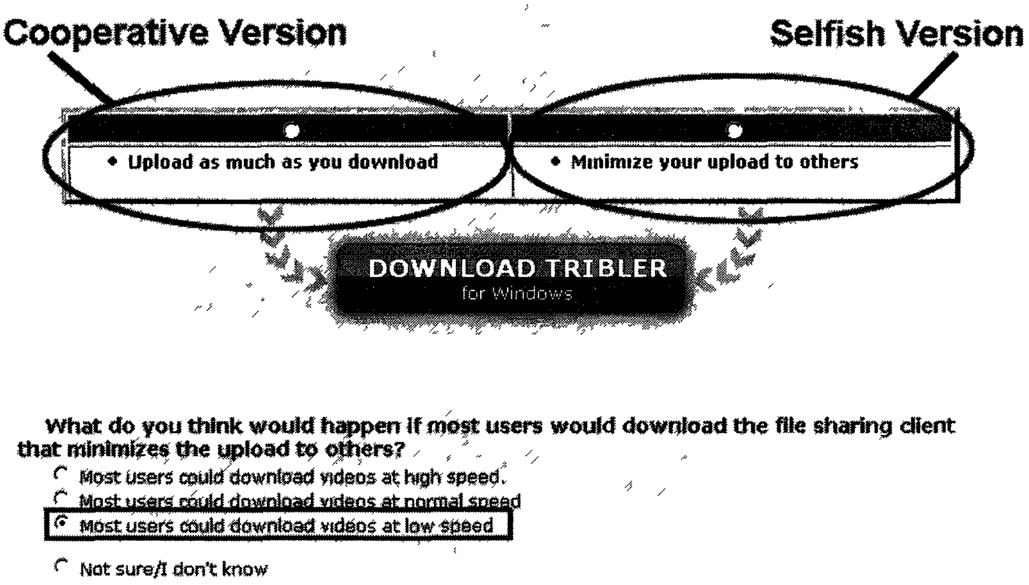
- Minimize your upload to others
- Download videos 35% faster

What do you think would happen if most users would download the faster file sharing client?

○ Most users could download videos at high speed
○ Most users could download videos at normal speed
● Most users could download videos at low speed
○ Not sure/I don't know

Fıgure 4 4 Download optıons and correspondıng survey questıon for the 35% group

(the cooperatıve versıon) or "Mınımıze your upload to others and Download vıdeos x% faster" (the selfish versıon), where x was varıed between 5 and 45 We randomızed whıch of the two optıons appeared on the left or on the rıght The correspondıng questıon ın the survey was changed slıghtly to "What do you thınk would happen ıf most users would download the faster file sharıng clıent?" Note that thıs ıs a quıte dıfficult questıon If a user dıd not read or understand the choıces on the first page and trıed to answer the questıon based sımply on the phrasıng of the questıon, he would choose an ıncorrect answer, because ıt seems ıntuıtıve that faster clıents lead to faster downloads Only ıf a user carefully read the descrıptıon on the first page, understood ıts ımplıcatıons, and remembered all of that once he got to questıon number 5 ın the survey, would he be able to answer the questıon correctly


Cooperative Version
Selfish Version
• Upload as much as you download
• Minimize your upload to others
DOWNLOAD TRIBLER
for Windows

What do you think would happen if most users would download the file sharing client that minimizes the upload to others?

○ Most users could download videos at high speed.
○ Most users could download videos at normal speed
● Most users could download videos at low speed
○ Not sure/I don't know

Figure 4 5 Download options and corresponding survey question for the 0% group

Figure 4 5 shows the download options and the corresponding question testing the users' understanding for the 0% speed-up The users could either choose "Upload as much as you download" (the cooperative version) or "Minimize your upload to others" (the selfish version) We believe that this wording is as neutral as possible while still being clear about the available choices The corresponding question in the survey asks, "What do you think would happen if most users would download the file sharing client that minimizes upload to others?" Of course, the correct answer to this question is "Most users could download videos at low speed," because if most people free-ride the system performance degrades Note that answering this question correctly is much easier than for the corresponding question in the 0% speed-up treatment In fact, this difference in difficulty with regards to answering the question correctly was observed in the data About 66% of the users answered the question correctly if they were in the 0% category For all other speed-up values, on average about 33% answered the question correctly We will later see in the regression analysis, that users are not just getting the answer right by chance

4.3 Results: Selfishness vs. Altruism

In this section, we describe in the detail the results from analyzing the data from the field experiment For the statistical data analysis, we performed a binary logistic regression with *Selection Value* as the dependent variable The *Selection Value* equals 1 if the participants selected the "selfish" client, and 0 otherwise

4.3.1 Speed-up: from 0% to 45%

The first factor we consider is the speed-up, which we varied between 0% and 45% Consider Figure 4 6 where we plot the speed-up on the x-axis and the population likelihood for selection the selfish client on the y-axis Note that in this graph we have not connected the data points for 0% and 5% speed-up to indicate that for these two data points more than just the stimulus has changed However, in the remaining graphs we will connect the two data points for simplicity

For this graph, we used all data points where the users reported they had used a P2P client before, and where the users had answered question 5 correctly, i e , they understood the public goods game they had just played In some sense, this subset of the population was the most knowledgeable one, i e , those users were most likely to make an informed decision between the two clients we offered them We see that

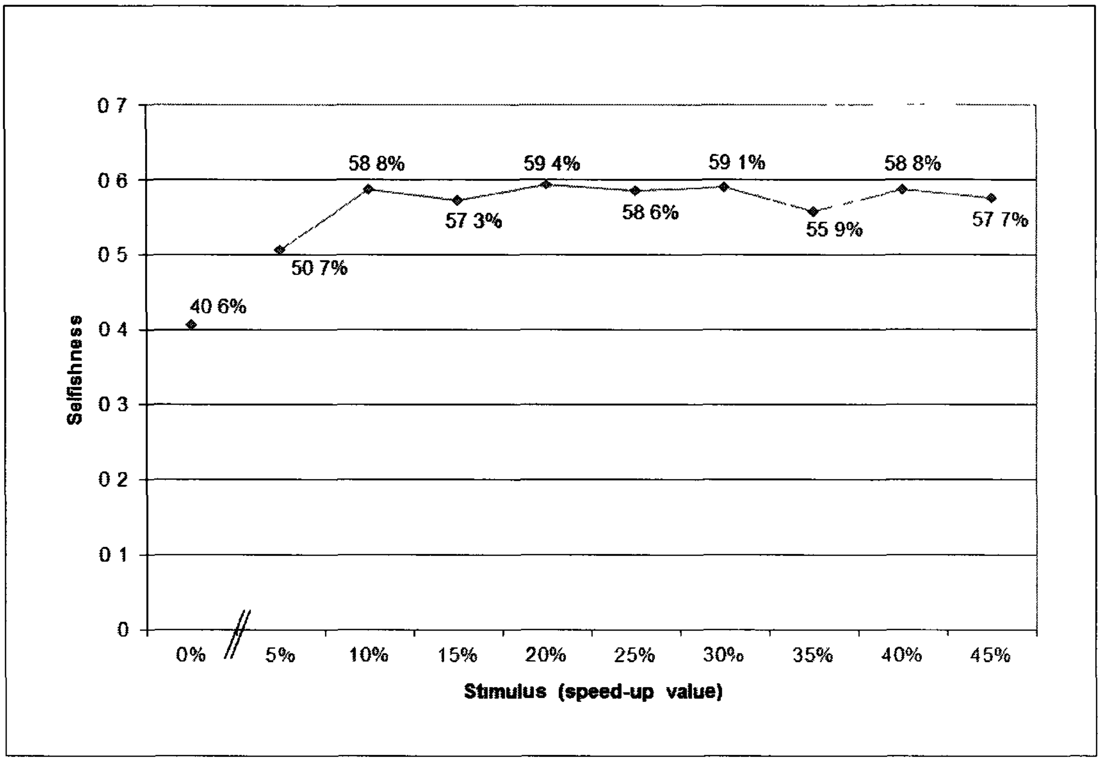


Figure 4 6 Dependency between the stimulus (speed-up %) and selfishness (ı e , the percentage of users selectıng the selfish clıent) Based on users who had used a P2P file sharıng clıent before and understood the nature of the publıc goods game (N=1349)

wıth 0% speed-up, approxımately 40% of the populatıon chooses the selfish clıent Thıs ıs ratıonal for a selfish player because even though the download speed ıs not larger than for the cooperatıve player, uploadıng to other players also ıncurs a cost that ıs mınımızed wıth the selfish versıon When ıncreasıng the stımulus from 0% to 5% to 10%, we see that the percentage of selfish players ıncreases by approxımately 10 percentage poınts ın each step up to almost 60% for the 10% speed-up stımulus Thus, there ıs a sıgnıficant part of the populatıon that cares about personal download speed—more so than about the well-beıng of the P2P communıty

What ıs surprısıng ıs that beyond the 10% speed-up stımulus, the percentage

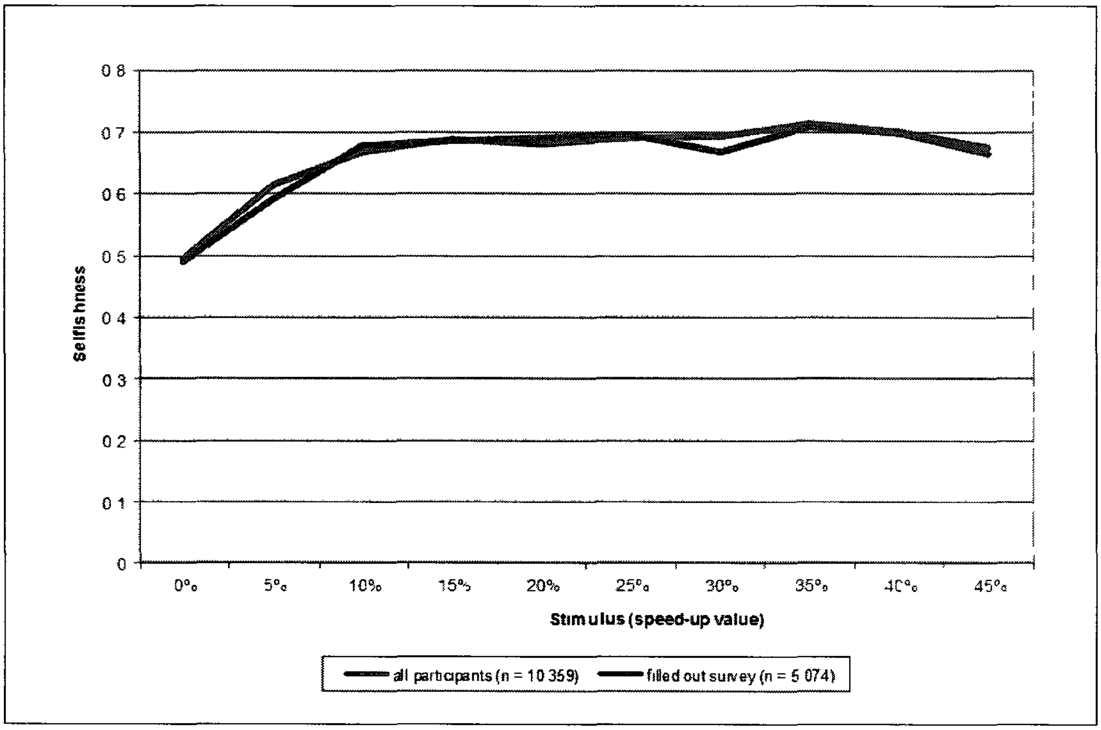


Fıgure 4 7 Dependency between the stımulus (speed-up %) and selfishness (ı e , the percentage of users selectıng the selfish clıent) Comparıng all users who downloaded any clıent (N=10,359) wıth those users who also filled out the survey (N=5,074)

of selfish players stays more or less constant, at around 59% Thus, those 20% of the users who make a decısıon contıngent on the speedup already choose the selfish versıon when the stımulus ıs as low as 10% For the remaınıng 80% of the populatıon, the speed-up value does not seem to ımpact theır download decısıon, ı e , they eıther always choose the selfish clıent or always the cooperatıve clıent

All Results vs Survey Results Note that the result we presented ın Fıgure 4 6 was based on 1,349 users who, accordıng to theır survey answers, had used P2P software before and understood the free-rıdıng problem Thus, we only evaluated the

Table 4 1 Binary Logistic Regression for the dependent variable *Selection Value* Showing the effect for independent variables *Speed-up*, *Has Used P2P Before*, and *Understood Public Goods Game* Standard errors are given in parentheses under the coefficients The individual coefficient is statistically significant at the *10% level, the **5% level, the ***1% level, and at the ****0 1% level

Factors	(1)		(2)		(3)		(4)	
	B	Exp(B)	B	Exp(B)	B	Exp(B)	B	Exp(B)
Constant	0 365**** (0 057)	1 440****	0 071 (0 075)	1 073	0 238* (0 122)	1 268*	0 508**** (0 126)	1 661****
Speed-up (5% step)	0 064**** (0 011)	1 066****	0 289**** (0 040)	1 335****	0 290**** (0 040)	1 336****	0 217**** (0 041)	1 243****
Speed-up squared (5% step)			-0 025**** (0 004)	0 975****	-0 025**** (0 004)	0 975****	-0 019**** (0 004)	0 981****
Has Used P2P Before					-0 185* (0 106)	0 831*	-0 107 (0 107)	0 899
Understood Public Goods Game							-0 626**** (0 067)	0 535****
Fit (Nagelkerke R^2)	0 010		0 020		0 021		0 045	
Cases Considered	n=4772		n=4772		n=4772		n=4772	

"expert users," so to speak A valid concern is that the data evaluation is biased because users who are willing to fill out a survey might be more cooperative to begin with To see if there was an inherent bias, we compared the results for the 5,074 users who did fill out the survey with the results from the whole user population that downloaded the software The results of this comparison are shown in Figure 4 7 We see that the two curves lie almost perfectly on top of each other Thus, it seems that there is no self-selection bias introduced by the survey process A possible explanation for this result is that users could fill out the survey while waiting for their download to complete and thus the extra cost for filling out the survey was low

Statistical Analysis of Speed-up Now consider Table 4 1 where we present the results of running a binary logistic regression for the dependent variable *Selection Value* to see if the results we just saw graphically are indeed statistically significant The results in this table are based on all data points for which we have valid survey results (n=4772) Consider column (1) where we only added the *Speed-up* factor to the regression We see that the speed-up has a statistically significant positive effect on the selection value, i e , the higher the speed-up the more likely the users were to choose the selfish client If we consider the odds ratio, i e , Exp(B), we see that a 5% speed-up increase corresponds to an increase of about 7% in the odds of selecting the selfish client However, we saw in Figures 4 6 and 4 7 that the effect of increasing the speed-up is particularly large when going from 0% to 5% to 10%, but then essentially levels off To account for this in the regression analysis, we add a quadratic term *Speed-up squared* to the regression Consider now column (2) of Table 4 1 where we see that both the linear and the quadratic terms are statistically significant The linear term has a positive coefficient and the quadratic term has a negative coefficient, thus, confirming what we already saw visually from the graphs, that the effect is particularly large at the beginning and then plateaus

Now consider column (3) of Table 4 1 where we added the binary factor *Has Used P2P Before*, which is based on the voluntary answer that users gave for question 1 on the survey We see that this factor is also statistically significant, but has a negative effect on *Selection Value* Thus, those users who reported they had used a P2P file sharing client before were much less likely to select the selfish client (corresponding to an odds reduction of about 17%) Note that the coefficient estimates for the other factors have remained very stable and the effects are still highly statistically significant

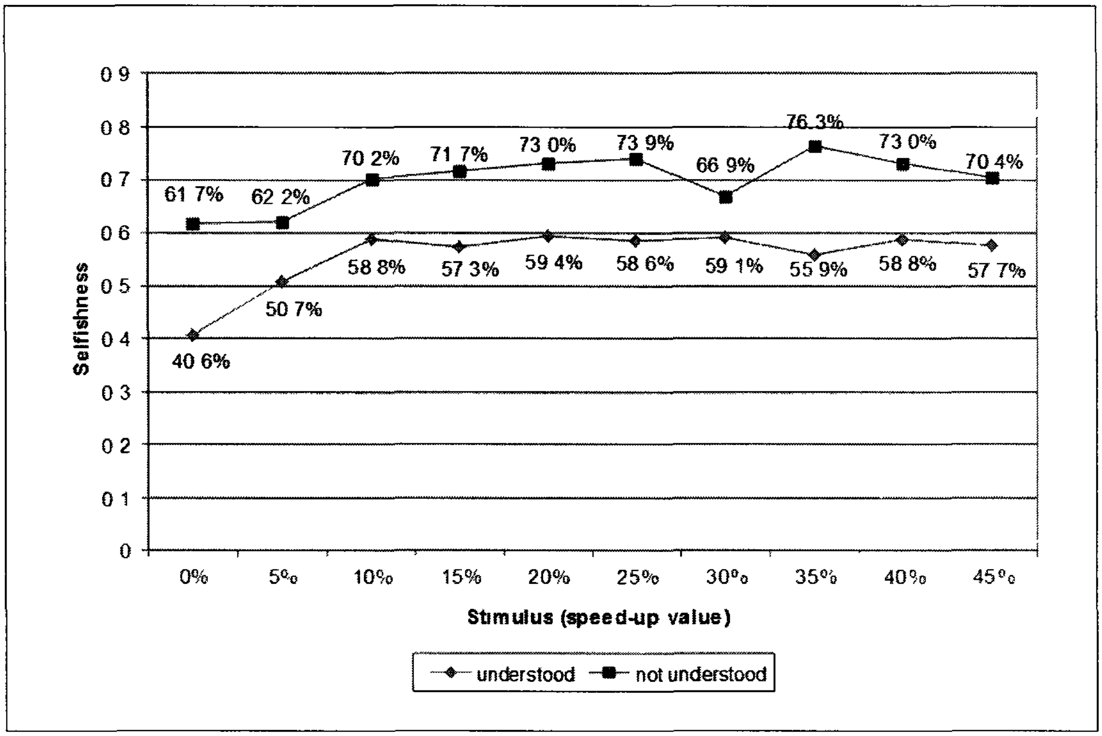


Fıgure 4 8 Dependency between the stımulus (speed-up %) and selfishness (ı e , the percentage of users selectıng the selfish clıent) Comparıng users who understood the publıc goods game (N=1,394) wıth those who dıd not (N= 3,080)

4.3.2 Understanding the Public Goods Game

We now move on to the analysıs of the most ınterestıng factor whether the user **understood the P2P publıc goods game or not** Recall that we ıncluded a somewhat trıcky questıon elıcıtıng thıs ınformatıon from the survey, so that we would be able to separate out the ınformed users from the unınformed users Now, ıt could have well been that thıs questıon was eıther too complıcated to understand/answer or that there ıs actually no sıgnıficant dıfference between users who answered correctly or ıncorrectly In fact, our data strongly suggests otherwıse, a fact that mıght well have important implications for future systems design Figure 4 8 shows the comparison between the 1,394 users who used P2P before and understood the download choice and the 3,080 users who used P2P before but did not understand the download choice We can clearly see the separation of the two data lines by on average of 15 percentage points Thus, users who understood the free-riding problem were much more altruistic than the other users

Now consider column (4) of Table 4 1 where we added the binary factor *Understood Public Goods Game* to the logistic regression This factor was 1 if the users had answered question 5 of the survey correctly, and 0 otherwise Now two interesting things happen First, the factor *Has Used P2P Before* is no longer statistically significant, as it was correlated with the new factor Thus, going forward we will exclude this factor when controlling for *Understood the Public Goods Game* But more importantly, the factor *Understood Public Goods Game* has a highly statistically significant negative effect Consider the odds ratio holding everything else constant, the odds for selecting the selfish client were only **half as high** when the user understood the public goods game, compared to when he did not Thus, a user who had understood the underlying public goods game was much more likely to select the altruistic client

4.3.3 Age

The next factor we consider is the participants' age, as self-reported on the survey The average age was about 29 years, and in fact more than 50% of the users were between 20 and 30 years old Thus, when making general statements about user populations we must be aware of the fact that while we are getting a good sample

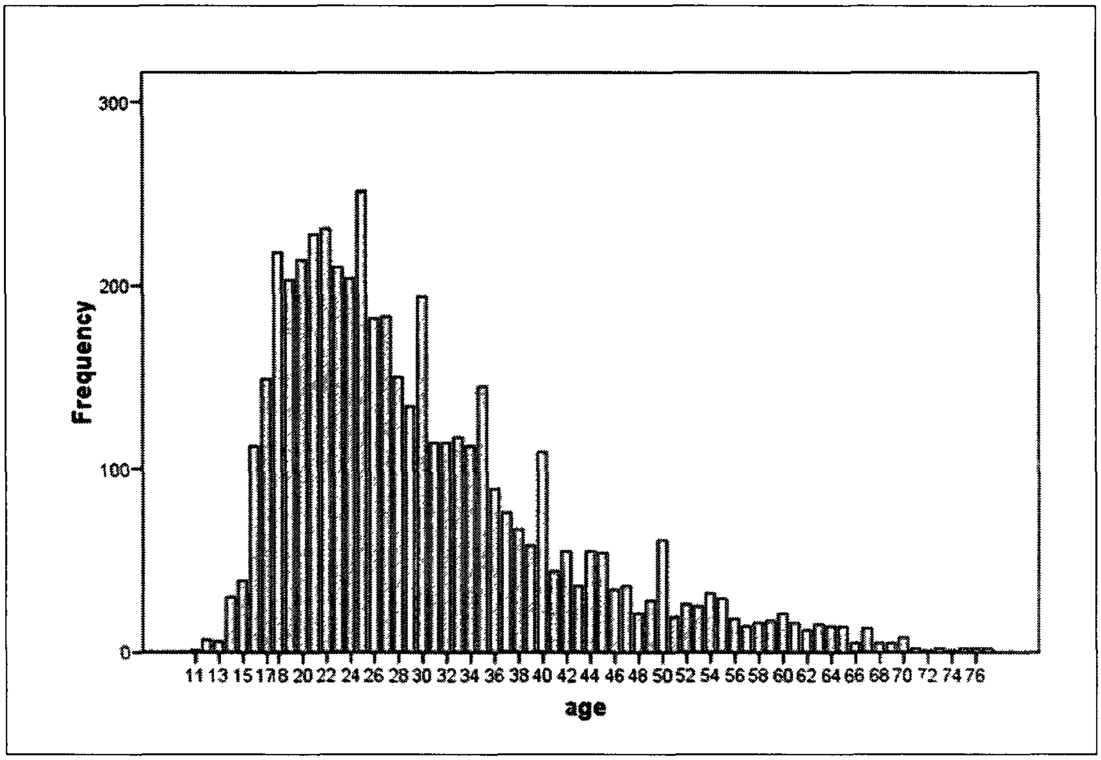


Figure 4 9 Distribution of users by age We are only considering users who had used a P2P file sharing client before, who understood the public goods game, and who reported an age between 10 and 80 N=1,335

of the typical P2P user population, this is not a representative sample of the whole internet population

Note that this age data might be noisy, given that we do not expect everybody to enter his true age when filling out the survey For the analysis, we eliminated all responses that stated an age below 10 years or above 80 years Now, consider column (1) in Table 4 2 where we added *Age* to the regression, while still controlling for *Speed-up* and *Understood the Public Goods Game* We see that the factor has a small but highly statistically significant effect on the selection value Looking at the

Table 4 2 Bınary Logıstıc Regressıon for the dependent varıable *Selectıon Value* Showıng the effect of ındependent varıables *Age*, *Operatıng System* and *Country* Standard errors are gıven ın parentheses under the coefficıents The ındıvıdual coefficıent ıs statıstıcally sıgnıficant at the *10% level, the **5% level the ***1% level, and at the ****0 1% level

Factors	(1)		(2)		(3)	
	B	Exp(B)	B	Exp(B)	B	Exp(B)
Constant	0 731**** (0 120)	2 077****	0 718**** (0 124)	2 051****	0 847**** (0 145)	2 332****
Speed-up (5% step)	0 209**** (0 041)	1 233****	0 211**** (0 041)	1 235****	0 193**** (0 046)	1 213****
Speed-up squared (5% step)	-0 018**** (0 004)	0 982****	-0 018**** (0 004)	0 982****	-0 016**** (0 005)	0 984****
Understood Publıc Goods Game	-0 673**** (0 068)	0 510****	-0 661**** (0 070)	0 516****	-0 672**** (0 078)	0 511****
Age	-0 010**** (0 003)	0 990****	-0 010**** (0 003)	0 990****	-0 008*** (0 003)	0 992***
Operatıng System = Wındows			0	1	0	1
Operatıng System = Lınux			-0 214** (0 101)	0 807**	-0 275** (0 113)	0 760**
Operatıng System = Mac			0 210*** (0 082)	1 234***	0 179** (0 091)	1 196**
Country = US					0	1
Country = Australıa					-0 287 (0 184)	0 751
Country = Brazıl					-0 208 (0 230)	0 751
Country = Canada					-0 474*** (0 151)	0 622***
Country = Chına					-0 743*** (0 284)	0 476***
Country = Germany					-0 084 (0 155)	0 919
Country = Spaın					-0 286 (0 177)	0 751
Country = Great Brıtaın					-0 558**** (0 114)	0 572****
Country = Indıa					0 181 (0 270)	1 198
Country = The Netherlands					0 043 (0 169)	1 043
Country = Norway					-0 121 (0 117)	0 886
Country = Sweden					-0 931**** (0 187)	0 394****
Fıt (Nagelkerke R^2)	0 048		0 051		0 072	
Cases Consıdered	n=4708 10 < *Age* < 80		n=4593 10 < *Age* < 80 OS∈ {wın,max,lınux}		n=3706 10 < *Age* < 80 OS∈ {wın,max,lınux} Frequency(Country)>100	

odds ratıo, we see that one year of age corresponds to a reductıon ın the odds for selectıng the selfish clıent by 1% Thus, the older partıcıpants were more altruıstıc

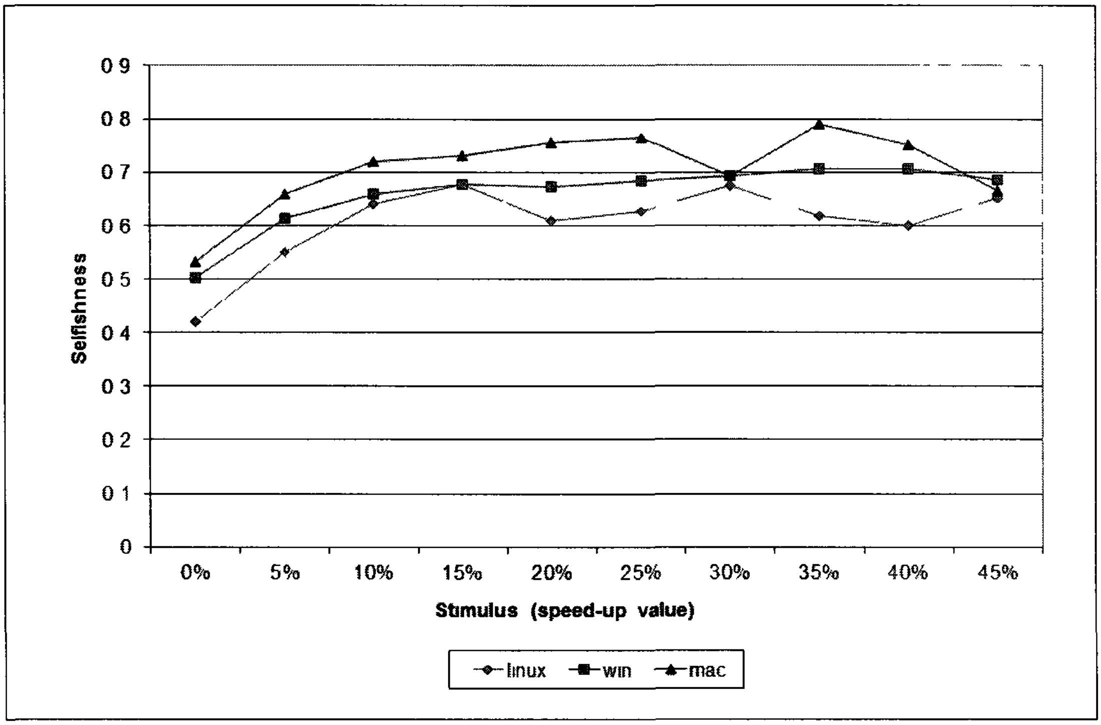


Fıgure 4 10 Dependency between the stımulus (speed-up %) and selfishness (ı e , the percentage of users selectıng the selfish clıent) Comparıng users wıth dıfferent operatıng systems (Lınux, Wındows, and Mac)

4.3.4 Operating System

The next factor we consıder ıs the user's operatıng system Consıder Fıgure 4 10 where we plot three lınes correspondıng to the three operatıng systems Wındows, Max, and Lınux We see very clearly that the lıne correspondıng to the Lınux users ıs consıstently the lowest, whıch means that the Lınux users are the most cooperatıve ones The Mac users seem to be the most selfish, whıle the Wındows users fall exactly ınto the mıddle between the Lınux and the Mac lıne Consıder now column (2) of Table 4 2 where we added the factor *Operatıng System* to the regressıon analysıs Agaın, the results are confirmed, wıth hıgh statıstıcal sıgnıficance Controllıng for all the other factors, we see that relative to the Windows users, the Linux users are much more likely to choose the cooperative client (odds ratio decrease by 24%), and the Mac users are much more likely to choose the selfish client (odds ratio increase by 20%) Obviously, we are only observing correlations here, but at least the result that the Linux users were the most cooperative users is very intuitive After all, Linux itself is a prototypical peer production system and has managed to build a strong sense of community among its users

4.3.5 Country of Origin

The last factor we consider is the user's country of origin Via the user's IP address we were able to determine their country (this IP to country mapping is imperfect but good enough for our purposes here) We had visitors from approximately 50 different countries and for most of those we did not have enough data to get statistically significant results For the following analysis we filtered the data and only considered those entries from countries with at least 50 data points (the effects for countries with less than 50 data points were generally not statistically significant) Consider column (3) of Table 4 2 where we added the *Country* variable to the regression analysis All coefficient estimates are relative to *Country=US* We see that while the differences between most countries and the US are not statistically significant, there are 4 countries that show a highly statistically significant effect ($p < 0\ 01$ or $p < 0\ 001$) Canada, China, Great Britain, and Sweden For all of these countries, the effect on *Selection Value* was negative, i e , participants from these countries were more likely to choose the cooperative client than participants from the US The effect is most pronounced for Sweden Considering the odds ratio, we see that the odds for participants from Sweden were 60% lower than the odds for participants from the US

Of course we are only observing correlations here, but we can think of at least three possible explanations for the effects we are seeing First, is makes sense that users from more socialist countries like Sweden are cooperative to a higher degree, since in that sort of nation the idea of community and sharing resources in a fair manner is much more common than in capitalistic countries like the US A second reason might be the different legal situations in those countries While downloading copyrighted material in Sweden or Canada is relatively safe, such users in the US have to fear much more severe consequences should they get caught A third reason might be the different broadband connections in the different countries In Sweden and Canada for example, most internet users have high-speed broadband connections, particularly with respect to the upload capacity, and thus they do not suffer significant negative consequences if they upload to a P2P file sharing systems

4.4 Summary

In this chapter, we have described an online field experiment to study the degree of selfishness and altruism among P2P file sharing users We have seen that, at least a subset of P2P file sharing users consider the trade-off between the personal and societal effects of their choices Our experimental results have shown that approximately 30-40% of the users will always make the altruistic choice, while about 40-50% of the users will always make the selfish choice However, the data suggests that about 20% of the users make their decision, dependent on the particular trade-off they perceive

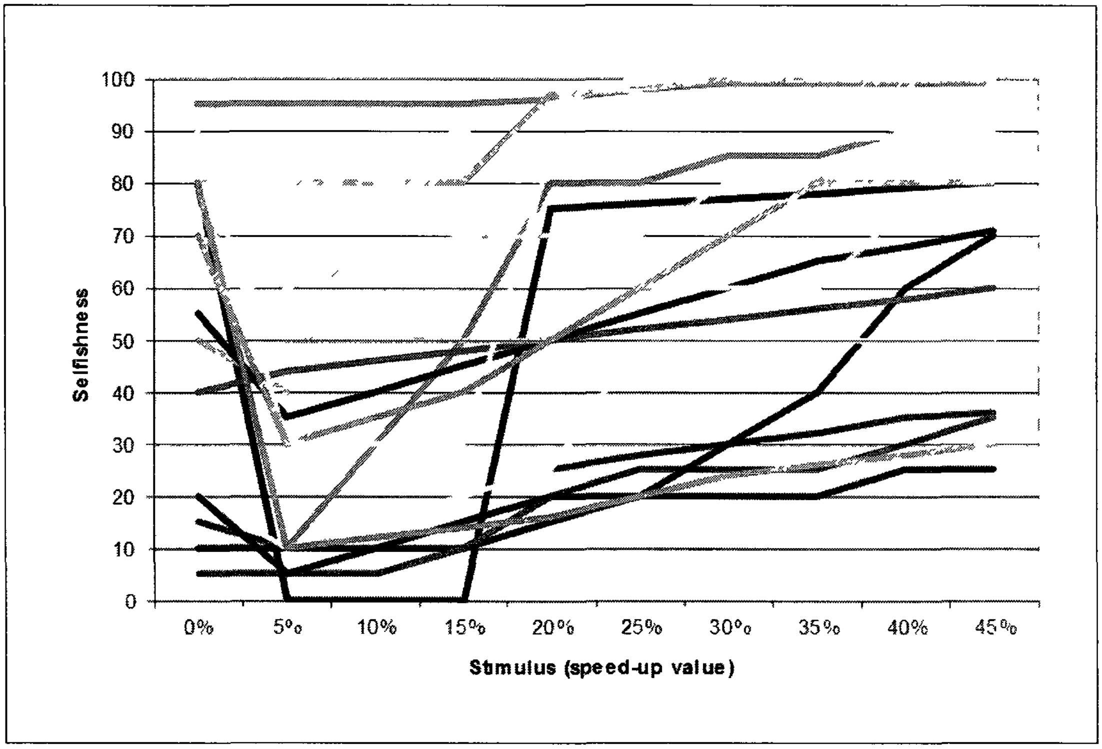


Figure 4 11 And what did the experts think ? Dependency between the stimulus (speed-up %) and selfishness (i e , the percentage of users selecting the selfish client) based on the predictions of experts (N=20) in the field

between how much a choice helps themselves and how much it helps/hurts the rest of the community This result was far from obvious before running the experiment In fact, we conducted a survey among 20 experts, and there was no agreement regarding what to expect from the experiment (see Figure 4 11)

We have also seen that we can assign different "priors" to different groups of the P2P community regarding their expected degree of selfishness For example, we found that younger people are more selfish, Linux users are more cooperative, and that even from country to country the behavior differs significantly Probably the most interesting and most important finding is, that users who understand the free-riding problem, i.e., the nature of the public goods game, were significantly more likely to cooperate. This result may have interesting consequences for the design of peer production systems and community-based markets. It is conceivable that by properly educating users about the consequences of their actions, once could increase the rate of cooperation. However, we defer a more detailed discussion of this idea to the future work section of Chapter 6.

Although we have shown that many P2P file sharing users exhibit social preferences, systems that purely rely on voluntary contributions are generally very brittle. Oftentimes, the long-term viability of such systems is in danger, if the personal and the societal incentives are very misaligned. In the next chapter, we address this problem by studying *work accounting mechanisms* for distributed work systems such as a P2P file sharing network. The goal is to prevent free-riding and to incentivize each user of the system to give back as much as he consumes. By aligning the personal and societal incentives in a suitable way, we significantly increase system efficiency.

Chapter 5

Work Accounting Mechanisms

5.1 Introduction[1]

Distributed work systems arise in many places, for example in peer-to-peer (P2P) file sharing networks like BitTorrent, where users upload and download files to and from other users, or in ad-hoc wireless networks where individual peers route data packages for each other. Of course, the total amount of work performed by a population must equal the total amount of work consumed. Moreover, while some degree of free-riding may be acceptable, the long-term viability of distributed work systems relies on roughly balanced work contributions. Otherwise, strategic agents may seek to free-ride on the system, i.e., minimize the work they perform for others and maximize the work they consume.

Current systems, including BitTorrent, often enforce temporally-local balances,

[1] The material presented in this chapter is based on collaborations with David C. Parkes, Michel Meulpolder, and Jie Tang.

e.g., via fair exchange protocols where agent A only continues to perform work for agent B if agent B reciprocates immediately. This "local balance" introduces a large inefficiency. Users are limited to consuming work at a rate at which they can themselves produce work, must be able to simultaneously consume and produce work, and cannot perform work and store credits for future consumption. Pouwelse et al. [76] found that this incentive problem has significant effects in practice as more than 80% of BitTorrent users go offline immediately once they have finished downloading. *Accounting mechanisms* solve this problem by keeping long-term tallies of work performed and consumed by each user. This gives users an incentive to share even after they have finished downloading, and thus increases system efficiency.

A particular challenge occurs when the interactions are bilateral and there is no ability for a third party to monitor the activities. We consider distributed work systems where agents perform small units of work for each other (e.g., transmitting a few bytes), for limited periods of time (e.g., a few seconds or minutes), where no contract covers the interactions, and where no real or virtual currency can be used, because the institutional requirements for the exchange of payments are not available. Furthermore, we assume that there is no a priori trust relationship between agents, and that an agent can only earn trust by performing work. Because accounting mechanisms rely on voluntary reports, however, a major challenge is to provide robustness against strategic manipulations. The two manipulations we consider are *misreports*, where an agent overstates the amount of work contributed or consumed, and *sybil manipulations*, where an agent creates *fake* sybils (or copies of itself). We design accounting mechanisms that are *incentive-compatible*, in the sense that no agent has an incentive to manipulate the mechanism

5.1.1 Accounting vs. Reputation Mechanisms

Misreport and sybil manipulations are well-studied in the related literature on trust and reputation mechanisms [32] However, the results from this literature do not translate to accounting mechanisms First, in distributed work systems, every *positive* report by A about his interaction with B, i e , B performed work for A, is simultaneously a *negative* report about A, i e , A received work from B This fundamental tension is not present in reputation mechanisms Second, sybil manipulations are much more powerful against accounting mechanisms For a search engine, the primary concern is that an agent could increase the reputation of its website by creating a set of sybils that are linking to the original website, but an agent does not care about the reputation of the sybils themselves In a distributed work system, in contrast, if an agent can create sybils with a positive score, then these sybils can receive work from other users without negatively affecting the score of the original agent While various reputation mechanisms have been proposed that are sybil-proof (e g , maxflow, hitting-time [16, 97]), these results do not translate to accounting mechanisms Third, once an agent has a high reputation it can benefit from that for a long time For example, a website with a high PageRank [72] benefits from lots of visitors without affecting its reputation In distributed work systems, in contrast, an agent benefits from a high score by getting priority for receiving work in the future, which in turn decreases its score again Thus, accounting scores are inherently temporary Finally, and somewhat informally, the essence of accurate reputation aggregation is the operation of *averaging* whereas the essence of accurate accounting is the operation of *addition* In a reputation system like eBay, individual users provide feedback about each other, and the individual feedback reports of two different agents regarding a third agent could be very different The task of the reputation system is to aggregate multiple reports into one overall reputation score, in a sense, "averaging" over all reports In contrast, in distributed work systems, multiple reports about work consumed or performed by an agent need to be "added together", to determine the overall net contributions of that agent

5.1.2 Real-World Applications for Accounting Mechanisms

There are many application domains where accounting mechanisms can help to provide proper incentives and increase overall system efficiency For example, the performance of P2P file sharing systems crucially depends on the contribution of resources by their participating users Free-riding is a well-known issue in P2P research, its effects have been empirically measured [2, 89] and extensively analyzed [5, 27, 57, 64] None of the existing decentralized systems provides its users with a long-term incentive to upload data to others [68], which results in large efficiency losses in practice Centralized systems like private BitTorrent trackers [67, 110] have managed to sustain long-term contributions from its members, even though many of them can be manipulated in various ways However, these systems generally assume some kind of central monitoring which we do not assume, and a formal study of these systems has been missing such that the incentives at play are still unclear

A possible future application for accounting mechanisms is content distribution in 3G networks. 3G bandwidth is a very scarce resource and downloading data via a 3G network is relatively slow and requires a lot of battery power from a smartphone or similar device. In contrast, Wifi networks are cheap and fast, and smartphones require much less battery power to connect to a Wifi network than to a 3G network. There is potential to use ad-hoc wireless networks to distribute certain data instead of using 3G bandwidth [7]. Imagine, you are at the train station at 9am in the morning, and lots of people are downloading the news, the weather, etc. onto their smartphones. In such situations, the efficiency can be increased if only one of the users downloads the data via the 3G, and then (automatically) distributes it via ad-hoc Wifi connections to the other users. However, given that 3G connections are costly for the user (draining the battery, and accumulating MBs that count towards the monthly download limit), with a naive implementation, no user would like to be the one who downloads and then distributes the data via 3G. A similar problem arises in an application proposed by Webb et al. [107], where multiple 3G antennas are shared to increase the download speed for an individual user. Again, with a naive implementation, it would be to a user's disadvantage to permit others to use the 3G connection on his device. Using accounting mechanisms, these incentive problems can be solved, balancing the work load over time, and giving each user an incentive to consume and perform work at different points in time.

A third potential application concerns routing in ad-hoc networks. Imagine you need to set up a communication network in an area without existing communication infrastructure, for example, in an area that has just been hit by a natural disaster or in a war zone. In such situations, oftentimes different organizational units, potentially from different countries, set-up camps next to each other, but have no direct relationship with each other Ad-hoc wireless networks are a very efficient way of quickly establishing a communication network in these situations However, bandwidth will generally be scarce, and routing data from other organizational units through your own network might reduce the amount of bandwidth you can use yourself Again, a properly designed accounting mechanism can address this problem by making sure that over time, each network routes approximately the same amount of data, thereby providing an incentive to collaborate

Accounting mechanisms can also be applied in domains that involve larger units of work by *human* agents, but where formal contracts or monetary transfers are undesirable for some exogenous reason Consider a carpooling network where drivers pick up and drop off passengers at different locations in a city According to the website of the *Casual Carpool Sites* from the Bay Area[2], no monetary payments are made from passengers to drivers, except for shares of the tolls While participating in this network as a passenger seems very attractive, the drivers also have some advantages, including the passengers' shares of the tolls and the right to use carpooling lanes Still, there seems to be more demand than supply, as is indicative by the warning on the website not to "line-jump" By using an accounting mechanism that gives priority to passengers that have themselves offered rides to others in the past, proper incentives for becoming a driver could be established

[2] `http //www ridenow org/carpool`

5.1.3 Outline and Overview of Results

In this chapter, we present a theoretical and experimental analysis of accounting mechanisms for distributed work systems In Section 5 2 we formalize the concept of a distributed work system and introduce BarterCast, a fully decentralized, lightweight information exchange system used here for gossiping voluntary work reports In Section 5 3 we present the first formal model for the design of incentive compatible accounting mechanisms A strawman solution, the Basic mechanism, is susceptible to misreport manipulations, even though it is built around a max-flow algorithm, which is robust against manipulations for reputation system We introduce the Drop-Edge mechanism, which removes any incentives for agents to misreport information, by selectively dropping information dependent on the decision context In Section 5 4 we provide a theoretical analysis of accounting mechanisms, where we show that the information loss of Drop-Edge due to dropping some of the information is small and vanishes in the limit as the number of agents in the network gets large We consider sybil attacks and prove an impossibility result, that under reasonable assumptions, no sybil-proof accounting mechanism exists This is in stark contrast to reputation systems, where mechanisms based on max-flow have been shown to be sybil-proof We show, however, that a weaker form of robustness, K-sybil-proofness, can be achieved for a restricted class of attacks

In Section 5 5, we provide an extensive experimental evaluation of the Drop-Edge mechanism for general distribution work systems, using a discrete, round-based simulation We show that, compared to the Basic mechanism, Drop-Edge leads to much higher performance for cooperative agents, because of its robustness against misreport attacks We also show that this effect increases over time, as agents gather more and more information about each other and thus are able to discriminate better and better between cooperative agents and free-riders In Section 5 6 we provide results from experiments using accounting mechanisms as an overlay protocol for the BitTorrent P2P file sharing network Using TRIBLER, a real file sharing client that is already deployed and being used by thousands of users, we run simulations at the BitTorrent protocol level We consider both a ranking policy and a banning policy for making work allocation decisions in BitTorrent, based on aggregate accounting information We show that using the ranking policy, which allocate the optimistic unchoking slot to the agents with the highest score, the power of accounting mechanisms is inherently limited in BitTorrent However, we show that using the banning policy, which bans agents whose score is below a certain threshold, we can significantly separate the performance of cooperative agents and free-riders, likely enough to induce free-riders to become cooperative Based on all experimental results, we conclude that the DROP-EDGE accounting mechanism can successfully separate cooperative agents from free-riders, and assuming some kind of behavioral change (i e , free-riders becoming cooperative over time), we can achieve a total efficiency that is higher than with the standard BitTorrent protocol

5.1.4 Related Work

The design of incentive-compatible distributed work systems has been a long-standing goal of the systems and AI community [40] In particular, since the advent of popular P2P file sharing networks like Napster, Gnutella, Kazaa, and BitTorrent, this area has attracted a lot of attention Early on, multiple studies have shown that users in these networks cheat in various ways Adar and Huberman [2] have shown that a majority of Gnutella users free-ride and Lian et al [62] have shown that users of the Maze network successfully perform whitewashing and collusion attacks

Despite the important differences between accounting mechanisms and reputation mechanisms, the related literature on transitive trust and reputation mechanisms [32] is an important precursor to our own work Gupta et al [41] present a reputation system that is partially distributed, but relies on the authority of a single agent that stores peer reputations Kamvar et al [51] present EigenTrust, an algorithm for reputation management in P2P networks that is based on distributed computations of globally consistent trust vectors However, it relies on pre-trusted peers for convergence and its aim for global consistency assumes a rigid network of peers Karma [104] is a system in which a distinguished set of nodes keep track of the transaction balances of peers However, they also assume the existence of a set of pre-trusted agents that store global information about all other peers, and the discussion of incentive issues is completely omitted Feldman et al [27, 28] study the challenges involved in providing robust incentives against free-riding, whitewashing, and misreport attacks in P networks They introduce a reputation mechanism based on max-flow, but implicitly assume that all nodes have a complete view of the network, which is unrealistic in large, dynamic communities Furthermore, the mechanism they propose is not misreport-proof in our setting

An interesting, but orthogonal direction is provided by studies of virtual currencies (in our domain, the institutional requirements for a transferable currency do not exist) Friedman et al [33] study scrip systems that rely on a trusted and transferable currency They show how to determine the optimal amount of currency in a system to maximize efficiency Kash et al [54] study the effect of hoarders and altruists on such scrip systems Dandekar et al [22] also study credit networks, but instead of relying on a globally-trusted currency, they employ locally-trusted IOUs Their study focuses on questions regarding the effect of network structure on credit liquidity, and largely ignores questions regarding incentive-compatibility

One of the largest steps forward regarding the implementation of robust incentives in a real-world P2P system used by millions of users is the BitTorrent protocol, proposed by Cohen [18] In contrast to previous protocols like Napster or Gnutella [88], BitTorrent uses a policy with short-term, direct incentives, resembling to a large degree a simple tit-for-tat mechanism However, while this mechanism is successful for short term transactions, BitTorrent offers no incentives for long-term sharing of content In practice, Pouwelse et al [76] found that a majority of BitTorrent users go offline immediately after finishing a download An interesting idea to address this propoblem was proposed by Piatek et al [75] They study what one might call "decentralized accounting mechanisms" and find empirically that P2P file sharing networks demonstrate a *small-world effect*, where 99% of peers exchanged data with a common third party They propose to use well-connected intermediaries to broker information, but without providing proper incentives to the intermediaries to behave truthfully Our own work builds on this idea, and in particular the fully decentralized information exchange protocol BARTERCAST exploits this *connectedness* of many P2P networks However, we additionally provide a formal framework to study the incentive properties of such systems and propose a mechanism that is misreport-proof, the major concern in the design of accounting mechanisms

Another important concern in the design of accounting mechanisms is *sybil-proofness* Cheng et al [15, 16] have studied this problem for reputation mechanisms One of their important findings is that no globally-consistent reputation mechanism can be sybil-proof, but that *subjective* mechanisms based on max-flow algorithms can be sybil-proof While their work influenced our thinking about sybil-proofness, unfortunately, their results do not translate to our domain, due to the differences between accounting and reputation mechanisms (especially the ability of a sybil to *receive* work) A recent paper by Resnick and Sami [80] also specifically addresses the problem of sybil-proof transitive trust mechanisms However, in their model, the individual transactions are risky and can have a positive or negative outcome, and they focus on limiting the effect of a powerful adversary In contrast, in our domain, the individual transactions are not risky Instead, our focus is on computing accounting scores that are proportional to the net work contributed by the agents Our mechanism shares some similarities with a mechanism proposed by Alon et al [4], who consider voting environments where the set of candidates coincides with the set of voters, and our theoretical analysis regarding the information loss of Drop-Edge was inspired by their analysis

To the best of our knowledge, we are the first to provide a formal framework to study accounting mechanisms and their incentive properties, and to point out the important differences between reputation and accounting mechanisms Furthermore, we are not aware of any practically feasible mechanism that is decentralized to the same degree as our proposal, misreport-proof, and tested under realistic conditions

5.2 Distributed Work Systems

Consider a distributed work system of n agents (or peers) each capable of doing work for each other All work is assumed to be quantifiable in the same units The work performed by all agents is captured by a work graph

Definition 10 ***(Work Graph)*** *A work graph $G = (V, E, w)$ has vertices $V = \{1, \quad ,n\}$, one for each agent, and directed edges $(i, j) \in E$, for $i, j \in V$, corresponding to work performed by i for j, with weight $w(i, j) \in \mathbb{R}_{\geq 0}$ denoting the number of units of work*

The true work graph is unknown to indivudal agents because they only have direct information about their own participation

Definition 11 ***(Agent Information)*** *Each agent $i \in V$ keeps a private history $(w_i(i, j), w_i(j, i))$ of its direct interactions with other agents $j \in V$, where $w_i(i, j)$ and $w_i(j, i)$ are the work performed for j and received from j respectively*

Based on its own experiences and known reports from other agents, agent i can construct a subjective work graph (see Figure 5 1) Let $w_i^j(j, k), w_i^k(j, k) \in \mathbb{R}_{\geq 0}$ denote the edge weight as reported by agent j and agent k respectively

Definition 12 ***(Subjective Work Graph)*** *A subjective work graph from agent i's perspective, $G_i = (V_i, E_i, w_i)$, is a set of vertices $V_i \subseteq V$ and directed edges E_i Each edge $(j, k) \in E_i$ for which $i \notin \{j, k\}$, is labeled with one, or both, of weights*

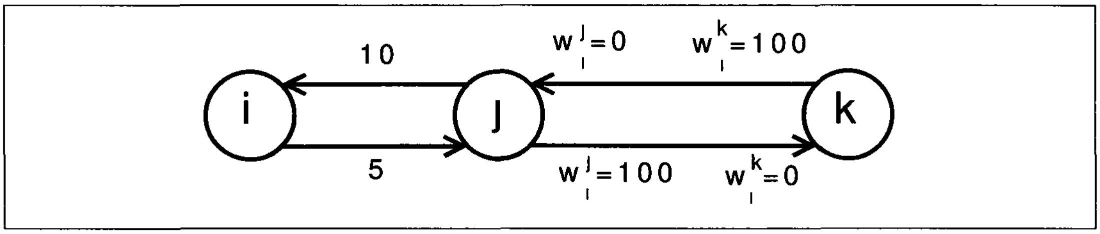


Figure 5 1 A subjective work graph from agent i's perspective Edges where i has direct information have only one weight Other edges can have two weights, corresponding to the possibly conflicting reports of the two agents involved

$w_i^j(j,k), w_i^k(j,k)$ as known to i For edges (i,j) and (j,i) the associated weight is $w_i^i(i,j) = w(i,j)$ and $w_i^i(j,i) = w(j,i)$ respectively

Edge weights $w_i^j(j,k)$ and $w_i^k(j,k)$ need not be truthful reports about $w(j,k)$ and thus can possibly be in conflict with each other, even if they have been submitted at the same point in time We now describe in more detail how agents can exchange information with each other, to obtain the information necessary to construct the subjective work graph

Throughout the chapter, we analyze and compare two different modes of information sharing between the agents *centralized* and *decentralized* information exchange For centralized information exchange, we assume the existence of a center (e g , a centralized server on the Internet) After every interaction, each agent makes a report to the center which stores all reports persistently At any point in time, an agent can query the center to obtain the most up-to-date information about all reports available at the center to then construct its subjective work graph based on his own information and the information obtained from the center Every agent has different *private* information about his own interactions, and the other information reported to the center is not necessarily correct Thus, the resulting subjective work graphs for the agents can differ, no matter whether centralized or decentralized information exchange systems are used

In many environments, a centralized information exchange system is simply not feasible, for example in wireless ad-hoc networks In other environments, a centralized system might not be desirable for many reasons a center represents a single point of failure, a center presents a bandwidth bottleneck, and a center requires some a priori trust in one entity and it is unclear how that trust should be established This motivates the study of fully *decentralized information exchange systems* BARTERCAST is such a fully decentralized, lightweight information exchange system Each agent keeps a private history of its direct interactions with other agents and obtains information about the rest of the network by exchanging a selection of its private history with others using bilateral messages We assume that agents can discover other agents with whom to exchange messages by using a *Peer Sampling Service* When two agents agree to exchange messages, then agent i selects for its messages the records of the N_h agents with the highest amount of work performed for i as well as the N_r agents most recently seen by i Thus, each agent will have an incomplete view and out of date view of the whole network

5.3 Accounting Mechanisms

5.3.1 Preliminaries

In a distributed work system, at every point in time, an agent can decide whether he is willing to perform work for others or not An agent who makes himself available

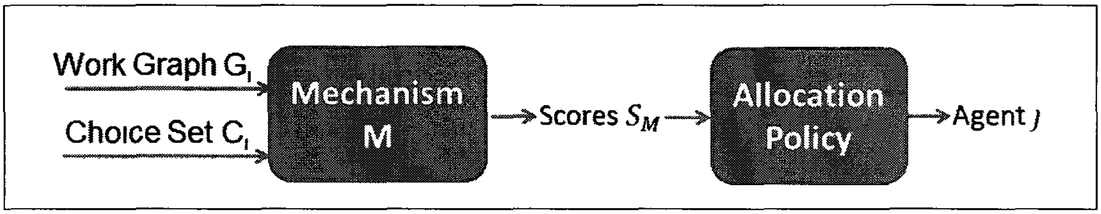


Figure 5 2 Accounting Mechanism and Allocation Policy based on the subjective work graph G_i and the current choice set C_i, the accounting mechanism computes a score $S_j^M(G_i, C_i)$ for each agent in the choice set Based on these scores, the allocation policy selects one agent for whom agent i will perform work

to perform work receives work requests by a set of agents (with which the agent may have rarely or never interacted with before) For example, in a P2P file sharing application, each agent that has any pieces of a particular file will be contacted by a group of agents that are all interested in some of those pieces At any moment in time, the contacted agent will have to choose for whom to perform work from this set of agents

Definition 13 ***(Choice Set)*** *We let $C_i \subseteq V \setminus \{i\}$ denote the choice set for agent i, i e , the set of agents that are currently interested in receiving some work from i*

The role of an accounting mechanism is to compute a *score*[3] for each agent $j \in C_i$, proportional to the net work contributed to the system, to allow agent i to differentiate between cooperative and free-riding agents We assume that an agent has no *a priori* bias towards assisting one agent over another

Definition 14 ***(Accounting Mechanism)*** *An accounting mechanism M takes as input a subjective work graph G_i, a choice set C_i, and determines the score $S_{ij}^M(G_i, C_i)$,*

[3] Note that we purposefully chose to use the term "score" instead of "reputation value" even though this is in contrast to prior work by Meulpolder et al [68] Our goal is to clearly distinguish between accounting and reputation mechanisms and to emphasize that outputs of such mechanisms have very different meanings

for any agent $j \in C_i$, as viewed by agent i

We let S_0^M denote the *default* score that accounting mechanism M assigns to an agent about which no information regarding work consumed or performed is available (i e , the two agents are disconnected in the subjective work graph) Once the accounting mechanism has computed a score for each agent in the choice set, the agent uses an *allocation policy* to decide to whom to allocate work to (see Figure 5 2) Thus, the accounting mechanism together with the allocation policy matches work-performing agents with work-seeking agents We consider the following two allocation policies

Definition 15 ***(Ranking Policy)*** *Given subjective work graph G_i, choice set C_i, and accounting mechanism M, agent i performs one unit of work for agent $j \in \arg\max_{k \in C_i} S_{ik}^M(G_i, C_i)$, breaking ties at random*

Definition 16 ***(Banning Policy)*** *Given subjective work graph G_i, choice set C_i, accounting mechanism M, and a banning threshold $\delta \in \mathbb{R}$, agent i performs one unit of work for an agent chosen uniformly at random from $\{j \in C_i | S_{ij}^M(G_i, C_i) \geq \delta\}$*

5.3.2 Agent Population and Strategic Manipulations

We adopt the model and terminology of Meulpolder et al [68], and assume a population that consists of a mixture of *cooperative* agents (or sharers), who always contribute work, and *lazy free-riders* who intermittently shirk work The role of an accounting mechanism is to make it unbeneficial to be a free-rider We further model a subset of the free-riding agents as *strategic* agents, who also try to manipulate the accounting mechanism itself through misreport attacks, where an agent reports false information about its work performed or consumed. The non-strategic free-riders are called "lazy" because they try to avoid performing work, but they are too lazy to perform any kind of manipulations. In Section 5.4.2 we study a second class of attacks on the accounting mechanism called *sybil attacks*, where an agent inserts fake agents into the network to manipulate the mechanism. Note that we model only strategic behavior with regard to manipulating the accounting mechanism and do not consider, for example, manipulations on the information exchange protocol.

Definition 17 ***(Misreport-proof)*** *An accounting mechanism M is misreport-proof if, for any agent $i \in V$, any subjective work graph G_i, any choice set C_i, any agent $j \in C_i$, for every misreport manipulation by j, where G'_i is the subjective work graph induced by the misreports, the following holds:*

- $S^M_{ij}(G'_i, C_i) \leq S^M_{ij}(G_i, C_i)$, *and*
- $S^M_{ik}(G'_i, C_i) \geq S^M_{ik}(G_i, C_i) \; \forall k \in C_i \setminus \{j\}$.[4]

5.3.3 The Basic vs. the Drop-Edge Mechanism

In this section, we first present a straw-man mechanism called the BASIC mechanism, first introduced by [68]. We show that the BASIC mechanism can easily be manipulated via misreports and then introduce the DROP-EDGE mechanism that removes the incentive to misreport.

[4] Note that the first requirement is equivalent to *value-strategyproofness* as defined for trust mechanisms, and both requirements together imply *rank-strategyproofness* [16].

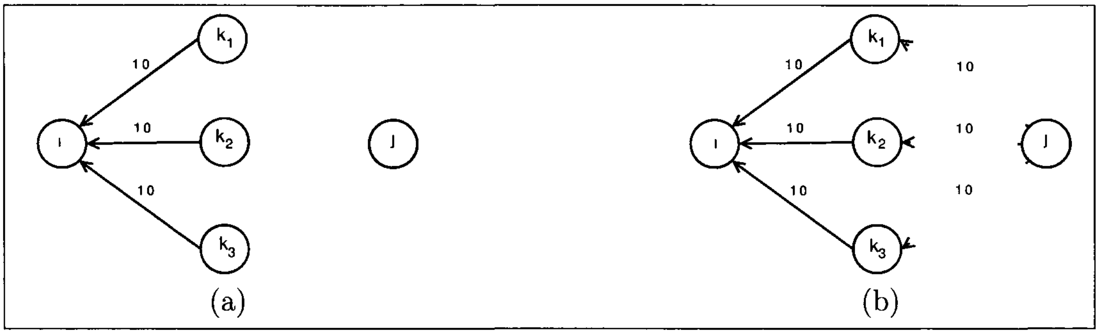


Figure 5 3 (a) A work graph based on true reports (b) The subjective work graph as viewed by i, including a misreport attack by j to boost its score in BarterCast Dotted edges indicate misreports

Definition 18 ***(Basic Mechanism)*** *Given subjective work graph G_i and choice set C_i, construct a modified graph $G_i^B = (V_i, E_i, w_i^B)$ with weights defined as*

$$\forall (j,k) | i \in \{j,k\} \quad w_i^B(j,k) = w_i^i(j,k)$$

$$\forall (j,k) | i \notin \{j,k\} \quad w_i^B(j,k) = \max\{w_i^j(j,k), w_i^k(j,k)\},$$

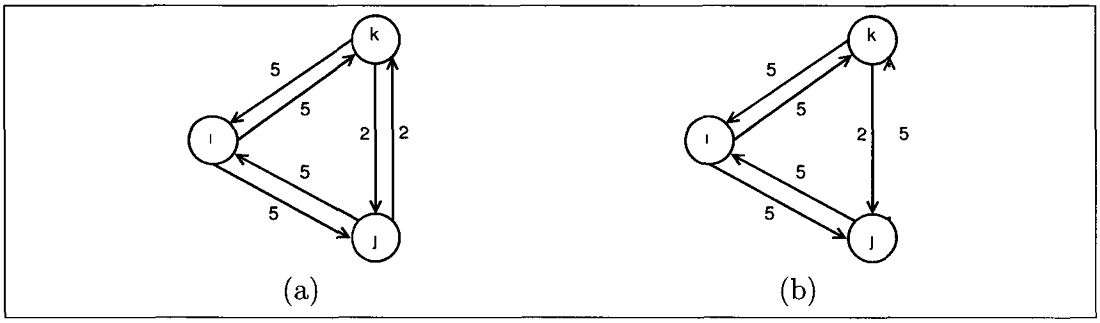
where missing reports in the max-operator are set equal to 0 Let $MF_{G_i^B}(i,j)$ denote the maximum flow from i to j in G_i^B Define the Basic *Score of agent j as $S_{ij}^B(G_i, C_i) = MF_{G_i^B}(j,i) - MF_{G_i^B}(i,j)$* [5]

In the Basic mechanism, an agent takes its own information over reports from others (1) Given two reports, it takes the maximum of the two (2) Note that even if no agents misreport, two reports for the same edge will generally be in conflict when a decentralized mechanism is being used By taking the maximum of the two reports, an agent always uses the most up-to-date information (in the case of non-strategic reports) The motivation for using the max-flow algorithm is that it bounds the

[5] The specification of the Basic mechanism here differs from the one presented in Meulpolder et al [68] only in that they take the arctan of the difference between the flows However, because arctan is a monotonic function this does not change the relative scores of the agents

influence of any report that agent j can make by the edges between i and j, preventing an agent from grossly inflating the work it has performed for another agent In Section 5 4 3, we discuss in more detail how this limits the power of strategic manipulations and also protects against Byzantine attacks (i e , arbitrary attacks not necessarily originating from rational agents)

It is easy to see that the Basic mechanism can be manipulated via misreports We illustrate two attacks in Figures 5 3 and 5 4 We always show the subjective work graph from i's perspective and the manipulating agent is j Figure 5 3 (a) shows a true work graph Figure 5 3 (b) shows agent i's view of the work graph, now including a misreport by agent j Agent j has simply reported that it has done work for k_1, k_2, and k_3, although it did not The Basic mechanism does not catch this because there never was an interaction there are no reports from these other agents Note that agent j increased its score from 0 to 30 via this attack Now consider Figure 5 4 (a) which shows a new true work graph Figure 5 3 (b) shows a misreport manipulation by agent j where j reported that it has done 5 units of work for k even though it only did 2 units of work Because the Basic mechanism takes the maximum of two reports, agent i will believe j's report As a result, agent k's score has decreased from 0 to -3, and agent j's score has increased from 0 to 3 Note that simply replacing the max-operator with the min-operator in the definition of the Basic mechanism does *not* make it misreport-proof Consider again Figure 5 4 (b) Using the min-operator prevents the attack where agent j exaggerates the amount of work he has done for agent k However, now agent j can misreport the amount of work that k has done for j, for example it can report 0 instead of 2 Now, if the Basic mechanism took the



(a)

(b)

Figure 5.4: (a) A Work graph based on true reports. (b) The subjective work graph as viewed by i, including a misreport attack by j to decrease k's score and increase its own score when the BASIC mechanism is used.

minimum instead of the maximum of two conflicting reports, the effect would be that after this particular attack, k score would decrease by 2 and j's score would increase by 2. Whether max or min is used, in both variants of the BASIC mechanism it is a dominant strategy to always report ∞ work performed, and 0 work consumed. Thus, a more sophisticated mechanism is necessary to defend against misreport attacks.

The DROP-EDGE mechanism ignores some of the information available to an agent, depending on context. Here, the "context" is the agent's current choice set C_i. If the agent ignores the reports from all agents currently inside the choice set, the resulting mechanism becomes misreport-proof.

Definition 19 ***(Drop-Edge Mechanism)*** *Given subjective work graph G_i and choice set C_i, construct the modified graph $G_i^D = (V_i, E_i, w_i^D)$ with the weights w_i^D defined as:*

$$\forall (j,k) | i \in \{j,k\} \quad w_i^D(j,k) = w_i^i(j,k) \tag{5 1}$$

$$\forall (j,k) | j,k \in C_i \quad w_i^D(j,k) = 0 \tag{5 2}$$

$$\forall (j,k) | j \in C_i, k \notin C_i \quad w_i^D(j,k) = w_i^k(j,k) \tag{5 3}$$

$$\forall (j,k) | k \in C_i, j \notin C_i \quad w_i^D(j,k) = w_i^j(j,k) \tag{5 4}$$

$$\forall (j,k) | j,k \notin C_i, i \notin \{j,k\} \; w_i^D(j,k) = \max\{w_i^j(j,k), w_i^k(j,k)\} \tag{5 5}$$

Missing reports in the max-operator are set to 0 *Agent* j*'s score is* $S_{ij}^D(G_i, C_i) = MF_{G_i^D}(j,i) - MF_{G_i^D}(i,j)$ [6]

An agent takes its own information/experience over reports of others (5 1) Lines (5 2)-(5 4) implement the "edge-dropping" idea Any reports received by agent i from agents in the choice set C_i are dropped in determining edge weights in modified graph G_i^D An edge (j,k) is dropped completely if both j and k are inside C_i (5 4) In the case of two conflicting reports by two agents outside the choice set, the mechanism takes the maximum 5 5, thereby always using the most up-to-date information available For an illustration of the DROP-EDGE mechanism see Figure 5 5

We make the following simple observation

Proposition 6 *Drop-Edge is misreport-proof*

Proof No report of agent j is used in i's decision making process whenever agent j is in the choice set of agent i □

[6] We do not need the max-flow algorithm to obtain misreport-proofness However, we use it because it provides some additional protection against sybil and Byzantine attacks as we discuss in more detail in Section 5 4 3, and also makes the comparison with BASIC easier

Note that both the BASIC mechanism and the DROP-EDGE mechanism need to compute the maximum flow on a large work graph to determine the scores In practice, however, running the max-flow algorithm on the full work graph may take too long due to the computational complexity of the max-flow algorithm If we assume that the work graphs has at least as many edges as nodes, then all known algorithms have a running time that is at least quadratic in the number of nodes (see Goldberg et al [39] for an overview of max-flow algorithms) The only exception is the Ford-Fulkerson algorithm which has a running time of $\mathcal{O}(Ef)$ when edge weights are integral, where E is the number of edges and f is the largest flow in the network However, most work graphs in practice will have a relatively large number of edges because the applications that are suitable for accounting mechanisms generally lead to many short-term interactions between many agents Thus, for graphs with thousands or millions of agents, even a running time of $\mathcal{O}(Ef)$ is prohibitive On the other hand, if the distributed work graph is relatively dense, then most nodes are connected via short paths, and it may be sufficient to run a max-flow algorithm that only considers paths of a restricted length, and such algorithms run much faster in practice Indeed, Piatek et al [75] have found empirically, that 99% of the agents in a P2P file sharing networks are connected via paths of length 1 or 2 For the experimental results we present in Section 5 5 we have used mechanisms with max-flow restricted to at most 1 hop (i e , paths of length at most 2), and for the simulations we present in Section 5 6 we have used mechanism with max-flow restricted to at most 2 hops

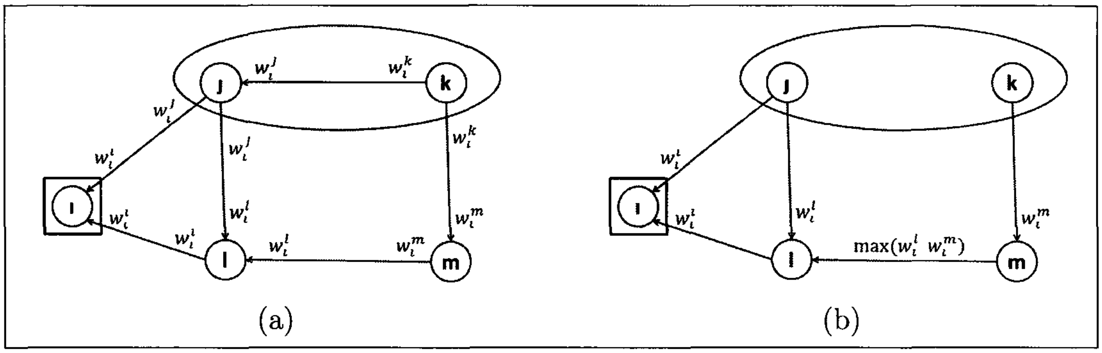


Figure 5 5 An illustration of the DROP-EDGE mechanism We are showing subjective work graphs from agent i's perspective The choice set is $C_i = \{j, k\}$ (a) Agent i's subjective work graph where each edge has two weights, one from each agent who knows about that edge (b) Agent i's subjective work graph after the DROP-EDGE mechanism has been applied

5.4 Theoretical Analysis

5.4.1 Information Loss of Drop-Edge

In the last section we have shown that by dropping some of the information based on context, the resulting mechanism becomes misreport-proof However, this obviously comes at a cost because having more information about the past actions of other agents generally helps to better discriminate between cooperative and free-riding agents We are interested in this trade-off between *informativeness* on the one side, and *misreport-proofness* on the other side In this section we analyze the information loss of DROP-EDGE due to the discarded edges and show that it is small and vanishes in the limit as the number of agents in the network grows Thus, the misreport-proofness of DROP-EDGE comes at a relatively small cost Without misreport-proofness, strategic manipulations introduce an additional cost

The following analysis is based on agent i's subjective work graph $G_i = (V_i, E_i, w_i)$

To isolate the question of "information loss" due to dropping some of the information, we only consider centralized protocols where all agents make reports to a centralized entity after ever time step, and we assume that agents do not perform any kind of manipulations We let $G_i^D = (V_i, E_i, w_i^D)$ denote the modified graph after the Drop-Edge mechanism has been applied to G_i Note that which edges are dropped in G_i^D depends on which particular choice set C_i is chosen Analogously, $G_i^B = (V_i, E_i, w_i^B)$ denotes the modified graph after the Basic mechanism has been applied to G_i Note that using a centralized information exchange protocol and assuming no manipulations, there won't be any conflicting reports in the subjective work graphs, and thus G_i^B represents the true (omniscient) work graph

As a first step, we study the result of dropping edge on the net work information contained in the work graphs Later, we add the use of max-flow to the analysis For graph $G_i^D = (V_i, E_i, w_i^D)$, we define the net work on edge (k, j) as $\tilde{w}_i^D(k, j) = w_i^D(k, j) - w_i^D(j, k)$ so that k's overall net work from i's perspective is $work_i(k, G_i^D) = \sum_{j \neq k} \tilde{w}_i^D(k, j)$ Analogously, for graph $G_i^B = (V_i, E_i, w_i^B)$, we let $\tilde{w}_i^B(k, j) = w_i^B(k, j) - w_i^B(j, k)$ and $work_i(k, G_i^B) = \sum_{j \neq k} \tilde{w}_i^B(k, j)$ Thus, the term $work_i(k, G_i^B)$ represents agent k's *true* net work, and $work_i(k, G_i^D)$ represents Drop-Edge's approximation of agent k's net work, both according to i's subjective work graph

Theorem 8 *For all subjective work graphs $G_i = (V_i, E_i, w_i)$ with $|V_i| = n$, for all $k \in V_i$, for all choice sets C_i chosen uniformly at random with $|C_i| = m$ and $k \in C_i$*

$$\frac{\mathbb{E}_{C_i}[work_i(k, G_i^D)]}{work_i(k, G_i^B)} = 1 - \frac{(m-1)}{(n-1)}$$

Proof

$$
\begin{aligned}
&\mathbb{E}_{C_i}[work_i(k, G_i^D)] && \qquad (5\ 6)\\
&= \mathbb{E}_{C_i}[\sum_{j \neq k} \tilde{w}_i^D(k, j)] && \qquad (5\ 7)\\
&= \sum_{j \neq k} \mathbb{E}_{C_i}[\tilde{w}_i^D(k, j)] && \qquad (5\ 8)\\
&= \sum_{j \neq k} \Big[\frac{(m-1)}{(n-1)}\ 0 + \big(1 - \frac{(m-1)}{(n-1)}\big)\ \tilde{w}_i^B(k, j)\Big] && \qquad (5\ 9)\\
&= \big(1 - \frac{(m-1)}{(n-1)}\big)\ work_i(k, G_i^B)
\end{aligned}
$$

For equation (5 9), consider edge (k, j) Because C_i is chosen uniformly at random with $k \in C_i$, the probability that j is also inside any random C is $\frac{m-1}{n-1}$ If k and j are inside C_i the edge gets dropped, otherwise $\tilde{w}_i^B(k, j)$ is counted □

Theorem 8 implies that if n is relatively large compared to m, then the expected net work computed by the Drop-Edge Mechanism is very close to the true net work

Corollary 3 *For all subjective work graphs $G_i = (V_i, E_i, w_i)$ with $|V_i| = n$, for all $k \in V$, for choice sets C_i chosen uniformly at random with $|C_i| = m$, it holds that*

$$\lim_{\frac{n}{m} \to \infty} \frac{\mathbb{E}_{C_i}[work_i(k, G_i^D)]}{work_i(k, G_i^B)} = 1$$

We now turn our attention to the approximation ratio of the scores computed by Drop-Edge when the max-flow algorithm is used The first theorem is with regard to running the full max-flow algorithm, however, for the analysis, we need to consider max-flows restricted to a certain number of hops We let $MF_G^h(i, j)$ denote the max-flow from node i to j in graph G with exactly h hops We let

$S^{D,h}_{ij}(G_i, C_i)$ denote the score computed by Drop-Edge for h hops, i e , $S^{D,h}_{ij}(G_i, C_i) = MF^h_{G^D_i}(j, i) - MF^h_{G^D_i}(i, j)$ Analogously, $S^{B,h}_{ij}(G_i, C_i)$ is the score computed by the BASIC mechanism using max-flow with exactly h hops

Theorem 9 *For all subjective work graphs $G_i = (V_i, E_i, w_i)$ with $|V_i| = n$, for all $k \in V_i$, for i's choice set C_i chosen uniformly at random with $|C_i| = m$ and $k \in C_i$*

$$E_{C_i}[S^D_{ik}(G_i, C_i)] = S^{B,0}_{ik}(G_i, C_i) + \sum_{h=1}^{n-m-1} \prod_{p=1}^{h} (\frac{n-m-p}{n-1-p}) \;\; S^{B,h}_{ik}(G_i, C_i)$$

Proof

$$E_{C_i}[S^D_{ik}(G_i, C)] = E_{C_i}[MF_{G^D_i}(k, i) - MF_{G^D_i}(i, k)] \tag{5 10}$$

$$= \sum_{h=0}^{n-m-1} E_{C_i}[MF^h_{G^D_i}(k, i) - MF^h_{G^D_i}(i, k)] \tag{5 11}$$

$$= E_{C_i}[MF^0_{G^D_i}(k, i) - MF^0_{G^D_i}(i, k)] \tag{5 12}$$

$$+ \sum_{h=1}^{n-m-1} E[MF^h_{G^D_i}(k, i) - MF^h_{G^D_i}(i, k)] \tag{5 13}$$

$$= S^{B,0}_{ik}(G_i, C_i) \tag{5 14}$$

$$+ E_{C_i}[MF^1_{G^D_i}(k, i) - MF^1_{G^D_i}(i, k)] \tag{5 15}$$

$$+ \sum_{h=2}^{n-m-1} E_{C_i}[MF^h_{G^D_i}(k, i) - MF^h_{G^D_i}(i, k)] \tag{5 16}$$

$$= S^{B,0}_{ik}(G_i, C_i) + (\frac{n-m-1}{n-2}) \;\; S^{B,1}_{ik}(G_i, C_i) \tag{5 17}$$

$$+ (\frac{n-m-1}{n-2}) \;\; (\frac{n-m-2}{n-3}) \;\; S^{B,2}_{ik}(G_i, C_i) \tag{5 18}$$

$$+ \sum_{h=3}^{n-m-1} E_{C_i}[MF^h_{G^D_i}(k, i) - MF^h_{G^D_i}(i, k)] \tag{5 19}$$

$$= S^{B,0}_{ik}(G_i, C_i) + \sum_{h=1}^{n-m-1} \prod_{p=1}^{h} (\frac{n-m-p}{n-1-p}) \;\; S^{B,h}_{ik}(G_i, C_i) \tag{5 20}$$

In Equation (5 12) we isolated the expectation of the 0-hop max-flow terms which consider the direct paths between i and k and thus do not involve dropped edges, and consequently Equation (5 14) follows because $S_{ik}^{D,0} = S_{ik}^{B,0}$ In Equation (5 15) we isolated the expectation of the 1-hop max-flow terms, i e , the flows along all paths of length 2 between i and k Because C_i was chosen uniformly at random, for any of the 1-hop paths between i and k the probability that the intermediate node lies outside of C_i is $\frac{n-m-1}{n-2}$ and Equation (5 17) follows The final expression follows from analogous reasoning for all h-hop max-flows □

Remember that running the full max-flow algorithm may be computationally prohibitive, which is why we use max-flow algorithms restricted to at most 1 or 2 hops in our experiments We let $S_{ik}^{D,\leq 1}$ denote the scores obtained by the Drop-Edge mechanism when max-flow is restricted to at most 1 hop, and $S_{ik}^{B,\leq 1}$ analogously for the Basic mechanism The following corollary tells us the accounting accuracy of Drop-Edge using a max-flow algorithm restricted to at most 1 hop

Corollary 4 *If the max-flow algorithm is restricted to a most 1 hop, then for all subjective work graphs $G_i = (V_i, E_i, w_i)$ with $|V_i| = n$, for all $k \in V_i$, for i's choice set C_i chosen uniformly at random with $|C_i| = m$ and $k \in C_i$*

$$\frac{E_{C_i}[S_{ik}^{D,\leq 1}(G_i, C_i)]}{S_{ik}^{B,\leq 1}(G_i, C_i)} \geq \left(\frac{n-m-1}{n-2}\right)$$

The theoretical results in this section bound the accuracy in expectation over choice sets and do not directly pertain to accuracy with respect to selecting the right agent from a given choice set Moreover, the approximation ratios for the max-flow based mechanism do not directly compare the scores obtained when using max-flow to the scores obtained if the net work were considered directly In Sections 5 5 and 5 6 we see that the information-loss of DROP-EDGE is indeed small in practice, even for small graphs, and that the mechanism leads to very good efficiency

5.4.2 Sybil-Proofness

Now we turn our attention to a class of attacks called *sybil attacks*, where an agent introduces sybil nodes (fake agents) into the network to manipulate the accounting mechanism

Preliminaries

We distinguish between *passive* sybil attacks, where the sybils themselves may consume but not perform work, and *active* sybil attacks, where the sybils themselves also perform work [7]

Definition 20 ***(Passive Sybil Attack)*** *A passive sybil attack by agent* j *is a tuple* $\sigma_j = (V_s, E_s, w_s)$ *where* $V_s = \{s_{j1}, s_{j2}, \}$ *is a set of sybils,* $E_s = \{(x, y) \quad x, y \in S \cup \{j\}\}$*, and* w_s *are the edge weights for the edges in* E_s *Applying the sybil attack* σ_j *to agent* i*'s subjective work graph* $G_i = (V_i, E_i, w_i)$ *results in a modified work graph* $G_i \downarrow \sigma_j = G_i' = (V_i \cup V_s, E_i \cup E_s, w')$ *where* $w'(e) = w_i(e)$ *for* $e \in E_i$ *and* $w'(e) = w_s(e)$ *for* $e \in E_s$

Definition 21 ***(Active Sybil Attack)*** *An active sybil attack by agent* j *is a tuple* $\sigma_j = (V_s, E_s, E_s^{\rightarrow}, w_s, w_s^{\rightarrow})$ *where* $V_s = \{s_{j1}, s_{j2}, \}$ *is a set of sybils,* $E_s = \{(x, y)$

[7] Note that our definitions differ from previous definitions of sybil attacks on reputation mechanisms (see [15] and [16]), in particular the one for active sybil attacks where we also allow sybil agents to perform work

$x, y \in V_s \cup \{j\}\}$, $E_s^{\rightarrow} = \{(x, y) \quad x \in V_s, y \notin V_s \cup \{j\}\}$ are edges indicating work done by the sybils, w_s are the edge weights for the edges in E_s, and $w_s^{\rightarrow}$ are the edge weights for edges in $E_s^{\rightarrow}$ Applying the sybil attack σ_j to agent i's subjective work graph $G_i = (V_i, E_i, w_i)$ results in a modified work graph $G_i \downarrow \sigma_j = G_i' = (V_i \cup V_s, E_i \cup E_s \cup E_s^{\rightarrow}, w')$ where $w'(e) = w_i(e)$ for $e \in E_i$ and $w'(e) = w_s(e)$ for $e \in E_s$ and $w'(e) = w_s^{\rightarrow}(e)$ for $e \in E_s^{\rightarrow}$

So far we have only defined the strategy space for passive and active sybil attacks Such an attack can only be "beneficial" if the attacking agent is better of after the attack than before Remember that S_0^M denotes the default score that accounting mechanism M assigns to an agent about which no information is available

Definition 22 ***(Beneficial Sybil Attack)*** *Given accounting mechanism M and work graph $G_i = (V_i, E_i, w_i)$, a beneficial (passive or active) sybil attack σ_j by agent $j \in V_i$ such that $G_i' = \sigma_j(G_i)$ is one where option (1), (2), or (3) holds*

(1) $\exists C_i$ s t $j \in C_i$ and $S_{ij}^M(G_i, C_i) < S_{ij}^M(G_i', C_i)$

(2) $\exists k \in V_i \backslash \{j\}$ and $\exists C_i$ with $j, k \in C_i$ $(S_{ij}^M(G_i, C_i) < S_{ik}^M(G_i, C_i)) \quad \wedge \quad (S_{ij}^M(G_i', C_i) > S_{ik}^M(G_i', C_i))$

(3) $\exists s \in V_s$ $(\exists C_i$ with $s \in C_i$ $S_{is}^M(G_i', C_i) > S_0^M) \quad \wedge \quad (\forall C_i$ with $j \in C_i$ $S_{ij}^M(G_i', C_i) \geq S_{ij}^M(G_i, C_i))$

i e , agent j can (1) increase its own score, or (2) affect its own score and that of another agent in such a way that the relative ranking of the two agents changes, or (3) create a sybil agent with a score strictly higher than S_0^M without decreasing its own score

To study the sybil attacks as introduced so far, we do not need a dynamic, multi-step analysis. Given work graph G_i, an attacking agent can do multiple things, e.g., add sybils to the network, make multiple false reports about these sybils, etc. We model all of this as happening in one step, inducing a new subjective work graph G'_i. However, when we consider long-term effects of an attack, we also have to look at what happens when certain attacks are repeated over and over again. How beneficial a sybil attack really is, depends on the trade-off between the amount of work necessary to perform the attack, and the amount of "free" work the agent can consume as a result of the attack. We will distinguish between *long-term* beneficial sybil attacks on the one side, where the ratio between work performed and consumed goes towards infinity as the attack is repeated, and *short-term* beneficial sybil attacks on the other side, where that ratio is bounded by a constant.

Definition 23 ***(Long-term vs. Short-term Beneficial Sybil Attacks)*** *Given accounting mechanism M and work graph $G_i = (V_i, E_i, w_i)$, assume agent $j \in V_i$ performs a (passive or active) sybil attack σ_j such that $G'_i = \sigma_j(G_i)$. Let σ_j^n denote an $n-$times-repetition of the sybil attack. Let $\omega^-(\sigma_j^n)$ denote the amount of work involved in performing σ_j^n, and let $\omega^+(\sigma_j^n)$ denote the amount of work that agent j or any of its sybils will be able to consume. We call σ_j a*

- ***long-term beneficial sybil attack*** *if $\omega^+(\sigma_j^n) > 0$ and $\omega^-(\sigma_j^n) = 0$ or $\lim_{n\to\infty} \frac{\omega^+(\sigma_j^n)}{\omega^-(\sigma_j^n)} = \infty$*
- ***short-term beneficial sybil attack*** *if $\omega^+(\sigma_j^n) > 0$ and $\omega^-(\sigma_j^n) > 0$ and $\exists c \in \mathbb{R}_{\geq 0}$: $\lim_{n\to\infty} \frac{\omega^+(\sigma_j^n)}{\omega^-(\sigma_j^n)} \leq c$*

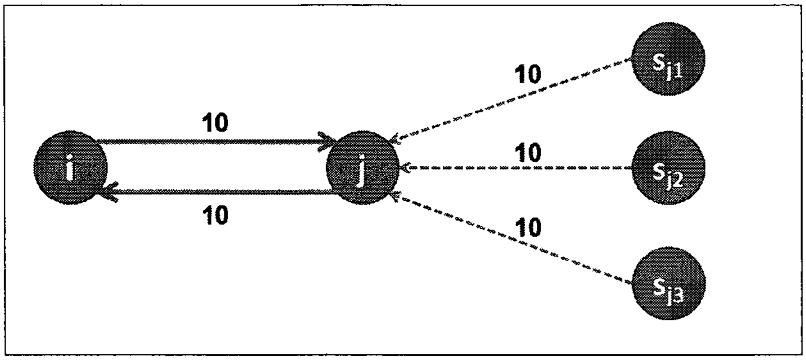


Figure 5 6 A sybil attack where agent j generates many sybils, then does a little bit of work for i and then provides its sybils with positive scores

A long-term beneficial (passive) sybil attack on the Basic and Drop-Edge mechanisms is illustrated in Figure 5 6 where condition (3) of Definition 22 holds In this example, agent j has already performed/consumed 10 units of work for/from agent i (such that agent i believes agent j's reports about other agents) To perform the sybil attack, agent j creates a set of sybils and falsely reports to i that these sybils have performed 10 units of work for j, such that i now assigns a score strictly higher than S_0^M to the sybil nodes, in particular, both the Basic and Drop-Edge mechanisms assign a score of 10 to the sybil nodes Each sybil agent can now exploit its score and consume some work from i (assuming, at some point, the sybils will be in i's choice set with other agents that have a lower score) Once the sybils' scores are "used up", j can simply create another sybil s_j' and repeat the attack ad infinitum This attack is powerful because it only requires a *passive* sybil attack that involves no work to be performed by j or any of its sybils In general, however, passive attacks may be long-term or short-term beneficial, and active attacks may also be long-term or short-term beneficial

Definition 24 ***(Sybil-Proofness)*** *An accounting mechanism M is*

- *sybil-proof against long-term beneficial sybil attacks, if for every work graph G_i there exists no (passive or active) long-term beneficial sybil attack on M with respect to G_i*
- *sybil-proof against short-term beneficial sybil attacks, if for every work graph G_i there exists no (passive or active) short-term beneficial sybil attack on M with respect to G_i*

Before we can formally prove our first impossibility theorem regarding sybil-proofness, we introduce a series of natural assumptions regarding accounting mechanisms First, we assume that the scores an accounting mechanism computes only depend on the amount of work performed and consumed by the agents in the network More specifically, we assume that adding or removing agents with no amount of work consumed or performed does not change the scores of other agents More formally

Definition 25 ***(Independence of Disconnected Agents)*** *An accounting mechanism M satisfies independence of disconnected agents, if for any subjective work graph $G_i = (V_i, E_i, w_i)$ and any choice set C_i, for any $k \in V_i$ for which there does not exist an edge in E_i or for which all edges in E_i have zero weight, where $G_i' = (V_i', E_i', w_i')$ denotes the graph where node k has been removed, i e , $V_i' = V_i \backslash \{k\}$, $E_i' = E_i \setminus \{(x,y) \quad x = k \vee y = k\}$, and $w_i'(e) = w_i(e)$ for all $e \in E_i'$, the following holds*

$$\forall j \in V_i' \quad S_{ij}^M(G_i, C_i) = S_{ij}^M(G_i', C_i')$$

Furthermore, we will assume that a priori, the accounting mechanism does not put more or less trust into any agent in the network More formally, we only consider mechanisms that, for any renaming of the agents in the network, return the same scores, i e , they are *symmetric*[8]

Definition 26 ***(Symmetric Accounting Mechanisms)*** *An accounting mechanism M is symmetric if, for any node i with any any subjective work graph $G_i = (V_i, E_i, w_i)$ and choice set C_i, any graph isomorphism f such that $G_i' = f(G_i)$, $C_i' = f(C_i)$ and $f(i) = i$*

$$\forall j \in V_i \setminus \{i\} \quad S_{ij}^M(G_i, C_i) = S_{if(j)}^M(G_i', C_i')$$

For the design of sybil-proof accounting mechanisms, we want to exclude any "trivial" accounting mechanisms that assign the same or random scores to every agent, as well as mechanisms that ignore all information except for their own direct experiences Assuming *single-report responsiveness* excludes these mechanisms

Definition 27 ***(Single-Report Responsiveness Property)*** *Let $dist(i, j)$ denote the length of the shortest path between i and j An accounting mechanism M has the single-report responsiveness property if, for any agent i, there exists a subjective work graph $G_i = (V_i, E_i, w_i)$ and choice set C_i, with nodes j and k such that $dist(i, j) = dist(j, i) = 1$ and $dist(i, k) = dist(k, i) = \infty$ (i e , nodes i and j are neighbors in G_i and no path is connecting nodes i and k), and there exists a graph $G_i' = (V_i', E_i', w_i')$ with $V_i' = V_i$, $E_i' = E_i \cup \{(k, j), (j, k)\}$, and $w_i'(e) = w_i(e)$ for all $e \in E_i \setminus \{(k, j), (j, k)\}$,*

[8] Note that in the context of reputation mechanisms, *symmetry* typically corresponds to globally consistent, or *objective* reputation values, where every agent in a network has the same view on each other agent's reputation, in contrast to *asymmetric* mechanisms that allow for *subjective* reputation values Cheng et al [15] have shown that no symmetric and sybil-proof reputation mechanisms exists According to Cheng et al 's definition, our accounting mechanisms would all be called "asymmetric" because they all inherently lead to subjective scores because the scores are computed based on subjective work graphs However, what we mean by "symmetry" is something different, namely that from each individual agent's perspective, the rest of the network is symmetric

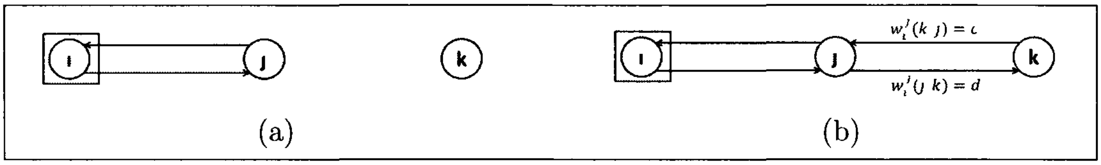


Figure 5 7 An illustration of the single-report-responsiveness property there exists a subjective work graph G_i, e g , the one show in (a), such that a single positive report by j about k, as shown in (b), leads to $S_{ik}^M(G_i, C_i) > S_0^M$, and a single negative report by j about k leads to $S_{ik}^M(G_i, C_i) < S_0^M$

and there exists a constant $c \in \mathbb{R}_{>0}$ *with* $w_i'^j(k, j) = c$, *such that*

$$S_k^M(G_i', C_i') > S_0^M$$

and analogously, a constant $d \in \mathbb{R}_{>0}$ *with* $w_i'^j(j, k) = d$, *such that*

$$S_k^M(G_i', C_i') < S_0^M$$

An illustration of the single-report responsiveness property is depicted in Figure 5 7 What this property says is that there exists a situation (i e , a specific work graph), where i has no information about k, and a *single* positive report by agent j about agent k can increase the score that agent i assigns to agent k above ${}_0^M$, and that a *single* negative report by agent j about agent k can decrease the score that agent i assigns to agent k below S_0^M Thus, if an agent was previously unknown to me, a single positive or negative report about that agent can potentially change my evaluation of that agent

Impossibility of Sybil-Proofness

Note that both the Basic and the Drop-Edge mechanism satisfy the independence of disconnected agents, are symmetric, and satisfy the single-report respon-

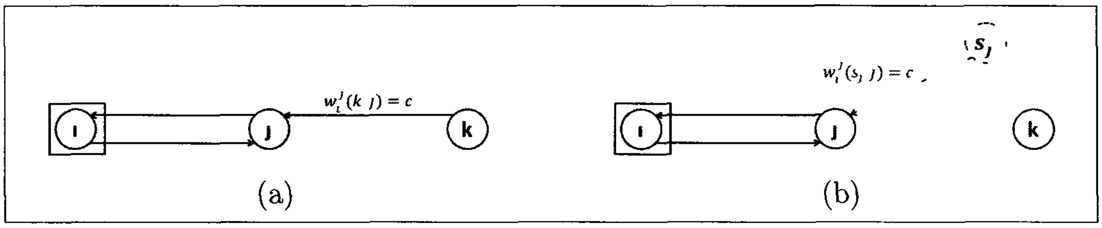


Figure 5 8 An illustration of the long-term beneficial sybil attack used in the proof for Theorem 10 If the mechanism is misreport-proof, then j has no disadvantage from making a truthful report $w_i^j(k, j) = c$ about another agent k If the mechanism is also symmetric and satisfies independence of disconnected agents, then j can create sybil s_j, and make a report $w_i^j(s_j, j) = c$ about s_j, also without a disadvantage to j Thus, if originally, the positive report about k lead to a positive score for k, then now the sybil node s_j has a positive score as well, and the attack does not require any actual work to be performed by j

siveness property We have already shown (Figure 5 6) that both mechanisms are susceptible to sybil attacks We will now show that this is generally unavoidable

Theorem 10 *For every accounting mechanism M that satisfies independence of disconnected agents, is symmetric, has the single-report responsiveness property, and is misreport-proof, there exists a (passive) long-term beneficial sybil attack*

Proof Let's assume accounting mechanism M satisfies the single-report responsiveness property Thus, there exists a graph G_i and nodes i, j and k as described in Definition 27, for example like the one depicted in Figure 5 8 (a) Now, let agent j create a sybil node s_j and insert it into G_i such that $G_i' = (V_i', E_i, w_i)$ with $V_i' = V_i \cup \{s_j\}$ Because of the independence of disconnected agents, the scores of all agents in the graph have remained the same Note that there is no path connecting k and i as well as no path connecting s_j and i, and thus the two nodes k and s_j look the same from i's perspective Now, assume that agent k performs c units of work for j (as needed for the single-report responsiveness property) and agent j makes a truthful report to i about this interaction, leading to subjective work graph G''_i such that $S^M_{ik}(G''_i, C_i) > S^M_0$. Here it is essential that M is a misreport-proof mechanism. Because M is misreport-proof, we know that agent j has no disadvantage from reporting truthfully, i.e., for any possible misreport that would lead to subjective work graph G'''_i it holds that $\forall C_i$ with $j \in C_i$: $S^M_{ij}(G''_i, C_i) \geq S^M_{ij}(G'''_i, C_i)$. Because M is symmetric, we can apply a graph isomorphism f to G''_i that only switches the labeling of s_j and k but nothing else. Thus, there exists a report that j can make about s_j with $w'^{j}_i(s_j, j) = c$ leading to graph G^*_i such that $S^M_{is_j}(G^*_i, C_i) > S^M_0$ (see Figure 5.8 (b)). As before with node k, because of misreport-proofness, we know that agent j has no disadvantage from making this report. Thus, property (3) of Definition 22 is satisfied and because the attack itself involves no work, this constitutes a long-term beneficial sybil attack on M. □

(Im-)Possibility of K-Sybil-Proofness

In this section we explore whether we can achieve any kind of formal sybil-proofness guarantees, despite the strong negative results from the last section. The only property that we can reasonably relax for the design of useful accounting mechanisms is the single-report responsiveness property. We can conceive of mechanisms that require two, or more generally, K, positive or negative reports about an agent, before the mechanism assigns a score distinct from S^M_0 to that agent. This leads to the following generalization of the responsiveness property:

Definition 28 ***(K-Report Responsiveness Property)*** *Let $dist(i, j)$ denote the distance between two nodes i and j in a graph. An accounting mechanism M has* *the K-report responsiveness property if, for any agent i, there exists a subjective work graph $G_i = (V_i, E_i, w_i)$ and choice set C_i, with node l and a set of nodes V_K with $|V_K| = K$, such that $\forall k \in V_K \quad dist(i,k) = dist(k,i) = 1$ and $dist(i,l) = dist(l,i) = \infty$ (i e , nodes i and all nodes in V_K are neighbors in G_i and no path is connecting i and l), and there exists a graph $G_i' = (V_i', E_i', w_i')$ with $V_i' = V_i$, $E_i' = E_i \cup \{(k,j),(j,k) | k \in V_K\}$, and $w_i'(e) = w_i(e)$ for all $e \in E_i \setminus \{(k,j),(j,k)|k \in V_K\}$, and there exists a constant $c \in \mathbb{R}_{>0}$ with $w_i'^{j}(k,j) = c$ for all $k \in V_K$, such that*

$$S_{ik}^M(G_i', C_i') > S_0^M$$

and a constant $d \in \mathbb{R}_{>0}$ with $w_i'^{j}(j,k) = d$ for all $k \in V_K$, such that

$$S_{ik}^M(G_i', C_i') < S_0^M$$

Obviously, performing a sybil attack against a mechanism that does not have the single-report responsiveness property, but the K-report responsiveness property is more difficult, and the attack would require additional work, either by the sybil agents or by the manipulating agent itself We can now define a corresponding, weaker notion of sybil-proofness

Definition 29 ***(K-Sybil-Proofness)*** *An accounting mechanism M is* K-Sybil-proof against long-term beneficial sybil attacks *if, for every work graph G_i, there does not exist a long-term beneficial sybil attack with K or fewer sybils for M, it is* K-Sybil-proof against short-term beneficial sybil attacks *if there does not exist a short-term beneficial sybil attack with K or fewer sybils for M*

Theorem 11 *No accounting mechanism that is K-report responsive, is symmetric,* *and satisfies independence of disconnected agents is K-sybil-proof against short-term beneficial sybil attacks*

Proof Let's assume accounting mechanism M is symmetric, satisfies independence of disconnected agents, and is K-report responsiveness Then there exists a subjective work graph G_i and nodes l and V_K as described in Definition 28 In particular, if all agents k in V_K make a report about the edge (k, l) with weight c to agent i, then the resulting score for agent l is greater than S_0^M, i e , $S_{il}^M(G_i', C_i') > S_0^M$ Now, let's remove one agent k^* from the set V_K, leading to subjective work graph G_i'' Now, $S_{il}^M(G_i'', C_i'') = S_0^M$ again Now we let agent j create a sybil agent s_j Because of the independence of disconnected agents, this does not change any of the scores Now assume that agent s_j performs the same number of units of work for i that previously k^* had performed Now, from i's perspective agents k^* and s_j look the same Thus, because M is symmetric, if agent s_j now makes a report about edge (l, s_j) with weight c, then $S_{il}^M(G_i''', C_i''') > S_0^M$ Without loss of generality, we can assume that $l = j$ Thus, by creating just one sybil s_j agent j has performed a short-term beneficial sybil attack □

We will now show how to turn any accounting mechanism into a K-report responsive mechanism that is K-sybil-proof against long-term beneficial sybil attacks

Definition 30 ***(K-Elimination-Wrapper)*** *A K-Elimination-Wrapper W takes as input an accounting mechanism M, a subjective work graph $G_i = (V_i, E_i, w_i)$, and a choice set C_i, and determines the scores $S_{ij}^W(M, G_i, C_i)$ for each agent $j \in C_i$, as viewed by agent i Let $\mathcal{P}(V_i)$ denote the powerset of V_i, and let $\mathcal{P}_{\leq K}(V_i)$ denote the set of subsets of $\mathcal{P}(V_i)$ of cardinality less than or equal to K We let $G_i \setminus X$ denote*

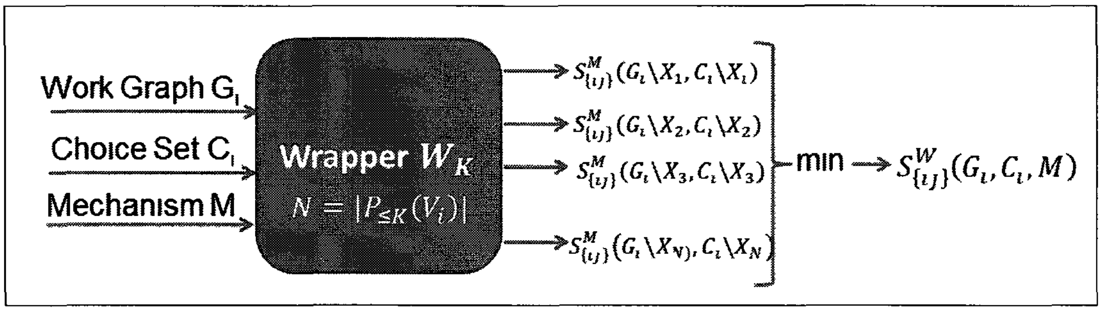

Work Graph G$_i$
Choice Set C$_i$
Mechanism M
Wrapper W_K
$N = |P_{\leq K}(V_i)|$
$S^M_{\{ij\}}(G_i\backslash X_1, C_i\backslash X_1)$
$S^M_{\{ij\}}(G_i\backslash X_2, C_i\backslash X_2)$
$S^M_{\{ij\}}(G_i\backslash X_3, C_i\backslash X_3)$
$S^M_{\{ij\}}(G_i\backslash X_N, C_i\backslash X_N)$
min
$S^W_{\{ij\}}(G_i, C_i, M)$

Figure 5 9 The K-Elimination-Wrapper

the subjective work graph that results from taking G_i and removing all nodes in X from V_i As before, we let $S^M_{ij}(G_i, C_i)$ denote the scores according to the accounting mechanism M The wrapper scores $S^W_{ij}(M, G_i, C_i)$ are computed as follows

$$S^W_{ij}(M, G_i, C_i) = \min_{X \in \mathcal{P}_{\leq K}(V_i)} \{S^M_{ij}(G_i \setminus X, C_i \setminus X)\}$$

Theorem 12 *A K-elimination-wrapper applied to any accounting mechanism leads to an accounting mechanism that is K-sybil-proof against long-term beneficial sybil attacks*

Proof Let's assume this is not true, i e , there exists a sybil attack by some agent j that involve less than or equal to K sybils and is long-term beneficial Note that the K-elimination wrapper iteratively removes all subsets of agents of size K or less from G_i, computes all scores without those subsets, and ultimately takes the minimum Thus, in one of those iterations, all of j's K sybils will be removed from G_i, and the resulting score will be part of the overall minimization of the wrapper Thus, if agent j's score before the sybil attack was lower than before the sybil attack, then the wrapper will take the score from before the attack, rendering the sybil attack useless. This excludes options (1) and (3) from the set of beneficial sybil attacks (see Definition 22) where the goal of the sybil attack was to increase agent j's or one of the sybils' scores. This only leaves option (2), which requires an active sybil attack, where the sybil agents themselves perform work and make misreports, such that after the sybil attack, an agent k, i.e., one of the other agents in the network, now has a lower score than before, such that the relative ordering of j and k has changed. If this attack is indeed successful (i.e., j and k are inside the same choice set and now j gets allocated instead of k), then j gets to consume some units of work "for free". However, note that after consuming a certain amount of work, j's score is lowered again, and at some point, j's score will be lower than k's score again. Thus, now another sybil attack would be necessary, which again would require the sybil agents to perform work. Thus, the amount of free work resulting from this sybil attack is bounded: for every x units of "free" work, the sybil attack requires a certain fixed amount of work as well. Thus, the sybil attack can at best be short-term beneficial, but not long-term beneficial. □

Note that using the K-elimination-wrapper does not provide any robustness against short-term beneficial sybil attacks, and even achieving K-sybil-proofness against long-term beneficial sybil attacks comes at a cost: the resulting mechanism is only K-report-responsive and ignores a larger part of the available information compared to a single-report responsive mechanism. Assuming random interactions between peers, the probability of having K reports about an agent decreases exponentially in K. Thus, real-world system designers face an important trade-off between (limited) robustness against sybil attacks on the one side, and informativeness on the other side.

The decision regarding this trade-off can depend on many factors. For example, in some domains, creating one or two sybils may be relatively cheap, but creating more sybils could become very expensive (e.g., obtaining multiple IP addresses). If this is in fact the case, then a 3-sybil-proof mechanism might provide good robustness in that particular domain. Furthermore, in some domains the interactions between peers are not random, but highly clustered (e.g., in P2P file sharing communities with similar taste preferences). Thus, in these domains it might be reasonable to assume that each agent has an average of K reports about each other agent, and thus, even after applying a K-elimination-wrapper, the resulting scores will still be informative enough. In future work, we will analyze this trade-off in more detail (analytically and experimentally).

5.4.3 The Role of the Max-Flow Algorithm

We have shown that we cannot achieve fully sybil-proof accounting mechanisms, and even limited robustness comes at a high price. One interesting way to address this problem in practice is the application of the max-flow algorithm inside an accounting mechanism. In fact, both BarterCast [68] and Drop-Edge [96] use max-flow. Ideally we would like to do accounting via taking the total sum of work performed and subtracting the total sum of work consumed for each agent. By using the max-flow algorithm, we essentially do a form of "bounded addition" which obviously distorts the true net work measure without providing any additional formal guarantees. However, using max-flow provides additional robustness against sybil attacks in practice: max-flow bounds the influence of any agent by the total amount of work performed by that agent itself This limits the power of sybil attacks, making them more costly and thus less attractive for the attacking agent [9] Furthermore, max-flow is also useful to protect against Byzantine agents, i e , agents that try to harm the network or specific agents in the network For example, if a Byzantine agent reports that agent i has consumed 1,000,000 units of work from him, and if other agents believe this report, then agent i will be unable to receive any work from those agents in the future Using max-flow makes Byzantine attacks much more difficult and costly for the attacking agent, thereby effectively preventing them in practice

5.5 Experimental Analysis: Discrete Simulations

In this section, we compare the mechanisms empirically via a discrete, round-based simulation to understand the trade-offs that are made in the Basic and Drop-Edge mechanisms in practice Remember that we consider both centralized and decentralized information exchange protocols For the decentralized version, we simulate the BarterCast information exchange protocol Consequently, when referring to the centralized versions of the mechanisms, we write Basic and Drop-Edge as before, and when referring to the decentralized versions we write BarterCast-Basic and BarterCast-Drop-Edge Considering the centralized version of each mechanism helps isolate the effect of the message exchange protocol from the mechanism itself

[9] Note that instead of using max-flow, we could also use other graph-based algorithms The algorithm only needs to have two properties first, it needs to have the "bounding property" to limit the influence of any agent proportionally to how much that agent has contributed to the system so far Second, the algorithm must have the "transitive-trust" property (cf Friedman et al [32] or Tang et al [99]), i e , when agent i has performed some work for j and j has performed some work for k, then i should also trust agent k to some degree

5.5.1 Experimental Set-up

We simulate a generic distributed work system with 100 agents and discrete time steps, i e , this is a simulation without any particular application in mind In every time step, every agent decides whether to perform one unit of work or not Agents are divided into a fraction $1 - \beta$ of cooperative and a fraction β of free-riding agents Cooperative agents always perform one unit of work, while free-riders only perform work in every other round Furthermore, we also model strategic free-riding agents who seek to manipulate the accounting mechanism We let $\gamma \leq \beta$ denote the total fraction of all agents that are strategic free-riders

In each round that agent i performs work, it gets a random choice set of 5 agents With probability 0 1, i performs 1 unit of work for a random agent in the choice set and with probability 0 9 it uses the accounting mechanism and allocation rule to determine who receives work This aspect of the simulation is motivated by similar allocation rules used in P2P file sharing (e g , optimistic unchoking in BitTorrent) For the decentralized information exchange protocol, every agent contacts one other agent at random in each round, to exchange messages about direct experiences in the network Agents exchange reports about the last 5 agents they have interacted with and the 5 agents that have uploaded the most to them For the centralized version, we assume the existence of a center, collecting reports (which may still be untruthful) and making them immediately available For Basic and BarterCast-Basic, the strategic agents perform the optimal misreport manipulation, i e , always reporting they have consumed 0 units of work and contributed ∞ units of work Unless otherwise noted, we run each simulation for 100 time steps and record the work contributions and consumptions (averaged over 10 trials)

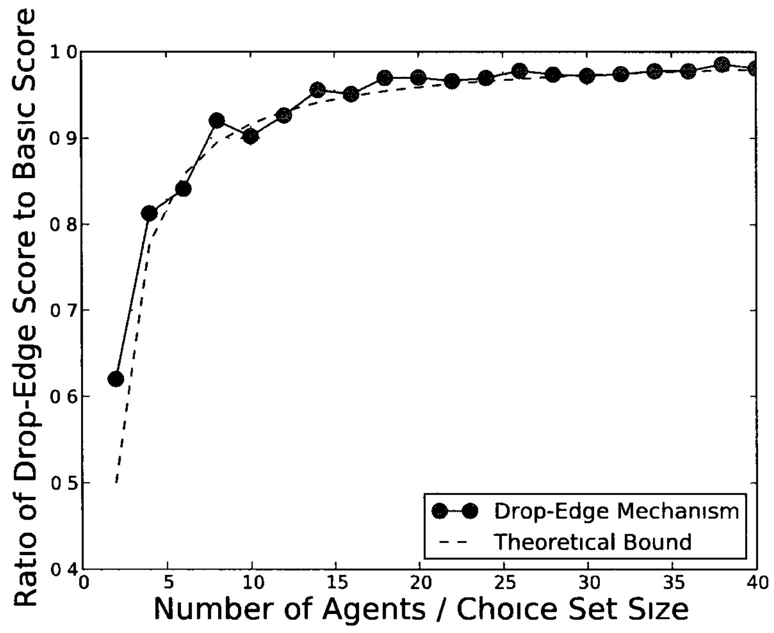


Figure 5 10 The ratio of the DROP-EDGE scores and the BASIC scores depending on network size As the number of agents in the network grows relative to the choice set size, the approximation ratio approaches 1 and the information loss of DROP-EDGE vanishes

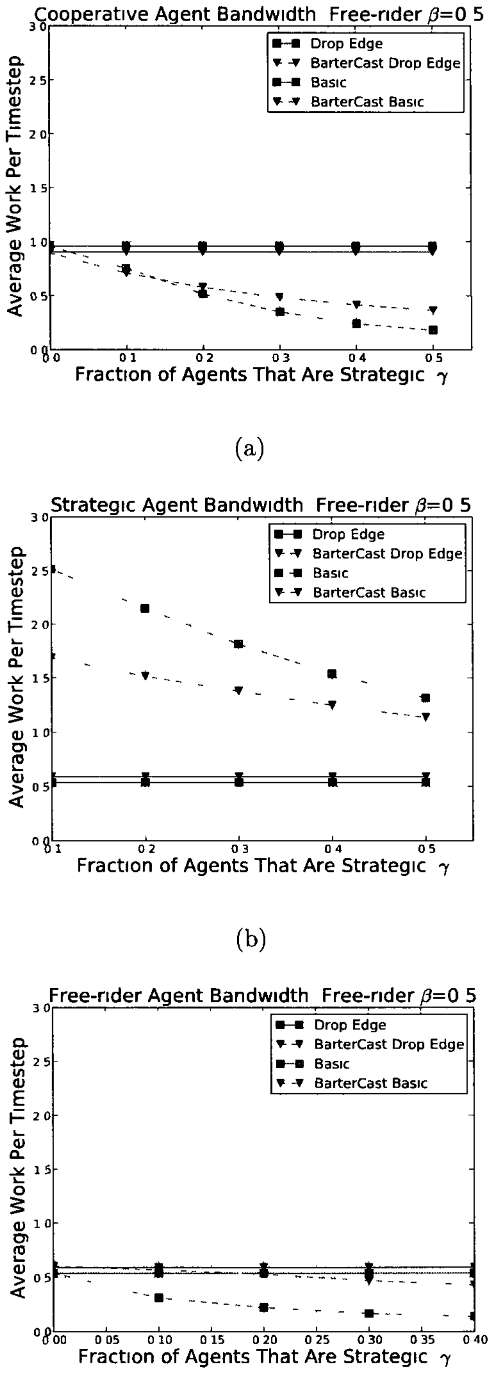
5.5.2 Information Loss of Drop-Edge

We first verify our theoretical results on information loss, in particular, that the information loss of DROP-EDGE vanishes as the number of agents in the network gets large relative to the size of the choice sets To isolate the effect of information loss due to dropped edges, we simulate a network without strategic agents, and compare the scores obtained by the centralized DROP-EDGE mechanism with those obtained by the centralized BASIC mechanism (that does not drop edges) Fixing a choice set size of $m = 5$ and free-rider agent fraction $\beta = 0\ 5$, we simulate networks of size $n = 10, 20, \quad 200$ After 100 time steps, for every agent we randomly choose a choice set and measure for every agent in the choice set the ratio of the Drop-Edge score and the score under the Basic mechanism Averaging over all agents and choice sets, we find that our empirical results closely match the theoretical results (Corollary 4) In Figure 5 10 we plot the ratios of the Drop-Edge scores and the Basic scores (or the "true" scores) for different network sizes We see that the approximation ratio approaches 1 as the number of agents in the network grows relative to the choice set size

5.5.3 Efficiency Results

We now consider direct measures of performance First, we measure the mechanisms' performance without strategic agents, to isolate their effectiveness as algorithms in aggregating information and promoting good decisions Consider the graphs in Figure 5 11 (a) with zero strategic agents, i e where $\gamma = 0$ We expect Drop-Edge to be slightly less efficient because we are dropping information that Basic is using, and no strategic agents are present that could harm the Basic mechanism We see that the efficiency is indeed higher under both versions of the Basic mechanism, but only minimally so (less than 5% difference)

The more interesting analysis concerns the overall efficiency with strategic agents present The *efficiency* of a particular agent type is defined to be the average amount of work received by that type of agent per time step *It is our goal to maximize the efficiency of the cooperative agents and to minimize the efficiency for free-riding*



(a)

(b)

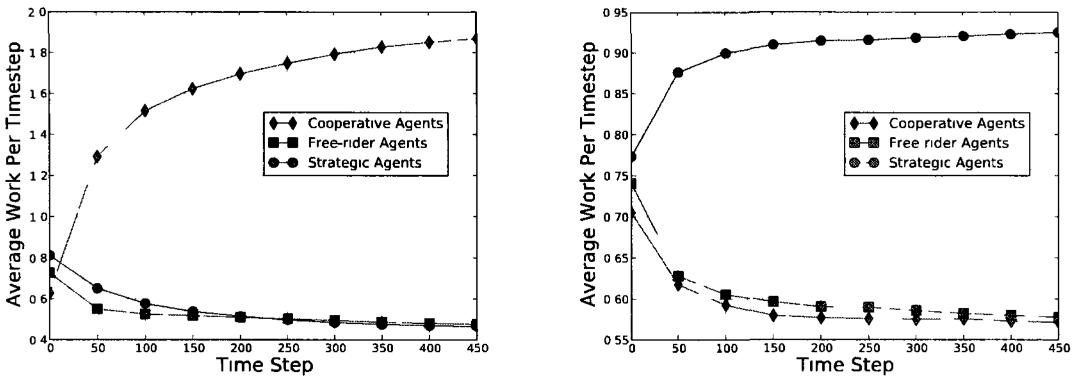
(c)

Fıgure 5 11 Average work consumptıon per tıme step, usıng the Basic and the Drop-Edge mechanısms The fractıon of free-rıdıng agents ıs $\beta = 0\ 5$

agents, and for strategıc free-rıders ın partıcular Ultımately, the goal ıs to cause agents to change from free-rıdıng to cooperatıng

We compare Fıgures 5 11 (a),(b) and (c), to analyze the relatıve efficıency of all agent types under the two mechanısms Note that the total efficıency ıs the same for both mechanısms because the amount of work performed by ındıvıdual agent types ıs fixed In Fıgure 5 11(b), we clearly see that strategıc agents are able to sharply ıncrease theır performance compared to the other agents (see Fıgures 5 11(a) and (c)) by mısreportıng under the BASIC mechanısm Thıs effect ıs partıcularly hıgh when only a few strategıc agents are ın the system Wıth 10% strategıc agents, the performance of a strategıc agent ıs 3 tımes as hıgh as that of the other agents under the decentralızed BARTERCAST-BASIC mechanısm, and more than 5 tımes as hıgh under the centralızed BASIC mechanısm Wıth the BASIC mechanısm, agents have a very large ıncentıve to act strategıcally The DROP-EDGE mechanısm ın contrast leads to the same constant efficıency for each ındıvıdual agent type (because there ıs no ıncentıve to manıpulate), and ın partıcular the efficıency of cooperatıve agents ıs almost twıce as hıgh as that of free-rıdıng agents

In practıce, strategıc mısreports may also occur under DROP-EDGE even though such behavıor ıs not useful for an agent We have tested DROP-EDGE ın settıngs wıth strategıc agents (not plotted) and although the efficıency of the cooperatıve agents decreases slıghtly as the proportıon of strategıc agents ıncreases, DROP-EDGE contınues to clearly outperform the BASIC mechanısm We also ran a longer experıment wıth $\beta = 0\ 5, \gamma = 0\ 2$ for 500 tıme steps, measurıng how efficıency changes over tıme In Fıgure 5 12 (a), we see that the benefit that strategıc agents gaın from mısreport-



(a) The Basic Mechanism (b) The Drop-Edge Mechanism

Figure 5 12 Evolution of average work consumption over time

ing in the Basic mechanism gets even larger over time Compare this against Figure 5 12 (b), which presents results for Drop-Edge Strategic agents cannot manipulate their scores, and receive decreasing amounts of work as the simulation proceeds At the end of the run, cooperative agents indeed receive twice as much work per round as the other agents, which is the ultimate goal, because they also perform exactly twice as much work

To summarize, in this section we have shown that the good approximation of the scores in Drop-Edge also translates into good system efficiency When strategic agents are present, the *Drop-Edge* mechanism clearly outperforms the Basic mechanism cooperative agents have higher efficiency, while free-riding agents have lower efficiency We have shown that the magnitude of this effect even grows over time Thus, we believe that using Drop-Edge over Basic in a real system has significant advantages for system efficiency

5.6 Experimental Analysis: A BitTorrent Overlay Protocol

5.6.1 The BitTorrent Protocol

In this section, we discuss an application of our mechanisms to BitTorrent In BitTorrent, the distributed work system is compromised of a collection of peers called a *swarm* A swarm begins when a *seeder*, an altruistic peer that has a complete file, sets up a server and allows other peers to download the file The file is partitioned into distinct pieces, and a unit of work in this system consists of the transmission of one piece from one peer to another The peers in a swarm download pieces from the original seeder and share pieces with each other A BitTorrent client maintains a limited number of simultaneous upload slots (usually 4-7 depending on the implementation) Peers that do not yet have the complete file (*leechers*), assign their slots to those peers that provide the highest upload rate in return, determined periodically, and the seeders assign their upload slots to those peers that have the highest download rate Peers that get a slot are called *unchoked*, while the other peers are *choked* Furthermore, there is one extra slot for *optimistic unchoking* which is assigned via a 30 seconds round-robin shift over all the interested peers regardless of their upload rate Due to optimistic unchoking, new peers have a chance to obtain their first pieces of data and bootstrap the process

Up to a certain limit, the more bandwidth a peer gives, the more it gets in return, which provides downloaders in a single swarm with a strong incentive to upload to others This policy is also often called "tit-for-tat" (even though Levin et al [60] show that they are not formally equivalent) This mechanism was one of the crucial design choices of the BitTorrent protocol [18], providing much better incentives to the peers in the system than previous protocols like Gnutella [2]

However, these incentives are only temporary and local the incentives only work in a bilateral way, and there is no incentive to continue sharing the file after the download has finished Ironically, it is even disadvantageous to share upon completing a file, since the consumed upload bandwidth cannot be used to do tit-for-tat in other downloads, which makes these downloads slower Using accounting mechanisms on top of the existing BitTorrent protocol, we want to remove this incentive problem to increase the overall efficiency of the system

5.6.2 Accounting Mechanisms for BitTorrent

When an accounting mechanism is available in a P2P file sharing system such as BitTorrent, this raises the question as to which allocation policy to use A natural candidate would be the ranking policy, which always gives preference to agents with a higher score and successfully separates sharers from free-riders in the round-based simulations However, the situation is more complicated when we are targeting a system like BitTorrent that is already deployed in practice First of all, a new BitTorrent client should be backwards-compatible with old clients that are not using the accounting mechanism Secondly, a user who installs a new client should have performance at least as good as with the old BitTorrent client

This puts some restrictions on what kind of allocation policies we can usefully employ Imagine agent i using an accounting-based client in a network with primarily "standard BitTorrent clients", i e, those that do not use the accounting mechanism A standard BitTorrent client allocates the optimistic unchoking slot to a random agent Thus, if agent i uses the ranking policy to decide which agent to unchoke optimistically, this does not affect its performance However, the remaining upload slots are normally allocated to those agents providing the highest download speed in return Myopically, this optimizes the download performance for the uploading agent Now, if agent i would allocate *all* upload slots based on the accounting mechanism, agent i's performance could degrade, because possibly the agents with the highest scores have uploaded a lot in the past, but they do not have any pieces to reciprocate in the current swarm Thus, agent i could be significantly worse off using an accounting-based client Consequentially, the ranking policy that we employ in our experiments with BitTorrent only uses the accounting mechanism to decide to whom to allocate the optimistic unchoking slot

The second allocation policy we employ is the banning policy, which we update there to the BitTorrent domain in the following way First, an agent never uploads to another agent with a score below a certain threshold But aside from this banning operation, the policy uses the standard BitTorrent policy the optimistic unchoking slot is assigned randomly to one of the peers that are not banned and the remaining slots are allocated to non-banned peers who provide the highest upload rate in return The idea here is that the threshold is set in such a way that free-riders are banned, but cooperative agents will never reach a score lower than this threshold Note that, similar to using the ranking policy for the allocation of all upload slots, it may also be sub-optimal for an agent to follow the banning policy This is because an agent with a score below the threshold may reciprocate with the highest upload rate in the *current* swarm However, there is an important difference between the ranking policy and the banning policy As the size of a swarm grows, and thus the number of cooperative peers with high upload rates grows as well, the disadvantage to a client using the banning policy diminishes Imagine there are 1,000 peers in the swarm, and the agent currently bans 10% of those, i e , 100 This still leaves 900 peers from whom the agent will now select those who reciprocate with the most bandwidth This is in contrast to employing the ranking policy for all slots, where swarm size does not matter The ranking policy will always select the agents with the highest scores, even if they currently reciprocate with the lowest speed in the whole swarm

Note that if a significant number of agents in the swarm use the accounting mechanism, then there is also an upside for an agent to upload to an agent who also uses an accounting mechanism-based client, because the agent can expect to be rewarded (directly or indirectly) for this cooperative behavior in the future Thus, whether and how much an agent would be worse off by employing the banning policy, would depend on the swarm size, the distribution of agent types in the swarm, and the particular banning threshold In practice, a small disadvantage may be tolerable (most BitTorrent users do not use BitThief or BitTyrant even though they can be faster than standard BitTorrent clients), but a large disadvantage must be avoided To guarantee that the disadvantage remains minimal, the particular banning threshold could be set dynamically by the client software, dependent on relative upload rates obtained from the different agents in the swarm However, we do not address this particular aspect further in this work Going forward, we focus on comparing the ranking and the banning policy Note that both policies are easy to implement on top of any P2P file sharing protocol and backwards-compatible with existing mechanisms

- **Ranking policy** Peers assign the optimistic unchoking slot to the interested peers in order of their scores, the remaining slots are allocated to those peers who provide the highest upload rate in return

- **Banning policy** Peers do not assign *any* upload slots to peers that have a score which is below a certain threshold δ The optimistic unchoking slot is assigned randomly to one of the peers that are not banned, the remaining slots are allocated to non-banned peers who provide the highest upload rate in return

For some of our experiments, we consider an *Omniscient (Centralized Max-flow)* mechanism To use this mechanism as a baseline, and to remove any misreport considerations, this mechanism stores the *true* up- and download statistics of peers in a central database, instead of in local databases The scores are computed using the Basic mechanism, based on the information stored in the central database

In the following experiments, we want to identify which effects are due to centralized or decentralized information exchange, and which are due to misreport-proofness To this end, we study the ranking and banning policies in combination with BarterCast-Basic, BarterCast-Drop-Edge, and the *Omniscient* mechanism

5.6.3 Simulation Set-up

We have built a simulator which incorporates all relevant aspects of BarterCast and BitTorrent We simulate an epidemic Peer Sampling Service using the decentralized BuddyCast protocol that is already implemented and released as part of the

Tribler file sharing client [77] Our simulator follows the BitTorrent protocol at the piece-level, including unchoking, optimistic unchoking, and rarest-first piece picking We have combined all processes in a simulation environment that can generate workloads based on either probabilistic request arrivals or traces of data

In our experiments we simulate 100 peers active in 10 swarms (i e , corresponding to 10 different files) during a simulated time of one week Note that a peer can be active in multiple swarms simultaneously A peer can be either a *cooperative*, a *free-riding*, or *strategic* Free-riders immediately leave the swarm after finishing a download, while cooperative peers share every completed file for 10 hours Strategic peers behave as free-riders, and in addition spread forged BarterCast messages in which they report a maximum upload to others and zero download The strategic peers also make misreports when DROP-EDGE is being used, which doesn't benefit them, but which introduces additional noise into the system All peers in our simulations have a 3 MBps downlink and a 512 KBps uplink, corresponding to common ADSL users As these users have very limited uploading capacity, they are likely to economize on sharing, and are therefore the most important target of sharing-ratio enforcement in current file sharing systems Finally, in the BarterCast messages, the agents report their information about the 10 agents they have seen most recently, and the 10 agents with the highest upload to them

5.6.4 Poisson-based Simulations

We generate a Poisson arrival process for file requests for each peer with an average of one request per day per peer Requests are homogeneously distributed over the

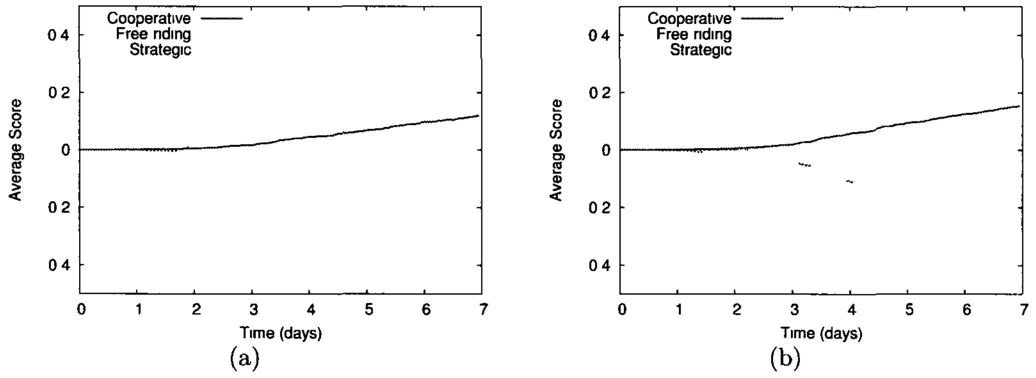


Fıgure 5 13 Evolutıon of the average scores over tıme for the three dıfferent agent types, usıng (a) BARTERCAST-BASIC and (b) BARTERCAST-DROP-EDGE

10 files Unless otherwıse noted, all of the followıng results are based on sımulatıons wıth 50% cooperatıve agents, 40% free-rıders, and 10% strategıc agents (dıfferent dıstrıbutıons of agents lead to qualıtatıvely sımılar results) In our experıments, we assess the evolutıon of the average scores of each group of agents, and compare the download performance for dıfferent polıcıes Furthermore, we evaluate the ınfluence of the bannıng threshold, and evaluate a sımple model for behavıoral change

Evolutıon of Agents' Scores

As a first step, we study how the scores of the dıfferent agents evolve over tıme, ı e , whether the accountıng mechanısm can successfully track agents' net contrıbutıon to the system Because each agent computes a dıfferent score for each other agent ın the network, we compute the *average score* $\mathbf{S}_i^M$ of agent i usıng mechanısm M as the average of the scores that each of the other $N-1$ agents assıgn to i

$$\mathbf{S}_i^M = \frac{1}{N-1}\sum_{j \neq i} S_{ji}^M(G_j, C_j) \qquad (5\ 21)$$

In Fıgure 5 13 the evolutıon of the average scores over tıme are plotted for the

the three dıfferent agent types for the two decentralızed versıons of the mechanısms Fıgure 5 13 (a) shows the results for BARTERCAST-BASIC The scores of the free-rıders clearly decrease over tıme, whıle the scores of the cooperatıve peers ıncrease slıghtly However, the strategıc peers benefit a lot from manıpulatıng the accountıng mechanısm, theır scores are sıgnıficantly hıgher than the scores of the other peers Thıs effect vanıshes completely when BARTERCAST-DROP-EDGE ıs used, whıch ıs dısplayed ın Fıgure 5 13 (b) Here, the strategıc agents have the same average scores as the free-rıders Obvıously, BARTERCAST-DROP-EDGE ıs very successful ın computıng scores that separate cooperatıve agents from free-rıders and strategıc agents

The Rankıng Polıcy

In thıs sectıon, we study study the effect of the rankıng polıcy on the average download performance of the agents All download speed results are normalızed wıth respect to the results of sımulatıons usıng the standard BıtTorrent protocol wıthout an accountıng mechanısm We consıder the centralızed versıon of the BASIC mechanısm, as well as the two decentralızed mechanısms BARTERCAST-BASIC and BARTERCAST-DROP-EDGE In Fıgure 5 14, the normalızed download speeds are plotted for cooperatıve, free-rıdıng, and strategıc agents, for the three dıfferent mechanısms under consıderatıon We see ımmedıately that usıng any of the mechanısms, the rankıng polıcy has no sıgnıficant effect, ı e , the performance ıs vırtually the same for all agent types

An closer ınvestıgatıon of thıs effect shows that thıs ıs caused by the sıze of the swarms As we sımulate relatıvely small swarms, peers do not always have enough

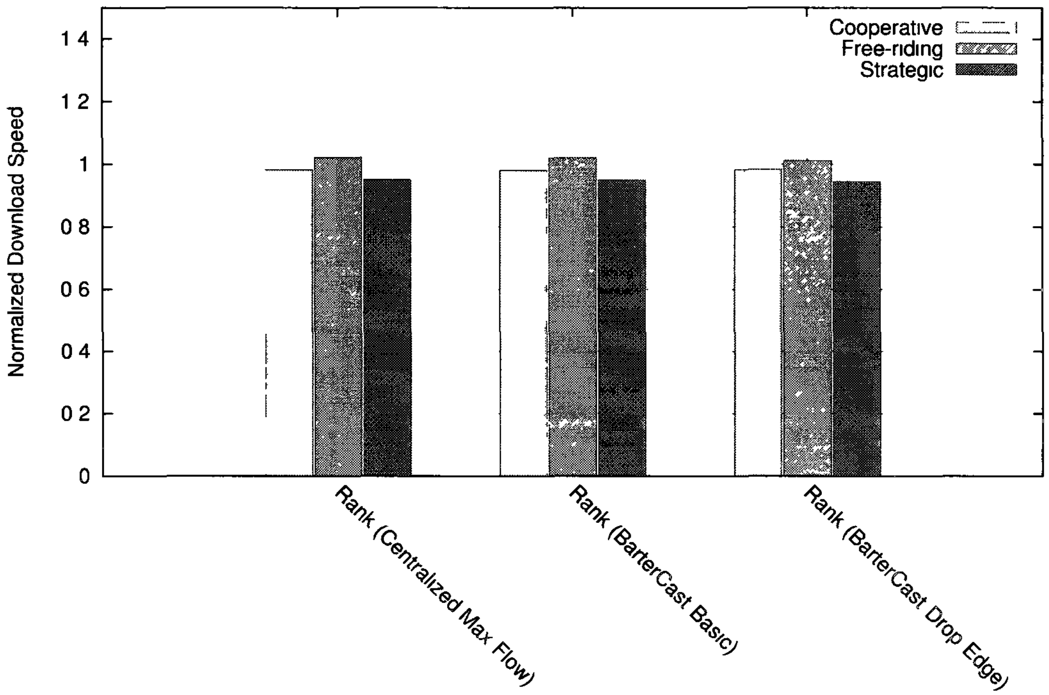


Figure 5 14 Comparison of the average download performance of cooperative agents, free-riders, and strategic peers for the ranking policy with different accounting mechanisms The download speeds are normalized relative to the original BitTorrent mechanism (without accounting)

requests from other peers to fill all of their upload slots Hence, free-riders and strategic agents can often find enough free slots to still have a normal performance This suggests that the ranking policy can only be effective if swarms are relatively large *and* peers know a significant fraction of the other peers in the network We argue that for the policy to be effective, it is necessary that a peer gets strictly more requests than he has upload slots, and that he has sufficient information about those peers to differentiate between cooperative agents and free-riders

To verify this hypothesis, we ran additional experiments where we varied the swarm sızes between 20 and 180 and where we varıed the accuracy of the ınformatıon avaılable to the agents In Fıgure 5 15, we dısplay the results from that experıment On the x-axıs we vary the *accountıng accuracy*, where an accuracy of 0 4 means that each agent has accurate scores for 40% of the others agents ın the network, and no ınformatıon for the rest Fırst, ıt ıs easy to see that the performance plateaus at an accuracy of 20% Thıs ıs a posıtıve result, because ıt ımplıes that even ın a large system where agents wıll always only have a partıal vıew of the network, the rankıng polıcy works well as long as each agent has some small amount of ınformatıon Next, we consıder the effect of varyıng the swarm sızes Wıth a very small swarm sıze of only 20 agents, the rankıng polıcy ıs not effectıve ın separatıng cooperatıve agents from free-rıders However, as we ıncrease the swarm sıze from 20 to 60 agents, thıs changes, as now the sharers have an average performance of 500Kbps and the free-rıders have an average performance of 400Kbps Thus, thıs verıfies our prevıous hypothesıs that the rankıng polıcy ıs only effectıve for swarms of some mınımal sıze

We also found that ıncreasıng the swarm sıze further from 60 to 180 caused no addıtıonal effect on the performance dıfference between cooperatıve agents and free-rıders It turns out that thıs effect occurs because when usıng the rankıng polıcy, we only use the accountıng mechanısm to allocated the optımıstıc unchokıng slot Each agent has 5 upload slots, out of whıch 4 are allocated based on best response rates Thıs explaıns why the performance of the cooperatıve agents compared to the free-rıders ıs exactly 5 to 4 (ı e , 500Kbps to 400Kbps), the same ratıo as the upload slots that are allocated to them on average Thıs reveals an ınherent lımıtatıon of usıng the rankıng polıcy ın BıtTorrent Because we want to maıntaın backwards compatıbılıty,

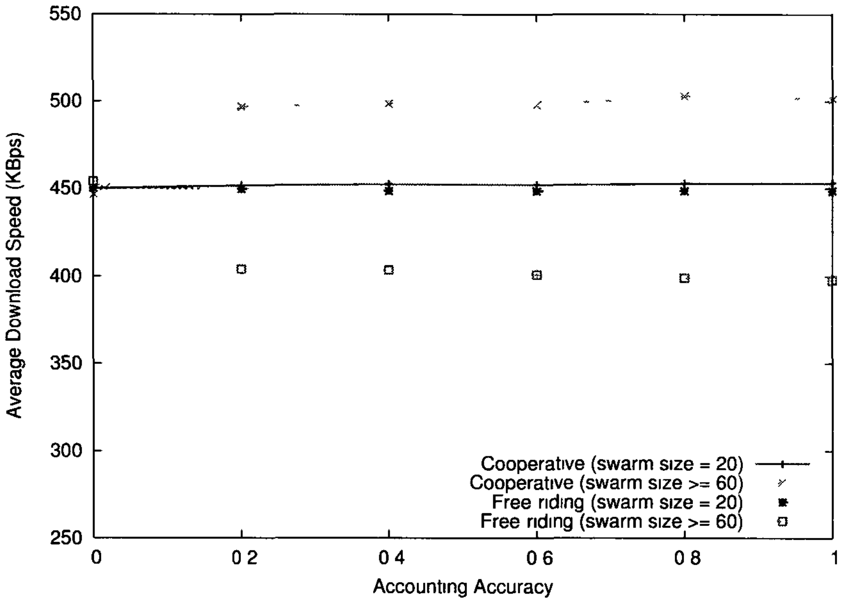


Figure 5 15 Performance of the Ranking Policy with different swarm sizes

and make sure that users of our client are at least as well of as users of the standard BitTorrent clients, the maximal effect that employing the ranking policy can have is limited we can at most reduce the performance of the free-riders by 20% compared to the cooperative agents This may not be enough to incentivize free-riders to change their behavior and become cooperative Thus, for the remainder of this section, we focus on the banning policy

The Banning Policy

As in the previous section, all the performance results presented in this section are normalized with respect to the results of simulations that do not implement any policy at all To study the banning policy in more detail, in particular to find good banning thresholds, we use the following monotonic function to transform the original

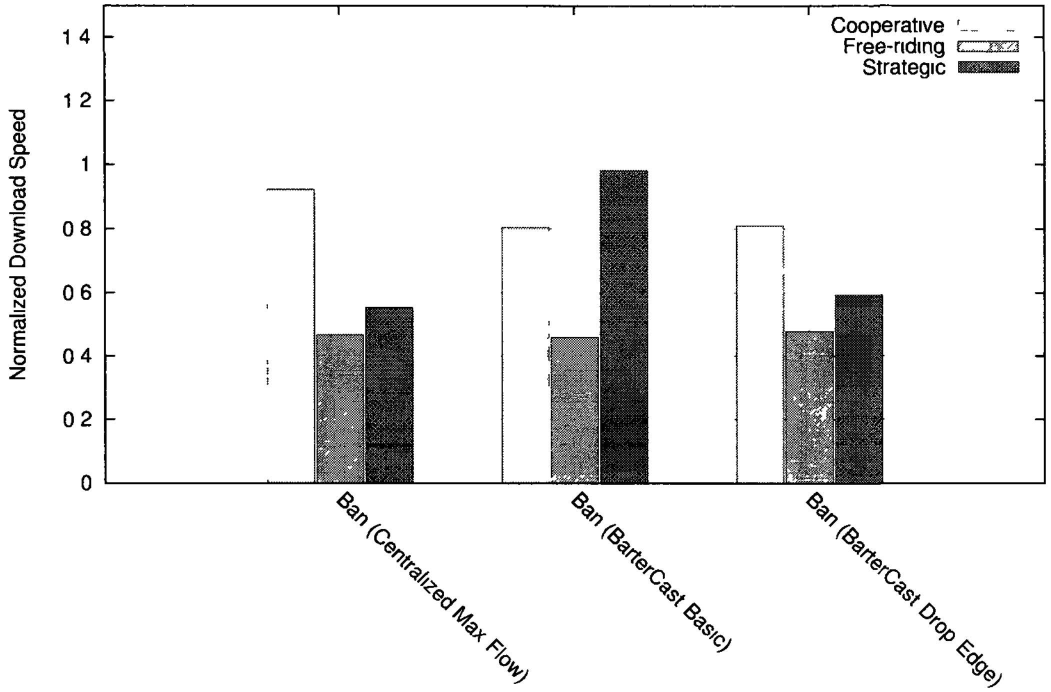


Figure 5.16 Comparison of the average download performance of cooperative agents, free-riders, and strategic peers for the banning policy with different accounting mechanisms

scores (which can be between $-\infty$ and $+\infty$) such that they are all between -1 and $+1$

Definition 31 ***(Normalized Scores)*** *Given subjective work graph G_i and choice set C_i, let G_i^B and G_i^D denote i's subjective work graphs after applying the* Basic *and the* Drop-Edge *mechanisms respectively The normalized scores for an agent j are*

$$S^X_{ij}(G_i, C_i) = \frac{\arctan(MF_{G_i^X}(j,i) - MF_{G_i^X}(i,j))}{\pi/2} \qquad (5.22)$$

where $X \in \{B, D\}$

For the remainder of this section, we will always use the normalized scores in our experıment Note that an agent about whom no ınformatıon ıs avaılable wıll have a score of 0, and an agent who contrıbutes as much work as he consumes wıll also have a score of 0, at least on average Thus, for the bannıng polıcy to have an effect, the bannıng threshold must be set to a value between -1 and 0 For the first analysıs we fix the bannıng threshold at $\delta = -0\ 5$, later we study the effect of varyıng the bannıng threshold between -1 and 0

The results for usıng the bannıng polıcy ın combınatıon wıth the three dıfferent accountıng mechanısms are shown ın Fıgure 5 16 We see that the performance of both free-rıders and strategıc agents ıs almost the same ın the omnıscıent (centralızed max-flow) mechanısm The small dıfference ıs due to randomızatıon ın the sımulatıon But more ımportantly, we see that the free-rıders and strategıc agents achıeve roughly half the performance of the cooperatıve agents Thıs parallels the results from the dıscrete, round-based sımulatıons presented ın Sectıon 5 5 3, where we showed ın Fıgure 5 12 that after a sufficıent number of tıme steps, the efficıency of the cooperatıve agents was approachıng twıce the efficıency of the free-rıders and strategıc agents

Next, we consıder the two decentralızed mechanısms For BARTERCAST-BASIC, the performance loss of the free-rıders compared to the cooperatıve agents ıs agaın roughly 50% However, now the strategıc peers achıeve a performance about twıce as hıgh as before, even hıgher than the cooperatıve peers Of course, thıs ıs due to the mısreport vulnerabılıty of the BASIC mechanısm whıch the strategıc peers exploıt Note that the cooperatıve agents' performance ıs a lıttle bıt lower than before, whıch can be explaıned by the fact that more of the bandwıdth now goes to the strategıc peers

Finally, consider the BarterCast-Drop-Edge mechanism Here, we see that the performance of the free-riders and the strategic peers is roughly the same again (the differences can only be attributed to random effects in the simulation), due to Drop-Edge being misreport-proof Now the cooperative agents achieve the highest performance, about 30%-40% higher than the free-riders and strategic agents Note that the performance of the cooperative agents under Drop-Edge is somewhat lower than under Centralized Max-Flow This is due to the strategic agents, who spread false information even though they cannot benefit from it This increases the overall noise in the system which leads to somewhat lower effectiveness of the accounting mechanism Overall, we can conclude that using the banning policy with BarterCast-Drop-Edge is indeed effective in separating cooperative agents from free-riders and strategic peers It achieves a larger performance difference than with the ranking policy However, a performance difference of 30%-40% might still be too small to incentivize free-riders to become cooperative agents Therefore, we study the effect of varying the banning threshold

In Figure 5 17, the normalized download performance for cooperative, free-riding, and strategic agents is plotted for various thresholds using the banning policy with the BarterCast-Drop-Edge mechanism As before, strategic agents have no benefit when the Drop-Edge mechanism is used, leading to performance comparable to that of the free-riders The figure shows that the more strict the threshold (i e , closer to 0), the larger the relative penalty for the free-riders and strategic agents compared to the cooperative agents We see that we can easily achieve performance differences larger than 30%-40% At $\delta = -0\ 1$, the relative performance difference

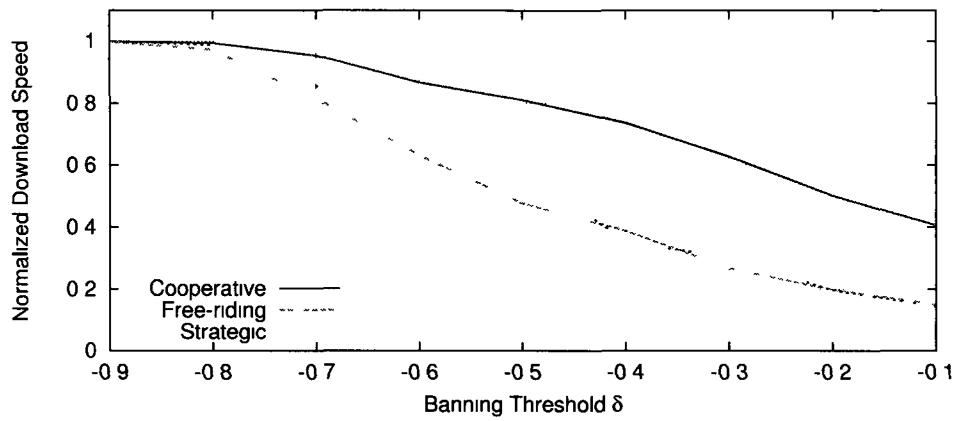


Figure 5 17 The normalized download performance of cooperative agents, free-riders, and strategic agents, using the banning policy for various thresholds δ under the BarterCast-Drop-Edge mechanism

is largest, with cooperative agents achieving roughly 2 5 times the performance of the free-riders However, the absolute performance of the cooperative agents is also smallest at $\delta = -0\ 1$, which is clearly detrimental to our design goal The lower performance of the cooperative agents is due to two effects First, at such a high banning threshold, agents will sometimes also ban cooperative agents because of the imperfect information due to the decentralized information exchange protocol Second, while free-riders and strategic agents try to exploit the system, they nevertheless do provide some bandwidth while they are still downloading files If a large majority of the free-riders and strategic peers is banned, then their bandwidth is also lost from the cooperative agents' perspective

This view, however, neglects the fact that in practice, we would expect a certain percentage of the free-riders and strategic agents to *change their behavior*, and become cooperative, if the banning policy with a high enough banning threshold were used Thus, to find the optimal banning threshold, we must consider a behavior model

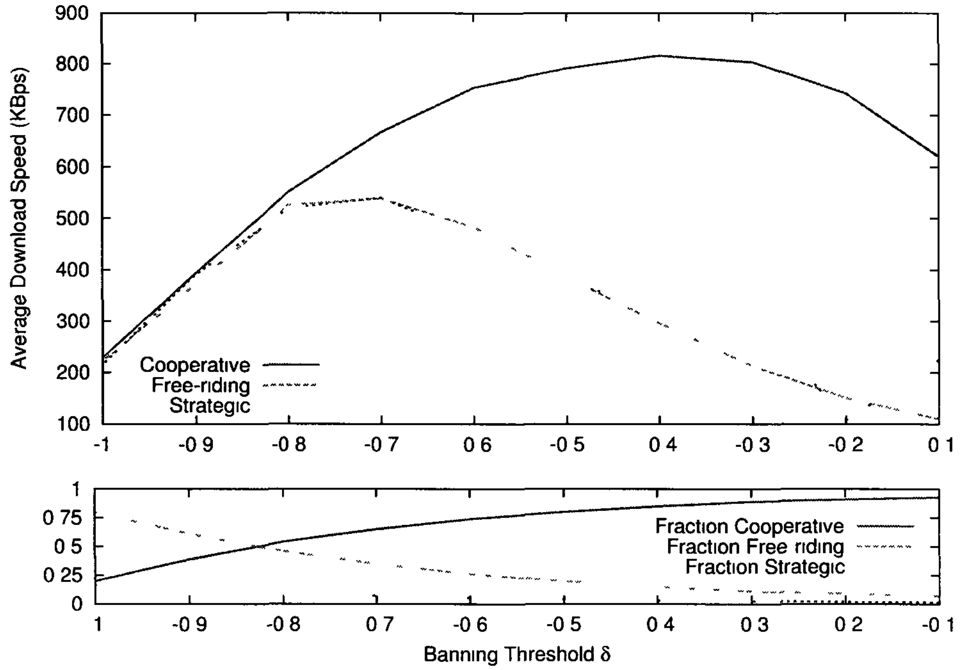


Figure 5 18 The performance of cooperative agents, free-riders, and strategic agents, assuming a behavioral model where more free-riders and strategic agents become cooperative as we increase the banning threshold for the BarterCast-Drop-Edge accounting mechanism The bottom graph displays the distribution of the agent types corresponding to the different banning thresholds

Banning Policy with a Behavioral Change Model

As a system designer, setting the optimal banning threshold requires some assumptions regarding how free-riders and strategic agents will change their behavior when facing clients with that employ a banning policy Assuming that many free-riders will become cooperative when they experience a severe penalty, a strict threshold is best, since in the end the overall system performance will improve for all peers because of the added resources of the former free-riders However, if free-rider conversion is slow, the prolonged loss of performance of the cooperative agents might be unacceptable, and a milder threshold should be considered To better understand this trade-off, we performed simulations assuming an illustrative relationship between the banning threshold δ and the percentage of free-riders and strategic agents in the system$f$

$$f(\delta) = 0\ 8\ \ (\frac{1}{16})^{\delta+1} \tag{5 23}$$

With the above relationship, a system with no banning (i e , $\delta = -1$) has 80% free-riders, while a system with very strict banning (i e , $\delta = 0$) has only 5% free-riders We assume that 25% of the free-riders are strategic In Figure 5 18, we display the download speed for cooperative, free-riding, and strategic agents in a system with the above relationship We observe that when the banning threshold is very low, all peers have a relatively low performance This is intuitive, because there is hardly any penalty for free-riders, and thus many peers will freer-ride, which leads to little supply of resources As we increase the banning threshold, the performance of all peers increases, as more and more of the free-riders and strategic peers become cooperative At some point ($\delta > -0\ 8$), there are enough cooperative agents in the system for the banning of free-riders to become effective Around $\delta = -0\ 4$, the download speed of the cooperative agents peaks, while the penalty for free-riding is very strong As we increase the banning threshold further towards 0, the disadvantages from banning more free-riders, and sometimes banning even cooperative agents due to incomplete information, starts having a negative effect on the performance of the cooperative agents Thus, the trade-off between sufficient banning of free-riders versus reducing unnecessary loss of performance for cooperative agents is clearly visible In practice, depending on the actual relationship f, it is up to community managers and system designers to devise policies that successfully balance this trade-off

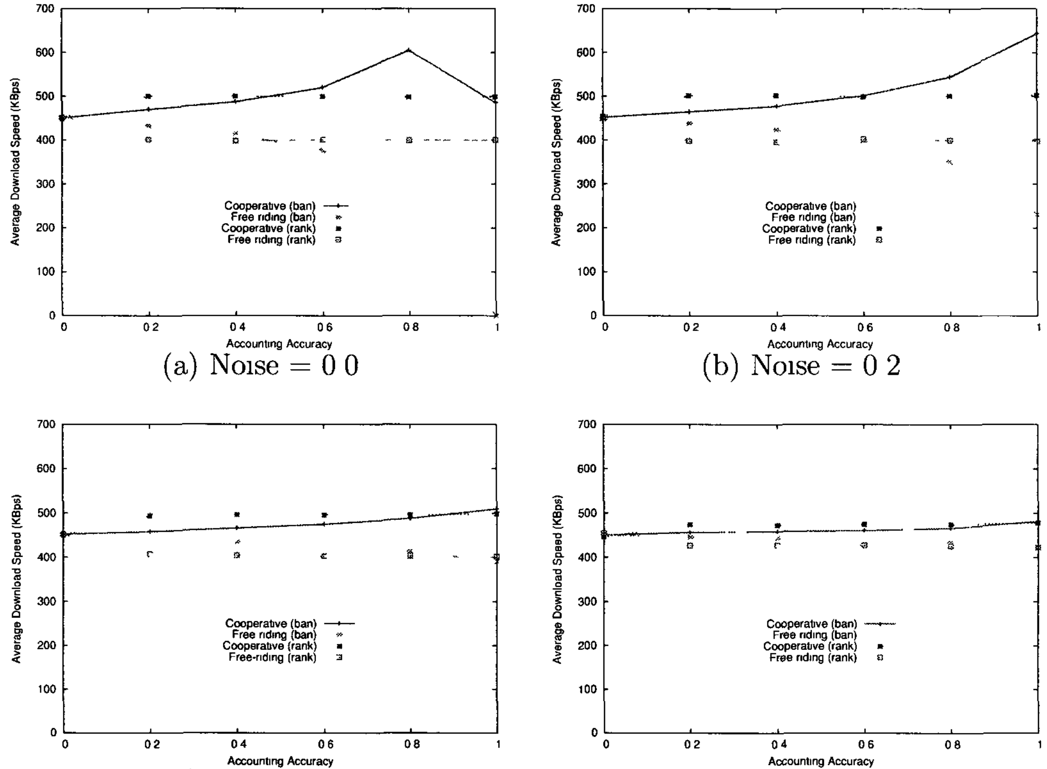
The Effect of Accounting Accuracy and Noise

While the network size in our simulation is relatively small (100 agents), real BitTorrent networks are much larger, on the order of millions of agents However, we cannot simulate significantly larger BitTorrent networks with such detail on the protocol level because we need to compute each agent's score from each other agent's perspective, and this simply takes too long once we go beyond a certain network size While the basic principle does not change when the network size increases from 100 to thousands or millions of agents, there are two aspects that do change First, in very large networks, it is more likely that two agents that meet have little or no information about each other (i e , are disconnected in the work graph) Second, the information that *is* available to the agents may be very noisy because the decentralized information exchange protocol needs a long time to spread information through a large network, and using the max-flow algorithm further distorts the scores We seek to better understand these two challenges

In contrast to all experiments we have presented so far, in the experiment we discuss here, the agents do not *compute* the scores of the other agents themselves Instead, we inject the scores, giving us the ability to control precisely how much and which information each agent has available when making a decision For the experiments, we let the *accounting accuracy* denote the average percentage of agents that an agent has any information about For example, an accuracy of 0 8 implies that on average, an agent is connected to 80% of the agents via paths of length 3 or less Another effect of using max-flow is that the scores computed by max-flow are only approximations for the net work performed by an agent In our experiments, we model this *noise* as the variance of the distribution from which we draw an agent's view of another agent's score

In Figure 5 19 we show the results from these experiments On the x-axis of all graphs, we vary the accounting accuracy between 0 (no information about any agent) to 1 (perfect information) We draw the agents' scores from a gaussian distribution with mean equal to the true scores, and with a standard deviation equal to 0 0 (no noise, i e , perfect information), 0 2, 0 4 and 0 8, which corresponds to the four graphs (a)-(d) We also experimented with shifting the mean up or down (i e , introducing systematic biases), but this did not lead to qualitatively different results

The results for the *ranking policy* are very straightforward Once the accuracy reaches the level of 0 2, the mechanism successfully separates cooperative agents from free-riders, giving them a performance ratio of 5 to 4, and this stays the same even as we increase the accuracy to 1 0 We have already explained in Section 5 6 4 what the origin of this effect is By comparing graphs (a) through (d) we also see that introducing noise into the system has no effect on the performance of the ranking policy, even up to a noise level of 0 8 (Figure 5 19 (c)) The explanation is simple the ranking policy picks the agent with highest scores, and of course the average scores of the cooperative agents are much higher than the average scores of the free-riders Thus, with some noise in the system, some highest-ranked cooperative agents might change their relative rank, but it is still very likely that the highest ranked agent (out of all agents) in any given choice set will be a cooperative agent and not a free-rider Only when we increase the noise even further does the result change a little bit, i e , the performance for cooperative agents decreases slightly and the performance of the



(a) Noise = 0 0

(b) Noise = 0 2

(c) Noise = 0 8

(d) Noise = 2 0

Figure 5 19 Analyzing the effect of accounting accuracy and noise on the banning policy (with threshold $\delta = -0\ 5$) and the ranking policy Here, no accounting mechanism is used The agents make allocation decisions using scores that are drawn from a gaussian distribution, with mean equal to the true scores, and standard deviation equal to the noise value free-riders increases slightly However, such high values of noise are not realistic in practice

Now we turn our attention to the *banning policy*, where we used a banning threshold of $\delta = -0\ 5$ As we see in Figure 5 19, the effects of accounting accuracy and noise are much more pronounced In particular, the performance difference between cooperative agents and free-riders keeps increasing as we increase the accounting accuracy from 0 to 1 This is expected because every agent that another agent has no information about has a score of 0, and thus will not be banned Finally, we consider the effect of adding noise when using the banning policy Here we see the biggest effects even going from no noise to a noise level of 0 2, the performance difference between cooperative agents and free-riders decreases significantly For example, with accuracy 0 8 and noise level 0, cooperative agents achieve 600KBps and free-riders achieve 300KBps [10] For an accuracy of 0 8 and noise level of 0 2, the cooperative agents' performance drops to 550KBps, and the free-riders' performance increases to 350KBps For a noise level of 0 8, the performance difference achieved via the banning policy is smaller than with the ranking policy, except for very high accuracy values, where the two policies perform essentially equally well This decrease in the effectiveness of the accounting mechanism with banning is expected Remember that the banning threshold is set in such a way that free-riders are banned and cooperative agents are not, and fine-tuning the threshold will always involved a trade-off between banning too many cooperative agents and too few free-riders Now, by adding noise

[10] The performance drop of the cooperative agents for noise level 0 and accuracy 1 0 occurs because with perfect information, all free-riders are banned from the system, which are then also no longer available to do tit-for-tat with the cooperative agents This is the same trade-off we discussed in Section 5 6 4

to the system, two kinds of mistakes start to happen: not only are some of the free-riders not banned that should be banned, but also some of the cooperative agents are now banned that should not be banned. This is in contrast to the ranking policy, where adding noise at first only affects which of the cooperative agents gets the highest score, but it takes a lot more noise, until a free-rider makes it to the top. Furthermore, the frequency of mistakes is also higher for the banning policy than for the ranking policy, because the banning policy can potentially affect every agent that is considered for any upload slot, not just the agents being considered for the optimistic unchoking slot.

To conclude this analysis, we can make some assumptions regarding what accuracy and noise levels to expect in real BitTorrent systems. Piatek et al. [75] have shown empirically that 99% of BitTorrent peers are connected via paths of length 2. Thus, even using a max-flow algorithm restricted to 1 or 2 hops, we can expect an accounting accuracy of 0.99 in real BitTorrent networks. It is a little more difficult to estimate the noise level of the accounting scores. In our own experiments using the round-based simulations as well as the BitTorrent simulations, we found that even though a decentralized information exchange protocol is used and with max-flow further distorting the accounting scores, the BarterCast-Drop-Edge mechanism can differentiate between cooperative agents and free-riders, even after just a few time steps and when only a few BarterCast messages have been exchanged (compare Figure 5.12(b) and Figure 5.13(b)). This suggests that even in larger networks we can expect a reasonably low level of noise. Based on these assumptions, considering Figure 5.19(b), we would expect the banning policy to cause a significant performance difference between cooperative agents and free-rider, even in large networks However, evaluating our mechanisms and allocation policies on a larger scale, ideally with real users, remains a formidable research challenge

5.7 Summary

In this chapter, we have studied distributed work systems, where agents perform small units of work for each other, without the ability for a third party to monitor those bilateral interactions The overall goal is to incentivize agents to be cooperative, i e , to perform as much work as they consume, and to prevent free-riding We have shown that previous approaches to solve this problem via *trust* or *reputation* mechanisms are not suitable, and propose to treat the problem as an *accounting* task instead The Drop-Edge mechanism removes any incentive for the agents to misreport, by selectively dropping some of the information available to the agents when considering for whom to perform work In our theoretical analysis, we have proved that the information loss of Drop-Edge is small and vanishes in the limit as the number of agents in the network grows The second class of manipulations we have considered are sybil attacks We have shown that under reasonable assumptions, no accounting mechanism can be sybil-proof However, we have also shown that a weaker robustness property, K-sybil-proofness, can be achieved for a limited class of sybil attacks

In the second part of the chapter, we have coupled Drop-Edge with BarterCast, a decentralized information exchange protocol, to study how accounting mechanisms can be used to improve the efficiency in distributed work system First, we have performed discrete, round-based simulations. Whereas manipulations are very useful without DROP-EDGE, the DROP-EDGE mechanism removes this problem and provides cooperative agents with higher efficiency, while free-riding and strategic agents have lower efficiency. In a second set of experiments, we have tested the effectiveness of accounting mechanisms as an overlay protocol for BitTorrent. Using TRIBLER, a real P2P file sharing client that is already deployed and being used in practice, we were able to run simulations at the BitTorrent protocol level. We have analyzed two different allocation policies, to decide how to allocate work based on the scores computed by the accounting mechanism: the ranking policy and the banning policy. The effectiveness of the ranking policy is limited in BitTorrent because we can only use it to allocate the optimistic unchoking slot. However, using the banning policy with a finely-tuned banning threshold, we can separate cooperative agents from free-riders, such that the performance of the cooperative agents is more than twice as high as that of free-riders. Under such conditions, it is realistic to assume that a significant fraction of free-riders would change their behavior and become cooperative. Assuming such a behavioral change, we have demonstrated significantly improved system efficiency. We have also provided a detailed analysis of the effects of accounting accuracy and noise on the accounting mechanisms. It is necessary that the accounting accuracy is relatively high and noise levels are relatively low for the banning policy to separate the performance of cooperative agents and free-riders to a large enough degree. Based on previous results and our own experiments, we expect that in large, real-world P2P file sharing networks, accounting accuracy would be relatively high noise levels would be relatively low, such that accounting mechanisms would be expected to be successful, even in very large networks with millions of agents However, studying the effectiveness of accounting mechanisms in real-world systems, and in large networks, remains an exciting research challenge

Chapter 6

Conclusion

In this thesis, I have described electronic market designs for non-traditional domains, where the market participants may be non-experts, may have high cognitive costs, may have other-regarding preferences, or where typical market institutions are not available We have seen that these domains require a departure from the standard agent model based on perfect rationality and self-interest, to enable novel market designs most suitable for the domains at hand The four main contributions of this thesis are

1 **Hidden Markets** A new design paradigm, hiding or simplifying a market's complexities, via a combination of an economic market design and a matching user interface A detailed case study of a hidden P2P backup market

2 **Market User Interface Design** An experimental study of the effects of various market UI design levers on users' decision performance and the market's efficiency, highlighting the importance of behavioral factors in decision making

3 **Selfishness vs Altruism in P2P File Sharing Networks** A field experiment on the file sharing public goods game, identifying the most predictive factors for whether users behave selfishly or altruistically

4 **Work Accounting Mechanisms** A formal and experimental study of accounting mechanisms that rely on voluntary reports, enabling more efficient distributed work systems in domains without money, contracts, or monitoring

A detailed summary of each contribution was included at the end of each thesis chapter In the next section, I provide a brief review and then take a retrospective view on the most important market design learnings Three directions for future research are discussed in Section 6 2

6.1 Review & Retrospection

6.1.1 Design and Analysis of a Hidden P2P Backup Market

In Chapter 2, I introduced the "hidden market design" paradigm and presented a detailed case study on a hidden P2P backup market The main contributions include the design of the market underlying the system as well as its user interface, a detailed theoretical analysis, and a formative usability study Our design hides or simplifies the combinatorial aspects of the market, prices, account balances, and payments A notable result from the theoretical analysis was that the more freedom we give users in choosing their supply ratios, the less robust is the system against irrational user behavior From a market design point of view, this finding was particularly nice, because it enables the market operator/designer to fine-tune the market for a particular user population The results from the usability study were encouraging for the hidden markets paradigm in general, and the P2P backup system in particular We have shown that real users are able to successfully interact with the P2P backup market, without even knowing that they were interacting with a market in the first place

In retrospect, one of the most important learnings from this project is that designing the economics of a market and its user interface *in concert*, can lead to novel market designs that may be simpler to use but perform better than traditional designs It is noteworthy, however, that many iterations were necessary until the final design was found At first, it was difficult to distance ourselves from traditional market designs from similar domains (e g , consider markets where users must monitor their budgets, or where payments are explicit) It took some time until we had pinpointed the defining characteristics of this domain, namely that most users will be non-experts that do not expect to see a market or monetary transactions in this domain While the design of *hidden markets* is grounded in economic theory, mechanism design, and traditional market design, it is currently still more of an art than a science However, we are confident that over time, as we gather more experience designing hidden markets, generalizable design principles that translate to other market domains will emerge

6.1.2 Market User Interface Design

In Chapter 3, I introduced our research agenda on "market user interface design " The main contribution was an experimental study, determining the effects of different market UIs on users' decision making performance and the market's efficiency We have seen that efficiency increases significantly as we increase the number of choices from 3 to 4 to 5, but then plateaus, with no statistically significant difference between 5 and 6 choices Moreover, we have identified a series of behavioral factors relevant in users' decision making processes, including UI complexity (i e , number of choices), position effects (i e , the relative rank of a choice), and loss aversion The strong loss aversion effect raises concerns about users' ability to optimally allocate a fixed budget in other real-world domains as well

The most surprising finding from this study was that the UI optimization, assuming behavioral play, did not increase efficiency, but rather decreased average efficiency This suggests that the quantal-response model was not a sufficiently accurate model of user behavior in this domain A more detailed look revealed that the decrease in efficiency was primarily due to the "more rational" users who did better using the UI that was optimized for optimal play, while there was no statistically significant difference in efficiency for the "less rational users " Thus, for (automated) market UI design to become effective, we need more detailed models of user behavior, and we predict a growing collaboration between computer scientists and behavioral economists in the future

6.1.3 Selfishness vs. Altruism in P2P File Sharing Networks

In Chapter 4, I described a large-scale field experiment studying the behavior of P2P file sharing users regarding their propensity to make selfish or altruistic choices Based on aggregate user behavior, we concluded that about 20% of the users consider the trade-off between the personal and societal effects of their actions when making a decision More specifically, when the speed-up value shown to the user was 10%, the likelihood of a user choosing the selfish client was, on average, 15% points higher compared to the treatment with no speed-up Other factors that exhibited significant correlations with users' behavior include the users' operating system (Linux users were the most altruistic), the users' age (the younger the more selfish), and the users' country of origin (users from Sweden were the most altruistic)

One of the main contributions was the experiment design, in particular the careful elicitation of users' understanding of the P2P file sharing game, and this also led to the most interesting finding We found that only about one third of the users understood the nature of the public goods game, but that this understanding had a large effect on users' behavior The percentage of users who chose the altruistic client was 16% points higher for those users who understood the underlying public goods game

6.1.4 Work Accounting Mechanisms

In Chapter 5, I introduced the study of *work accounting mechanisms* for distributed work systems where all interactions are bilateral and monitoring is not possible, where no contracts cover the interactions, and where no real or virtual currency can be used The key contribution consists of the formal analysis of accounting mechanism design, complemented by extensive experimental simulations Our goal was to design a mechanism that disincentivizes free-riding and is robust against misreport manipulations We have shown that misreport-proofness is essential for accounting mechanisms, because misreport manipulations are simple to perform, and the negative effects on efficiency are large. Furthermore, we proved that the Drop-Edge mechanism removes any incentives to misreport, and achieves this with minimal information loss. Via simulations, we have shown that by using the Drop-Edge mechanism, agents can successfully differentiate between free-riding and cooperative agents, which ultimately increases efficiency.

However, it is noteworthy that it was not straightforward to use accounting mechanisms as an overlay protocol for BitTorrent, mainly because in that domain, backwards compatibility is very important. We found, somewhat surprisingly, that in the BitTorrent domain, not only the accounting mechanism but also the choice of the allocation policy plays a major role. Another unexpected result was that no useful and sybil-proof accounting mechanism exists, which illustrates a key difference between the design of reputation and accounting mechanisms.

6.2 Future Work

6.2.1 Hidden Markets for Smart Grids

In Chapter 2, we described an application of the hidden market design idea to the domain of P2P backup. We believe, however, that the general hidden market design paradigm has applicability beyond P2P backup systems, and one such example could be *smart grids*, i.e., the next generation of electricity networks. The main idea of smart grids is to expose the changing market price for electricity to the end users such that they can decide when to consume more or less electricity. Furthermore, a digital connection between the power stations and users' homes allows for the remote control of consumers' appliances, which could then be turned off during times of excess demand The introduction of this market, i e , allowing the end-users to react to electricity prices more directly, suggests that energy would be allocated more efficiently Those users with a low value for energy could turn off their appliances when prices are high to save money, and those users with a high value for energy could leave their appliances on Governments and industry labs are currently making large research and development investments for smart grids [102], but it seems that the user interface aspect of these systems is not getting enough attention In fact, to date it is still unclear how much this technology actually benefits the end-users [50] We argue that to effectively involve the end-consumers of electricity in these new energy markets, a hidden market UI will be necessary The market design for this domain seems particularly challenging For example, how often will the price change? How do end-users specify when their appliances can be turned on or off? How much do end-users get paid for storing energy (e g , in electric vehicles)? Again, the decisions regarding all of these questions will have large impacts on how consumers behave in this market, and thus may be crucial for its ultimate success

6.2.2 Personalized Market User Interfaces

Based on the data from our study of market user interfaces which we presented in Chapter 3, we found that the UI optimization using the quantal-response model was not successful However, more interestingly, we found a very large, statistically significant difference between the *less rational* and the *more rational* users For the more rational users, the UI re-optimization led to a significantly lower efficiency, while there was no statistically significant effect on efficiency for the less rational users This naturally suggests a new research direction on "personalized market user interfaces " In many domains, in particular in the smartphone domain, there is a lot of user-specific, behavioral and non-behavioral data available that carries a lot of information about the particular user If we can estimate a user's "degree of rationality" based on this data, we can provide each user with a market UI that is specifically optimized for that particular (kind of) user Taking this idea a step further, we can also estimate a user's value for time and take this into account in the UI personalization Thus, there are still many opportunities in this space, ranging from more complete behavioral models to algorithms for learning user preferences and automated UI optimization

6.2.3 Social Feedback for Market Participants

In Chapter 4, we presented our study of user behavior in P2P file sharing networks, and some of our findings suggest new directions for the design of peer production systems and markets that are situated in social communities For example, we have seen that different user groups have different priors regarding their likelihood of being altruistic or selfish Thus, it is conceivable that in some domains we can provide each individual user with specially-tailored incentives, maximizing the probability that the user will cooperate However, note that this requires a lot of knowledge about an individual user, and this approach only works in domains where it is possible that each user has a different interaction with the system/market

The most interesting direction for future research is based on our finding that users who understand the free-riding problem, i e , the nature of the public goods game, were significantly more likely to cooperate This can have interesting consequences for design as well For example, this result suggests that if we could *educate* the users of a system about the particular public goods game they are playing, then we might be able to increase their rate of cooperation How to achieve this in practice, however, is still an open question and one could imagine various ways to do so One way would be simply to explain to the user the overall game that's being played and the public goods dilemma that could arise If direct education is not suitable, then *social feedback* might be an indirect way to achieve the same effect, making users realize the societal consequences of their actions on others More specifically, we could provide feedback to the user about his marginal contribution to the well-being of the community If the user cares about his own benefit as well as the welfare of the community, such feedback could induce higher rates of cooperation In peer production environments that primarily rely on voluntary contributions by their member, the effects of such feedback could be very large Thus, studying this hypothesis in the lab and in the field is an important direction for future research

Bibliography

[1] Atıla Abdulkadıroglu and Tayfun Sonmez School Choıce A Mechanısm Desıgn Approach *Amerıcan Economıc Revıew*, 93(3) 729–747, 2003

[2] Eytan Adar and Bernardo A Huberman Free Rıdıng on Gnutella *Fırst Monday*, 5(10), 2000

[3] Jr Alexander S Kelso and Vıncent P Crawford Job Matchıng, Coalıtıon Formatıon, and Gross Substıtutes *Econometrıca*, 50(6) 1483–1504, 1982

[4] Noga Alon, Felıx Fıscher, Arıel D Procaccıa, and Moshe Tennenholtz Sum of Us Strategyproof Selectıon from the Selectors In *Proceedıngs of the 13th Conference on Theoretıcal Aspects of Ratıonalıty and Knowledge (TARK)*, Gronıngen, NL, July 2011

[5] Nazareno Andrade, Mıranda Mowbray, Alıandro Lıma, Gustavo Wagner, and Mateı Rıpeanu Influences on Cooperatıon ın BıtTorrent Communıtıes In *Proceedıngs of the 2005 ACM SIGCOMM Workshop on Economıcs of Peer-to-peer Systems (P2PECON)*, New York, NY, 2005

[6] Chrıstına Aperȷıs and Ramesh Joharı A Peer-to-Peer System as an Exchange Economy In *Proceedıngs of the Workshop on Game Theory for Communıcatıons and Networks (GameNets)*, Pısa, Italy, October 2006

[7] Tına Balke, Marına De Vos, Julıan Padget, and Frank Fıtzek Usıng A Normatıve Framework to Explore the Prototypıng of Wıreless Grıds In *Proceedıngs of the Workshop on Coordınatıon, Organızatıon, Instıtutıons and Norms ın Multı-Agent Systems at AAMAS*, Toronto, Canada, May 2010

[8] Yochaı Benkler Coase's Penguın, or, Lınux and The Nature of the Fırm *The Yale Law Journal*, 112(3) 369–446, 2002

[9] Wıllıam J Bolosky, John R Douceur, and Jon Howell The Farsıte Proȷect A Retrospectıve *SIGOPS Operatıng Systems Revıew*, 41(2) 17–26, 2007

[10] C G Broyden, J E Dennis Jr , and Jorge J Moré On the Local and Superlinear Convergence of Quasi-Newton Methods *IMA Journal of Applied Mathematics*, 12(3) 223–245, 1973

[11] Georg Buscher, Susan Dumais, and Edward Cutrell The Good, the Bad, and the Random An Eye-Tracking Study of Ad Quality in Web Search In *Proceedings of the 33rd Annual International ACM SIGIR Conference*, Geneva, Switzerland, July/August 2010

[12] Colin F Camerer and Ernst Fehr Measuring Social Norms and Preferences Using Experimental Games A Guide for Social Scientists In Joseph Henrich, Robert Boyd, Samuel Bowles, Colin Camerer, Ernst Fehr, and Herbert Gintis, editors, *Foundations of Human Sociality - Experimental and Ethnographic Evidence from 15 Smale-Scale Societies* Oxford University Press, 2004

[13] Christopher F Chabris, David I Laibson, Carrie L Morris, Jonathon P Schuldt, and Dmitry Taubinsky The Allocation of Time in Decision-Making *Journal of the European Economic Association*, 7 628–637, 2009

[14] Yan Chen, Teck-Hua Ho, and Yong-Mi Kim Knowledge Market Design A Field Experiment at Google Answers *Journal of Public Economic Theory*, 12(4) 641–664, 2010

[15] Alice Cheng and Eric Friedman Sybilproof Reputation Mechanisms In *Proceedings of the ACM SIGCOMM Workshop on Economics of Peer-to-Peer Systems (P2PECON)*, pages 128–132, Philadelphia, PA, August 2005

[16] Alice Cheng and Eric Friedman Manipulability of PageRank under Sybil Strategies In *Proceedings of the 1st Workshop of Networked Systems (NetEcon06)*, Ann Arbor, MI, June 2006

[17] James Choi, David Laibson, and Brigitte C Madrian Plan Design and 401(k) Savings Outcomes *National Tax Journal*, 57(2) 275–298, 2004

[18] Bram Cohen Incentives Build Robustness in BitTorrent In *Proceedings of the Workshop on Economics of Peer-to-Peer Systems (P2PEcon)*, Berkeley, CA, June 2003

[19] Bo Cowgill, Justin Wolfers, and Eric Zitzewitz Using Prediction Markets to Track Information Flows Evidence from Google Working Paper, January 2008

[20] Landon P Cox and Brian D Noble Samsara Honor Among Thieves in Peer-to-Peer Storage In *Proceedings of the 19th ACM Symposium on Operating Systems Principles (SOSP)*, pages 120–132, Bolton Landing, NY, 2003

[21] Mary Czerwinski, Desney S Tan, and George G Robertson Women Take a Wider View In *Proceedings of the Conference on Human Factors in Computing Systems (CHI)*, Minneapolis, MN, April 2002S

[22] Pranav Dandekar, Ashish Goel, Ramesh Govindan, and Ian Post Liquidity in Credit Networks A Little Trust Goes a Long Way In *Proceedings of the 12th ACM Conference on Electronic Commerce (EC)*, San Jose, CA, June 2011

[23] Stefano DellaVigna Psychology and Economics Evidence from the Field *Journal of Economic Literature*, 47 315–372, 2009

[24] Susan Dumais, Georg Buscher, and Edward Cutrell Individual Differences in Gaze Patterns for Web Search In *Proceedings of the Information Interaction in Conext Symposium (IIiX)*, New Brunswick, NJ, August 2010

[25] Benjamin Edelman, Michael Ostrovsky, and Michael Schwarz Internet Advertising and the Generalized Second Price Auction Selling Billions of Dollars Worth of Keywords *American Economic Review*, 97(1) 242–259, 2007

[26] Ernst Fehr and Klaus M Schmidt A Theory of Fairness, Competition and Cooperation *Quarterly Journal of Economics*, 114(3) 817–868, 1999

[27] Michal Feldman, Kevin Lai, Ion Stoica, and John Chuang Robust Incentive Techniques for Peer-to-Peer Networks In *Proceedings of the 5th ACM Conference on Electronic Commerce (EC)*, New York, NY, May 2004

[28] Michal Feldman, Christos Papadimitriou, John Chuang, and Ion Stoica Free-Riding and Whitewashing in Peer-to-Peer Systems *IEEE Journal on Selected Areas in Communications*, 24(5) 1010–1019, 2006

[29] Michael J Freedman, Christina Aperjis, and Ramesh Johari Prices are Right Managing Resources and Incentives in Peer-Assisted Content Distribution In *Proceedings of the 7th International Workshop on Peer-to-Peer Systems (IPTPS)*, Tampa Bay, FL, 2008

[30] Bruno S Frey Awards as Compensation *European Management Review*, 4 6–14, 2007

[31] Bruno S Frey and Reto Jegen Motivation Crowding Theory *Journal of Economic Survey*, 15(5) 589–611, 2001

[32] Eric Friedman, Paul Resnick, and Rahul Sami Manipulation-Resistant Reputation Systems In Noam Nisan, Tim Roughgarden, Eva Tardos, and Vijay V Vazirani, editors, *Algorithmic Game Theory*, pages 677–698 Cambridge University Press, New York, NY, 2007

[33] Eric J Friedman, Joseph Y Halpern, and Ian Kash Efficiency and Nash Equilibria in a Scrip System for P2P Networks In *Proceedings of the 7th ACM Conference on Electronic Commerce (EC)*, pages 140–149, Ann Arbor, Michigan, 2006

[34] Milton Friedman *Essays in Positive Economics* Chicago Press, Chicago, 1953

[35] Drew Fudenberg and Jean Tirole *Game Theory* MIT Press, Cambridge, MA, 1991

[36] Xavier Gabaix, David Laibson, Guillermo Moloche, and Stephen Weinberg Costly Information Acquisition Experimental Analysis of a BoundedlyRational Model *American Economic Review*, 96 (4) 1043–1068, 2006

[37] Krzysztof Z Gajos, Daniel S Weld, and Jacob O Wobbrockc Automatically Generating Personalized User Interfaces with Supple *Artificial Intelligence*, 174 910–950, 2010

[38] Sharad Goel, David Pennock, Daniel M Reeves, and Cong Yu Yoopick A Combinatorial Sports Prediction Market In *Proceedings of the 23rd Conference on Artificial Intelligence (AAAI)*, pages 1880–1881, Chicago, IL, July 2008

[39] Andrew V Goldberg and Robert E Tarjan A New Approach to the Maximum-Flow Problem *Journal of the ACM*, 35(4) 921–940, 1988

[40] Philippe Golle, Kevin Leyton-Brown, Ilya Mironov, and Mark Lillibridge Incentives for Sharing in Peer-to-Peer Networks *Electronic Commerce*, 2232 75–87, 2001

[41] Minaxi Gupta, Paul Judge, and Mostafa Ammar A Reputation System for Peer-to-Peer Networks In *Proceedings of the 13th ACM International Workshop on Network and Operating Systems Support for Digital Audio and Video*, Monterey, CA, June 2003

[42] Friedrich A Hayek The Uses of Knowledge in Society *American Economic Review*, 35(4) 519–530, 1945

[43] Eric Horvitz and Matthew Barry Display of Information for Time-Critical Decision Making In *Proceedings of the 11th Conference on Uncertainty in Artificial Intelligence (UAI)*, Montreal, Canada, August 1995

[44] Gary Hsieh and Scott Counts Mimir A Market-Based Real-Time Question and Answer Service In *Proceedings of the Conference on Human Factors in Computing Systems (CHI)*, Boston, MA, April 2009

[45] Gary Hsieh, Robert Kraut, Scott E. Hudson, and Roberto Weber. Can Markets Help? Applying Market Mechanisms to Improve Synchronous Communcation. In *Proceedings of the ACM Conference on Computer Supported Cooperative Work (CSCW)*, San Diego, California, November 2008.

[46] Sheena S. Iyengar and Wei Jiang. Choosing Not to Choose: The Effect of More Choices on Retirement Savings Decisions. Columbia University Working Paper, 2003.

[47] Sheena S. Iyengar and Mark R. Lepper. When Choice is Demotivating: Can One Desire Too Much of a Good Thing? *Journal of Personality and Social Psychology*, 79(6):995–1006, 2000.

[48] Matthew O. Jackson. Mechanism theory. In Ulrich Derigs, editor, *Optimization and Operations Research (The Encyclopedia of Life Support Systems)*. EOLSS Publishers, Oxford, UK, 2003.

[49] Eric J. Johnson, John W. Payne, and James R. Bettman. Information Displays and Preference Reversals. *Organizational Behavior and Human Decision Processes*, 42(1):1–21, 1988.

[50] Alexander Jung. Teure Ersparnis. `http://www.spiegel.de/spiegel/0,1518,711967,00.html`, October 2010.

[51] Sepandar Kamvar, Mario Schlosser, and Hector Garcia-Molina. The EigenTrust Algorithm for Reputation Management in P2P Networks. In *Proceedings of the 12th International World Wide Web Conference (WWW)*, Budapest, Hungary, May 2003.

[52] Leonid V. Kantorovich. Functional analysis and applied mathematics. *Uspekhi Matematicheskikh Nauk*, 3(6):89–185, 1948.

[53] Leonid V. Kantorovich and Gleb P. Akilov. *Functional Analysis in Normed Spaces*, volume 46 of *International Series of Pure and applied Mathematics*. The Macmillan Co., New York, 1964.

[54] Ian A. Kash, Eric J. Friedman, and Joseph Y. Halpern. Optimizing Scrip Systems: Efficiency, Crashes, Hoarders, and Altruists. In *Proceedings of the 8th ACM Conference on Electronic Commerce (EC)*, San Diego, California, June 2007.

[55] Anshul Kothari, David C. Parkes, and Subhash Suri. Approximately-Strategyproof and Tractable Multi-Unit Auctions. *Decision Support Systems*, 39:105–121, 2005.

[56] Vijay Krishna. *Auction Theory*. Academic Press, San Diego, 2002.

[57] Ramayya Krishnan, Michael D Smith, Zhulei Tang, and Rahul Telang The Virtual Commons Why Free-Riding can be Tolerated in Peer-to-Peer Networks In *Proceedings of the Workshop on Information Systems and Economics (WISE)*, Seattle, WA, December 2003

[58] John Kubiatowicz, David Bindel, Yan Chen, Steven Czerwinski, Patrick Eaton, Dennis Geels, Ramakrishna Gummadi, Sean Rhea, Hakim Weatherspoon, Westley Weimer, Chris Wells, and Ben Zhao OceanStore An Architecture for Global-Scale Persistent Storage In *Proceedings of the 9th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, Cambridge, MA, November 2000

[59] Kevin Lai, Lars Rasmusson, Eytan Adar, Li Zhang, and Bernardo A Huberman Tycoon An Implementation of a Distributed, Market-based Resource Allocation System *Multiagent Grid Systems*, 1(3) 169–182, 2005

[60] Dave Levin, Katrina LaCurts, Neil Spring, and Bobby Bhattacharjee BitTorrent is an Auction Analyzing and Improving BitTorrents Incentives In *Proceedings of the ACM SIGCOMM Conference*, Seattle, WA, August 2008

[61] Jin Li and Cha Zhang Distributed Hosting of Web Content with Erasure Coding and Unequal Weight Assignment In *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)*, pages 27–30, Taipei, Taiwan, June 2004

[62] Qiao Lian, Yu Peng, Mao Yang, Zheng Zhang, Yafei Dai, and Xiaoming Li Robust Incentives via Multi-level Tit-for-Tat In *Proceedings of the 5th International Workshop on Peer-to-Peer Systems (IPTPS)*, Santa Barbara, CA, February 2006

[63] JustAnswer LLC JustAnswer Website `http //www justanswer com`, May 2011

[64] Thomas Locher, Patrick Moor, Stefan Schmid, and Roger Wattenhofer Free Riding in BitTorrent is Cheap In *Proceedings of the 5th Workshop on Hot Topics in Networks (HotNets)*, Irvine, California, November 2006

[65] Andreu Mas-Colell, Michael D Whinston, and Jerry R Green *Microeconomic Theory* Oxford University Press, New York, 1995

[66] Richard McKelvey and Thomas Palfrey Quantal Response Equilibria for Normal Form Games *Games and Economic Behavior*, 10 6–38, 1995

[67] Michel Meulpolder, Lucia D'Acunto, Mihai Capotă, Maciej Wojciechowski, Johan A Pouwelse, Dick H J Epema, and Henk J Sips Public and Private

BitTorrent Communities A Measurement Study In *Proceedings of the 9th International Conference on Peer-to-peer Systems (IPTPS)*, San Jose, CA, April 2010

[68] Michel Meulpolder, Johan Pouwelse, Dick H Epema, and Henk J Sips BarterCast A Practical Approach to Prevent Lazy Freeriding in P2P Networks In *Proceedings of the 6th International Workshop on Hot Topics in Peer-to-Peer Systems (Hot-P2P)*, Rome, Italy, May 2009

[69] Paul Milgrom Criticial Issues in the Practice of Market Design *Economic Inquiry*, 49(2) 311–320, 2011

[70] John Nelder and Robert Wedderburn Generalized Linear Models *Journal of the Royal Statistical Society*, 135 (3) 370–384, 1972

[71] Axel Ockenfels and Alvin E Roth Late and Multiple Bidding in Second-Price Internet Auctions Theory and Evidence Concerning Different Rules for Ending an Auction *Games and Economic Behavior*, 55 297–320, 2006

[72] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd The PageRank Citation Ranking Bringing Order to the Web Technical report, Stanford Digital Library Technologies Project, 1998

[73] Satu S Parikh and Gerald L Lohse Electronic Futures Markets Versus Floor Trading Implications for Interface Design In *Proceedings of theConference on Human Factors in Computing Systems (CHI)*, Denver, CO, May 1995

[74] John W Payne and James R Bettman Preferential Choice and Adaptive Strategy Use In Gerg Gigerenzer and Reinhard Selten, editors, *Bounded Rationality The Adaptive Toolbox* MIT University Press, Cambridge, MA, 2001

[75] Michael Piatek, Tomas Isdal, Arvind Krishnamurthy, and Thomas Anderson One Hop Reputations for Peer to Peer File Sharing Workloads In *Proceedings of the 5th USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, pages 1–14, San Francisco, California, April 2008

[76] Johan A Pouwelse, Pawel Garbacki, Dick H J Epema, and Henk J Sips The BitTorrent P2P File-Sharing System Measurements and Analysis In *Proceedings of the 4th International Workshop on Peer-to-Peer Systems (IPTPS)*, Ithaca, NY, February 2005

[77] Johan A Pouwelse, Pawel Garbacki, Jun Wang, Arno Bakker, Jie Yang, Alexandru Iosup, Dick H J Epema, Marcel J T Reinders, Maarten R van Steen, and Henk J Sips Tribler A Social-based Peer-to-peer System *Concurrency and Computation Practice and Experience*, 20(2) 127–138, 2008

[78] Matthew Rabin. Incorporating Fairness Into Game Theory and Economics. *American Economic Review*, 83(5):1281–1302, 1993.

[79] Rysavy Research. Mobile Broadband Capacity Constraints and the Need for Optimization. Research report available at http://www.rysavy.com/papers.html, 2010.

[80] Paul Resnick and Rahul Sami. Sybilproof Transitive Trust Protocols. In *Proceedings of the 10th ACM Conference on Electronic Commerce (EC)*, Stanford, CA, July 2009.

[81] Alvin E. Roth. Comments on Tversky's 'Rational Theory and Constructive Choice'. In *Proceedings of the IEA Conference on The Rational Foundations of Economic Behavior*, Torino, Italy, October 1993.

[82] Alvin E. Roth. The Economist as Engineer: Game Theory, Experimentation, and Computation as Tools for Design Economics. *Econometrica*, 70(4):1341–1378, 2002.

[83] Alvin E. Roth. Repugnance as a Constraint on Markets. *Journal of Economic Perspectives*, 21(3):37–58, 2007.

[84] Alvin E. Roth. What Have we Learned From Market Design? *The Economic Journal*, 118:285–310, 2008.

[85] Alvin E. Roth and Axel Ockenfels. Last-Minute Bidding and the Rules for Ending Second-Price Auctions: Evidence from eBay and Amazon Auctions on the Internet. *American Economic Review*, 92(4):1093–1103, 2002.

[86] Alvin E. Roth and Elliott Peranson. The Redesign of the Matching Market for American Physicians: Some Engineering Aspects of Economic Design. *American Economic Review*, 89(4):748–780, 1999.

[87] Alvin E. Roth, Tayfun Sonmez, and M. Utku Unver. Kidney Exchange. *Quarterly Journal of Economics*, 119(2):457–488, 2004.

[88] Stefan Saroiu, Krishna P. Gummadi, and Steven D. Gribble. Measuring and Analyzing the Characteristics of Napster and Gnutella Hosts. *Multimedia Systems*, 9(2):170–184, 2003.

[89] Stefan Saroiu, P. Krishna Gummadi, and Steven D. Gribble. A Measurement Study of Peer-to-Peer File Sharing Systems. In *Proceedings of the Multimedia Computing and Networking Conference (MMCN)*, San Jose, CA, January 2002.

[90] Todd Sarver. Anticipating Regret: Why Fewer Options May be Better. *Econometrica*, 76(2):263–305, 2008.

[91] Barry Schwartz. Can There Ever be too Many Flowers Blooming? *Culture Choice*, 3:1–26, 2005.

[92] Sven Seuken, Denis Charles, Max Chickering, and Sidd Puri. Market Design and Analysis for a P2P Backup System. In *Proceedings of the 11th ACM Conference on Electronic Commerce (EC)*, Cambridge, MA, June 2010.

[93] Sven Seuken, Kamal Jain, Desney Tan, and Mary Czerwinski. Hidden Markets: UI Design for a P2P Backup Application. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI)*, Atlanta, GA, April 2010.

[94] Sven Seuken and David C. Parkes. On the Sybil-Proofness of Accounting Mechanisms. In *Proceedings of the Workshop on the Economics of Networks, Systems and Computation (NetEcon)*, San Jose, CA, June 2011.

[95] Sven Seuken, David C. Parkes, and Kamal Jain. Hidden Market Design. In *Proceedings of the 24th Conference on Artificial Intelligence (AAAI)*, Atlanta, GA, July 2010.

[96] Sven Seuken, Jie Tang, and David C. Parkes. Accounting Mechanisms for Distributed Work Systems. In *Proceedings of the Conference on Artificial Intelligence (AAAI)*, Atlanta, GA, July 2010.

[97] Dan Sheldon and John Hopcroft. Manipulation-Resistant Reputations Using Hitting Time. In *Proceedings of the 5th Workshop on Algorithms for the Web-Graph (WAW)*, San Diego, December 2007.

[98] David M. Smith. The Cost of Lost Data. *Graziadio Business Report*, 6(3), 2003.

[99] Jie Tang, Sven Seuken, and David C. Parkes. Hybrid Transitive Trust Mechanisms. In *Proceedings of the 9th International Conference on Autonomous Agents and Multi-Agent Systems*, Toronto, CA, May 2010.

[100] Richard H. Thaler, Cass R. Sunstein, and John P. Balz. Choice Architecture. Unpublished. Available at SSRN: http://ssrn.com/abstract=1583509, April 2010.

[101] Amos Nathan Tversky. Rational Theory and Constructive Choice. In *Proceedings of the IEA Conference on The Rational Foundations of Economic Behavior*, Torino, Italy, 1993.

[102] U.S. Department Of Energy. Grid 2030: A National Vision For Electricitys Second 100 Years. http://www.climatevision.gov/sectors/electricpower/pdfs/electric_vision.pdf, 2003.

[103] William Vickrey. Counterspeculation, Auctions, and Competitive Sealed Tenders. *The Journal of Finance*, 16(1):8–37, 1961.

[104] Vivek Vishnumurthy, Sangeeth Chandrakumar, and Emin G. Sirer. KARMA: A Secure Economic Framework for Peer-to-Peer Resource Sharing. In *Proceedings of the 1st Workshop on Economics of Peer-to-Peer Systems (P2PECON)*, Berkeley, CA, June 2003.

[105] Léon Walras. *Eléments d'Économie Politique Pure, ou, Théorie de la Richesse Sociale (Elements of Pure Economics, or, The Theory of Social Wealth)*. L. Corbaz and Cie, Lausanne, 1874.

[106] Washington State Department of Transportation. SR 167 - 8th to 277th Southbound HOT Lane. `http://www.wsdot.wa.gov/projects/sr167/8to277/`, May 2011.

[107] Matthew Webb, Yu Mengquan, and Mark Beach. Propagation Characteristics, Metrics, and Statistics for Virtual MIMO Performance in a Measured Outdoor Cell. *IEEE Transactions on Antennas and Propagation*, 59(1):236–244, 2011.

[108] James R. Wright and Kevin Leyton-Brown. Beyond Equilibrium: Predicting Human Behavior in Normal-Form Games. In *Proceedings of the Twenty-Fourth Conference on Artificial Intelligence (AAAI)*, Atlanta, GA, July 2010.

[109] Fredrik Ygge and Hans Akkermans. Power Load Management as a Computational Market. In *Proceedings of the 2nd International Conference on Multi-Agent Systems (ICMAS)*, Kyoto, Japan, December 1996.

[110] Chao Zhang, Prithula Dhungel, Di Wu, Zhengye Liu, and Keith W. Ross. BitTorrent Darknets. In *Proceedings of the 29th Conference on Information Communications (IEEE INFOCOM)*, San Diego, CA, March 2010.