

# **Online Networks as Societies: User Behaviors and Contribution Incentives**

Lu Jia



# **Online Networks as Societies: User Behaviors and Contribution Incentives**

**Proefschrift**

ter verkrijging van de graad van doctor  
aan de Technische Universiteit Delft,  
op gezag van de Rector Magnificus prof.ir. K.C.A.M. Luyben,  
voorzitter van het College voor Promoties,  
in het openbaar te verdedigen op woensdag 30 oktober 2013 om 10:00 uur

door **LU JIA**

Master of Philosophy, The Chinese University of Hong Kong, China  
Master of Science, Harbin Institute of Technology, Harbin, China

geboren te Harbin, Heilongjiang, China

Dit proefschrift is goedgekeurd door de promotor:

Prof.dr.ir. D.H.J. Epema

Copromotor:

Dr.ir. J.A. Pouwelse

*Samenstelling promotiecommissie:*

Rector Magnificus

Prof.dr.ir. D.H.J. Epema

Dr.ir. J.A. Pouwelse

Prof.dr.ir. G.J.P.M. Houben

Dr. N. Carlsson

Prof.dr. D.M. Chiu

Dr.ir. D. Hiemstra

Prof.dr.ir. M.R. van Steen

Prof.dr. K.G. Langendoen

voorzitter

Technische Universiteit Delft

Technische Universiteit Eindhoven, promotor

Technische Universiteit Delft, copromotor

Technische Universiteit Delft

Linköping University, Sweden

The Chinese University of Hong Kong, China

Universiteit Twente

Vrije Universiteit Amsterdam

Technische Universiteit Delft, reservelid

Published and distributed by: Lu Jia

E-mail: [adele.lu.jia@gmail.com](mailto:adele.lu.jia@gmail.com)

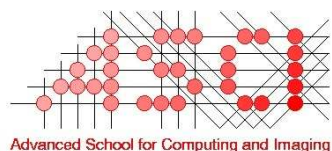
ISBN: 978-94-6186-224-2

Keywords: Online Networks, BitTorrent, Facebook, Incentives, Barter, Monetary schemes, Systemic risk, User interaction strength, Modeling, Simulation, Measurements, Implementation.

Copyright © 2013 by Lu Jia.

All rights reserved. No part of the material protected by this copyright notice may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage and retrieval system, without written permission of the author.

Printed in The Netherlands by: Wöhrmann Print Service.



*The work described in this thesis has been carried out in the ASCI graduate school. ASCI dissertation series number to be confirmed.*

*This work was supported by the Future and Emerging Technologies programme FP7-COSI-ICT of the European Commission through project QLectives (grant no.: 231200).*

*To my dearest mama*



# Acknowledgements

Generosity, enthusiasm, support, and inspiration—these were the responses that the preparation and progress of this thesis seemed most often to evoke. They made the journey to PhD possible as well as pleasurable. Here are the friends, advisors, and supporters whose particular kindnesses I would like to remember with very special thanks:

Dick, thank you for your kindness, your support, and your confidence in me. I respect your open mindness in treasuring innovative but rough ideas and your constant encouragement for them to become true. I appreciate your patience in reviewing my papers and your valuable comments about everything from the logic to the writing. I enjoy our casual talks and have been amazed by your extensive insights, whether they are about painting or music or literature. Working with you has been a very pleasant, and most importantly, inspiring experience.

Johan, thank you for your unflagging enthusiasm, your great confidence, and your eternal pursuit for a truly distributed system, without which the whole Tribler group could not have been gathered together and I could not have enjoyed my four years here. Particularly, I would like to thank you for your support in transforming my ideas into Tribler, and last but not least, for building up my vocabulary with all those fancy buzzwords.

Henk, thank you for the guidance you gave me in the early days of my PhD, which has ensured a smooth and pleasant orientation for me. I have enjoyed your stylish remarks on every aspect of life that always lighten up the atmosphere. Above all, thank you for running up the marvelous PDS group.

Dah Ming, thank you for your early enlightenment in distributed systems and your support for me pursuing a PhD, without which this thesis couldn't have become true. You have been gentle, encouraging, and kind, showing me a good example of how promising and pleasant the academia could be.

Rameez, Tamás, and Dave, the former Three Musketeers in room 09.280, thank you all for your inspirations in economics, sociology, and psychology. Rameez, I respect that, in the apparent over-rational academia, you are always willing to express yourself passionately, whether in emotional empathy or in perspicacious criticism. Tamás, I appreciate the generosity, the patience, and the kindness you have spared me in every lively discussion and every paper polishing we had together. Dave, I have enjoyed our talks and

your British accent. It's pity that you left so early, and yet because of that, your distinctive mind and behavior linger as a true legacy.

Boxun, thank you for your constant praise of me having a sense of Beijing-style humor. I have enjoyed those bizarre and yet brilliant chats we had, and I appreciate to have such a non-Chinese as you in the group.

Alex, thank you for your talkative chats that always lighten up the atmosphere, for your constant encouragement of my research and avocation, for your timely support during my downs, and above all, for setting up a good example of how energetic and fearless a person with passion could be.

Lucia, I have enjoyed every chat and gossip we had in almost everything and everyone. Thank you for your pasta recipes, for your live demonstration of making a legitimate Italian pizza with no pineapples, and for the lovely Totoro plush we made together.

Mihai, thank you for improving my English with endless debates on politics and human rights and sexism. Gradually, I have come to appreciate and enjoy them, and gradually, I have come to understand your kindness and gentleness.

Dimitra, thank you for building up my imagination and desire of those beautiful islands in Greece. Every chat we had about them served to enliven many a tedious afternoons.

Nitin, I have enjoyed the sharp and amusing comments you dropped when the group gathered together. Your vision and mind will for sure lead you to establish your own company.

Riccardo, thank you for being my office-mate at the beginning and the end of my PhD. I enjoyed very much your debates with Johan.

Rahim, thank you for your effort and patience in explaining every single detail of BarterCast to me. Good luck with the pursuit of the four wives you endorse.

Otto, you have been funny and interesting and amusing. I enjoyed our occasional conversations that were always full of jokes. Thank you for introducing Corina to me. Corina, you are such a sweet girl, and thank you for all those wonderful concerts we went together.

Siqi, Jianbin, Jie, Jong, and Kefeng, thank you for building up the Chinese invasion in this group, and for all those hilarious mornings of badminton. And to the other younger PhD students, Niels, Bogdan, and Alexey, it was nice meeting you and I wish all the best for you (and for your princess as well, Bogdan).

Boudewijn, thank you for your support and patience in our collaboration. Those long and exhausting discussions and experiments were in the end worth it.

Arno, thank you for your support in Tribler that made every experiment we did possible, and also for the book *By Horse to Jerusalem*. I enjoyed it very much.

Paulo, Munire, and Stephen, thank you for your always enthusiastic salute, for your constant technical support, and for ordering me a red Dell as I wanted. Ilse, Rina,



Monique, and Esther, thank you for taking care of the administrative issues and the daily routines for the whole group.

To the friends outside the PDS group: Peiyao, you have been such a sweet sister. You remind me of a younger version of myself and watching every step you take to your dreams is an exciting as well as serene experience. Yukun, Xiaogang, Dan, Xin, Zheng, thank you for all those fantastic dinners and wonderful trips we had together. Ramy, thank you for your inspiring horseback riding lessons. Xiaoyu, thank you for your lively piano instructions and demonstrations. Changyin and Changbin, thank you for your hospitality in London, and for helping a Les Mis fan's dream come true. Without your accompany, my life here couldn't have been this lively and colorful.

Finally, my special thanks go to my beloved family. Mama, thank you for always encouraging me to explore the world, for respecting and supporting my decisions, for never losing faith in me realizing my dreams, and above all, for raising me up as who I am. Shiming, my better half, thank you for your gentleness, patience, understanding, and above all, for sharing a lifetime of adventure with me.



# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	A brief history of online networks . . . . .	2
1.2	Grounding our work . . . . .	3
1.2.1	BitTorrent . . . . .	3
1.2.2	Facebook . . . . .	4
1.2.3	Tribler . . . . .	5
1.3	Research context: the Qlectives project . . . . .	6
1.4	User behaviors and contribution incentives in online networks . . . . .	7
1.4.1	User behaviors and their consequences . . . . .	7
1.4.2	Barter schemes and their limitations . . . . .	8
1.4.3	Monetary schemes and their risks . . . . .	9
1.4.4	User interactions and their implications . . . . .	11
1.5	Problem statement . . . . .	12
1.6	Contribution and thesis outline . . . . .	13
<b>2</b>	<b>Balancing reciprocity and inequality in BitTorrent</b>	<b>17</b>
2.1	BitTorrent overview . . . . .	18
2.1.1	The swarm . . . . .	18
2.1.2	The Tit-For-Tat incentive policy . . . . .	18
2.2	A Fluid Model for BitTorrent . . . . .	19
2.2.1	The basic idea . . . . .	19
2.2.2	Model description . . . . .	19
2.2.3	Bandwidth allocation . . . . .	21
2.2.4	Model Validation . . . . .	23
2.3	Analysis . . . . .	24
2.3.1	Enhancing reciprocity . . . . .	25
2.3.2	Reducing inequality . . . . .	27
2.3.3	Seeding policies . . . . .	31
2.4	Related work . . . . .	31
2.5	Conclusions . . . . .	33

---

<b>3</b>	<b>Modeling and analysis of Sharing Ratio Enforcement</b>	<b>35</b>
3.1	A simple model for Sharing Ratio Enforcement . . . . .	36
3.1.1	Preliminary: rational user behavior . . . . .	36
3.1.2	The basic model . . . . .	37
3.1.3	Within one swarm . . . . .	37
3.1.4	Within multiple swarms . . . . .	39
3.2	Analysis . . . . .	41
3.2.1	One swarm . . . . .	41
3.2.2	Multiple swarms . . . . .	45
3.2.3	Comparison with the real world . . . . .	45
3.3	Related work . . . . .	49
3.4	Conclusions and future work . . . . .	49
<b>4</b>	<b>Monetary schemes as contribution incentives: systemic risk and user-level performance</b>	<b>51</b>
4.1	Support from real world observations . . . . .	53
4.1.1	The existence of over-seeding behavior . . . . .	53
4.1.2	The oversupply . . . . .	54
4.1.3	Unproductive seeding . . . . .	55
4.2	Model description . . . . .	56
4.2.1	Credit-based versus SRE-based policies . . . . .	56
4.2.2	The basic model . . . . .	57
4.3	System-level performance: the systemic risk . . . . .	58
4.3.1	The crash and crunch . . . . .	58
4.3.2	Predicting crashes and crunches . . . . .	61
4.3.3	Adaptive credit for sustainability . . . . .	62
4.4	User-level performance: the positive and negative effects of SRE . . . . .	65
4.4.1	A simple model . . . . .	65
4.4.2	The effects of SRE . . . . .	69
4.4.3	Description of proposed strategies . . . . .	73
4.4.4	Strategy evaluation . . . . .	75
4.5	Dynamic file popularity . . . . .	77
4.5.1	Experimental setup . . . . .	78
4.5.2	The effects of SRE under dynamic file popularity . . . . .	79
4.5.3	Proposed strategies under dynamic file popularity . . . . .	82
4.6	Discussion: The change of user behavior? . . . . .	85
4.7	Related work . . . . .	86
4.8	Conclusion . . . . .	87

---

<b>5</b>	<b>User behaviors under contribution incentives: a measurement study</b>	<b>89</b>
5.1	Methodology . . . . .	90
5.2	The rich and the poor: positive and negative effects of SRE . . . . .	92
5.2.1	A general view . . . . .	92
5.2.2	Long seeding time, even for the poor . . . . .	93
5.2.3	Possible reasons? . . . . .	95
5.2.4	Why the poor are poor and how to become rich? . . . . .	97
5.3	The old and the new: how users evolve . . . . .	102
5.3.1	A general view . . . . .	103
5.3.2	Level of commitment: on the way to become more committed . . . . .	103
5.3.3	Average upload speed: the new have not yet suffered . . . . .	103
5.3.4	Sharing ratio: the spreading new and the conservative old . . . . .	105
5.4	The big and the small consumer: active users active in all . . . . .	105
5.5	The gamer and the dedicator: Gemini of the private community . . . . .	108
5.5.1	Seeding and leeching time: which decides a gamer? . . . . .	108
5.5.2	Sharing ratio: gamers seize the day . . . . .	109
5.5.3	Average upload speed: gamers know the way . . . . .	109
5.6	Discussion . . . . .	110
5.7	Related work . . . . .	111
5.8	Conclusion . . . . .	112
<b>6</b>	<b>Estimating user interaction strength in online networks</b>	<b>113</b>
6.1	User interactions . . . . .	114
6.2	Problem statement . . . . .	115
6.3	Design description . . . . .	116
6.3.1	Representing user interaction history . . . . .	117
6.3.2	Estimating user interaction strength . . . . .	118
6.3.3	Maintaining security . . . . .	119
6.3.4	Incorporating into distributed systems . . . . .	120
6.4	Interaction pattern detection . . . . .	121
6.4.1	Applying to interaction pattern detection . . . . .	121
6.4.2	Results . . . . .	121
6.5	Distributed online time estimation . . . . .	124
6.5.1	Applying to distributed online time estimation . . . . .	124
6.5.2	Implementing into Tribler . . . . .	125
6.6	Simulation . . . . .	126
6.6.1	Basic simulation model . . . . .	126
6.6.2	Accuracy . . . . .	127
6.6.3	Accuracy under partial information . . . . .	129

6.6.4	Scalability . . . . .	130
6.7	Emulation . . . . .	131
6.7.1	Emulation setup . . . . .	131
6.7.2	Synthetic user behavior . . . . .	131
6.7.3	Trace-based user behavior . . . . .	132
6.7.4	Preparing for the real world: strategies for reducing the workload	134
6.8	Real-world deployment . . . . .	135
6.8.1	Deployment techniques . . . . .	135
6.8.2	Evaluation . . . . .	136
6.9	Related work . . . . .	140
6.10	Conclusion . . . . .	141
<b>7</b>	<b>Conclusion</b>	<b>143</b>
7.1	Conclusions . . . . .	143
7.2	Suggestions for future work . . . . .	145
7.3	Epilogue . . . . .	146
	<b>Bibliography</b>	<b>148</b>
	<b>Summary</b>	<b>159</b>
	<b>Samenvatting</b>	<b>163</b>
	<b>Biography</b>	<b>167</b>

# Chapter 1

## Introduction

Online networks are complex distributed computer systems that involve potentially large numbers of humans with their collective inputs and decisions. Typical examples of online networks include email, Facebook, LinkedIn, Wikipedia, eBay, and BitTorrent-like Peer-to-Peer (P2P) systems. They have become popular and powerful infrastructures for communication and they provide various mechanisms for users to interact. For instance, in Facebook, users post messages on each other's walls and comment on each other's photos; in Wikipedia, users collectively edit articles in their areas of expertise; and in BitTorrent, users upload to and download from each other to share the contents of their common interests. As in these examples, online networks often rely on the cooperation and the contribution of their users. Nevertheless, users in online networks are often found to be selfish, strategic, or even malicious, rather than cooperative, and therefore they need to be incentivized for contributions. In this thesis, we aim to study the user behaviors and the contribution incentives—two underpinnings—in online networks. Whereas computer scientists often treat online networks purely as computer systems, in this thesis, we take one step further and approach online networks as societies as well. With this approach, we hope that on the one hand computer scientists will be motivated to think about the similarities between their artificial computer systems and the natural world, and on the other hand, that people outside computer science will understand online networks better.

With the astounding growth of user activity in online networks, these systems have deeply impacted and even changed the offline human society. For instance, online social networks like Facebook have provided a novel and often addicting context for people to build up their social ties, online social media like Twitter have vastly accelerated the information propagation to a level that can never be reached by traditional media, and P2P systems like BitTorrent have demonstrated a genuine and efficient method for sharing contents that benefits both service providers and consumers. At the same time, if we take a closer look we may find that online networks have also self-evolved into *online societies*. They possess elementary components similar to the ones often observed in the offline human societies. Users in online networks behave differently, just like the pop-

ulation in human society is composed by people with various behaviors. Organizations that maintain online networks play a similar role as governments and they face a variety of concerns, for example, how to enhance reciprocity and how to reduce inequality. The contribution and consumption of users in online networks form the supply and the demand, and as in economies of societies, they need to be closely monitored and carefully balanced. Particularly, with respect to the topic of this thesis—user behaviors and contribution incentives—many attempts have been made to design incentive policies, which range from *barter schemes* to *monetary schemes*, leading online networks into *barter economies* or *money economies*. Similar to the barter and money economies developed in human society, in online networks they both have their own effectiveness and limitations.

In this thesis, we provide theoretical and practical insights into the correlation between user behaviors and contribution incentives in online networks. We demonstrate user behaviors and their consequences at both the system and the individual level, we analyze barter schemes and their limitations in incentivizing users to contribute, we evaluate monetary schemes and their risks in causing the collapse of the entire system, and we examine user interactions and their implications in inferring user relationships. We often use BitTorrent and Facebook as case studies due to their popularity in today's Internet. In the end, the key research question we address is to create an understanding of the correlation between user behaviors and contribution incentives that is both generic and can be applied to a wide range of online networks that rely on the cooperation and the contribution of their participants.

## 1.1 A brief history of online networks

Online networks have emerged, developed, and proliferated with the *Internet*. Since the mid 1990s, the Internet has had a tremendous impact on our daily lives: the near instant communication supported by email and instant messaging has nearly reformed our daily contacts, online shopping sites have completely changed our purchase habits, online discussion forums have provided us with novel and interactive contexts for obtaining and polishing knowledge, online social networks have assisted us in building up both online and offline social ties, and online social media have accelerated the information propagation to a level that traditional media can never reach. While these online networks bring our lives to a whole new level, they often rely on central servers to provide resources and services, for example, caching emails, streaming videos, and retrieving profiles of online friends. With the astounding growth of online networks, service providers like YouTube and Facebook are facing significant challenges in satisfying users with enormous resources including bandwidth, storage space, and computing power.

As a more efficient way for obtaining and sharing resources, Peer-to-Peer (P2P) systems made their debut through the Napster file sharing system [75] in the late 1990s. Later,



Kazaa [59], Gnutella [2], eDonkey [42], BitTorrent [19], and Tribler [84] have emerged one after another, aiming to provide file sharing and streaming services with better performance through designing effective contribution incentives. Besides content sharing, P2P systems serve many other applications as well, such as Dalesa [23] for web caching, Yacy [118] for searching, Spotify [100] for music distributing, and Bitcoin [8] for digital currency. P2P systems at various times serve hundreds of millions of users. They are founded on the principle that the resources and services from each individual participant can be leveraged for the good of others, and thus, release servers from tremendous workloads and save service providers from immense expenses.

## 1.2 Grounding our work

In order to analyze user behaviors and contribution incentives in online networks, we ground our work in three classes of online networks, one with P2P services that are operated in a distributed manner, one with remarkable social features that rely on central servers, and one that combines social network features within distributed operations. In particular, we focus on three online networks, BitTorrent, Facebook, and Tribler. Below, we give a brief overview of each of them in turn.

### 1.2.1 BitTorrent

In 2001, the BitTorrent file sharing protocol [19] was released together with the first BitTorrent client. A BitTorrent network consists multiple *swarms*, each associated with a file divided into small *chunks* and with a number of users who are interested in this file. Users are located to swarms and are introduced to each other by central servers named *trackers*. Downloading in BitTorrent is not performed in a sequential order, so that users in the same swarm may have different chunks of the file and they can exchange what they have with each other. In BitTorrent's terminology, users who do not have all the file chunks are called *leechers* and users who already have the entire file and only stay to serve others are called *seeders*. Figure. 1.1 displays an overview of a BitTorrent network with trackers, files, swarms, leechers, and seeders. By now, BitTorrent has become the most popular P2P protocol, based on which hundreds of P2P clients have been developed, for instance, uTorrent [105], Vuze [111], and Tribler [84]. With their own extensions and improvements, these clients serve hundreds of millions of users with various applications ranging from the original file sharing to live streaming and video-on-demand.

BitTorrent does not provide content search service. Instead, the search function is provided by BitTorrent websites or communities like the PirateBay [80] and TvTorrents [104]. These communities also provide additional functions such as discussion forums and content moderation. While some BitTorrent communities, for example, the PirateBay,

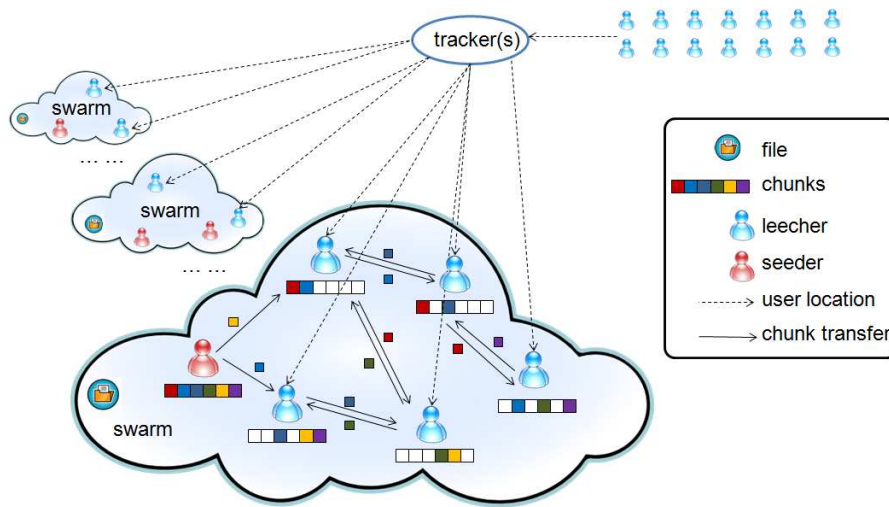


Figure 1.1: Overview of a BitTorrent network with files divided into chunks, swarms, leechers, seeders, and trackers.

are publicly available to all users, *private BitTorrent communities* require users to keep accounts that sometimes can only be obtained through invitations [10, 12, 15, 40, 41, 81, 104]. These communities adopt community-level policies, such as Sharing Ratio Enforcement (SRE), to incentivize users to contribute. Under SRE, members are required to maintain their sharing ratios (the ratio between a user’s upload and download amount) at least equal to a threshold set by the community administrator, otherwise they are banned from downloading or even expelled from the community. According to a recent study, private communities serve over 24 million active users combined, and are responsible for the majority of the BitTorrent activity in the world [124].

BitTorrent communities are playing such an important role in today’s Internet that they serve as ideal cases for studying user behaviors and contribution incentives. Particularly, the user-level identifications maintained in private BitTorrent communities, in contrast to the IP-level identifications in public communities, allow us to study the real *user* behaviors, rather than the superficial *peer* behaviors compromised in most previous works.

## 1.2.2 Facebook

Facebook is no doubt the most successful online social network in today’s Internet. Founded in February, 2004, Facebook originally served as a social toolkit for only Harvard students, and eight years later, it has attracted over one billion users all over the world. Facebook has had such a great influence on humans’ daily lives that researchers from various backgrounds, for example, sociologists, psychologists, economists, physi-

cist, and computer scientists, have spent tremendous effort in analyzing the *Facebook phenomenon* [20, 29, 88, 108, 114].

In Facebook, each user maintains a profile with photos, social interests, education experiences, and other personal information. Besides demonstrations through user profiles, users in Facebook can communicate with friends and other users through private or public *messages* and a *chat* feature. They can also create and join interest groups and vote *like pages*, some of which are maintained by organizations as a means of advertising. Multiple users can be gathered together through *Events* and *Games*. And as one of the most straightforward ways of sharing social lives, users can comment on the news, the photos, the videos, and the essays posted by their online friends.

With its vast popularity, Facebook serves as an ideal case for studying user behaviors and contribution (or participation) incentives. In addition, as mentioned above, Facebook provides abundant opportunities for users to interact. User interaction is a key aspect of user behavior. It is essential for inferring user relationships that can be further utilized for security enhancement, for cooperation promotion, for item recommendation, and most importantly, for contribution incentives.

### 1.2.3 Tribler

Tribler [84] is a fully distributed open-source online network for media and social applications like file sharing, live streaming, video-on-demand, content searching, voting, and interest-based channels. It is the research vehicle for research in P2P related topics in the Parallel and Distributed Systems group of TU Delft, where the research for this thesis has been carried out. Since its first release in 2006, the Tribler client has been downloaded over a million times.

Tribler uses the BitTorrent protocol for P2P file sharing and the Libswift protocol [77] for P2P streaming. Libswift is an IETF (Internet Engineering Task Force) standard protocol proposed by the Tribler group. In addition, Tribler provides various advanced features including a distributed reputation mechanism named BarterCast [71], a distributed service for content discovery supported by a dissemination and database synchronization protocol named Dispersy [123], and an advanced user interface that psychologically motivates users to contribute. Furthermore, a mechanism for estimating user interaction strength—which is presented as part of this thesis—has been integrated into the Tribler client.

To identify users across sessions, Tribler assigns each user a permanent identifier which, as in private BitTorrent communities, allows us to analyze the real user behaviors. On the other hand, unlike private BitTorrent communities where file sharing is the only service, Tribler provides socially enhanced applications such as interest-based channels, in which users can, in addition to upload and download, reply to other users' comments, vote the contents they like, and report suspicious contents such as spam. In other words,

Tribler serves as an online social network and an online P2P file sharing network combined, and thus, guarantees a promising environment for experimenting with user behaviors and contribution incentives.

Based on the insights gained from high-level models and theories, various algorithms have been designed, implemented, and deployed into Tribler. Particularly, Rahman *et al.* [90] have analyzed effort-based reciprocity and the sustainability of credit systems. They have also proposed a Design Space Analysis (DSA) for modeling contribution incentives that provides a tractable analysis of competing protocol variants within a detailed design space [91]. Capotă *et al.* [11] have developed an analysis of present methods for resource allocation in multimedia communities. D'Acunto *et al.* [21, 22] have analyzed strategies for the peer selection, the piece selection, and the bandwidth allocation in BitTorrent-like Video-on-Demand systems. Zeilemaker *et al.* [122] have introduced Open2Edit, an application with similar quality and activity as Wikipedia but implemented without central servers. Meulpolder *et al.* have analyzed the bandwidth allocation in BitTorrent [72] and proposed BarterCast [71], a decentralized reputation system, with real world implementation. Delaviz *et al.* have further proposed improvements on the accuracy and the coverage of BarterCast [25] and have designed a sybil-resistant approach [26]. Gkorou *et al.* [34] have developed a scheme for reducing the amount of history maintained in decentralized interaction-based reputation systems.

### 1.3 Research context: the Qlectives project

The research in this thesis has been carried out in the context of the European Framework Program 7 project *Qlectives* [86]. Qlectives is a project bringing together social modelers, Peer-to-Peer engineers and physicists to understand, to experiment, to design, and to build cooperative socially intelligent information systems. The aim of Qlectives is to combine three recent trends within information systems. The first of these is social networks, in which people link to others over the Internet to gain value and facilitate collaboration (e.g., Facebook). The second is peer production, in which people collectively produce informational products and experiences without traditional hierarchies or market incentives (e.g., Wikipedia). The third is Peer-to-Peer systems, in which software clients running on user machines distribute media and other information without a central server or administrative control (e.g., BitTorrent).

For the past four years, researchers in Qlectives have put great effort into topics covered by Qlectives. Research has been carried out on a wide range of subjects. To name a few examples, Medo *et al.* [68] have introduced a model for the growth of information networks that produces various degree distributions including those that are observed in important real systems such as scientific citation data or the World Wide Web. Wu *et al.* [115] have adopted an evolutionary game-theoretical approach to study social norms

and social phenomena involving cooperation or conflict. They have shown that cooperation is not such a strong social dilemma, and that it emerges and becomes stable over a fairly large range of model parameters and implementation details. Roca *et al.* [92, 93] have developed and implemented models, applying dynamic social impact theory, on analyzing the influence of trust and the emergence of cooperation.

## 1.4 User behaviors and contribution incentives in online networks

Online networks often rely on the cooperation and the contribution of their users. Nevertheless, users in online networks, like humans in human society, are often found to be selfish, strategic, or even malicious, rather than cooperative, and therefore need to be incentivized for contributions. Many attempts have been made to design incentive policies, which range from *barter schemes* to *monetary schemes*. Below, we will give a brief introduction on user behaviors and their consequences at both the system and the individual level, on barter schemes and their limitations in incentivizing users to contribute, on monetary schemes and their risks in causing the collapse the whole system, and on user interactions and their implications in inferring user relationships.

### 1.4.1 User behaviors and their consequences

Users in online networks, like humans in human society, are not always cooperative. Thus, without proper incentives, any system that requires the cooperation and the contribution of its participants potentially faces the *Tragedy of the Commons*, a social dilemma frequently occurring in human society [38]. The Tragedy of the Commons refers to the depletion of a shared resource, such as a public good, by individuals acting independently and rationally according to their self-interests, despite their understanding that depleting the common resource is contrary to the group's long-term best interests.

While the Tragedy of the Commons describes a group-level social dilemma on cooperation, the *Prisoner's Dilemma* [82] scales it down to two individuals. It refers to a situation in which two prisoners could cooperatively deny their crime and serve a moderate amount of time in prison, but in the end, to avoid being betrayed and facing the longest jail time alone, they both confess and serve the second longest time. Prisoner's Dilemma is now a canonical example of a game analyzed in game theory that shows why two individuals might not cooperate, even if it appears to be in their best interest to do so.

Online networks face these dilemmas as well. Taking BitTorrent for example, users are supposed to upload to and download from each other, and collectively achieve their common goal of obtaining a file. As it turns out, besides being cooperative, users in

BitTorrent can be lazy, strategic, or even malicious. Lazy users follow the original BitTorrent protocol, upload while downloading, but are reluctant to stay and to serve more users once their downloads are finished [71]. Strategic users, on the other hand, explore the protocol, tune parameters like the upload speed, or even modify the original protocol, so as to download the file with contributions as small as possible [62]. Finally, there are also malicious users aiming at destroying the whole system through, for example, spreading broken or fake file chunks [27]. These non-cooperative user behaviors cause severe consequences, ranging from the inefficiency to the fall of the whole system. There has been a number of studies demonstrating non-cooperative user behavior in P2P systems, from the early Gnutella network with nearly 70% users sharing nothing [95, 96], to BitTorrent networks with more than 80% users going offline immediately after they finish their downloads [83].

Online social networks also face potential manipulations from strategic or malicious users, for example, through the *Sybil Attack* [28, 109]. A large number of Sybil accounts have been found in Facebook [33], Twitter [101], and Renren [119]. Under a Sybil Attack, the attacker first generates multiple sybils with fake identities. Then, together they distribute false information to promote their status in the online network. Finally, from the forged reputation they gain undeserved advantages like being served without any contribution, or perform unjust commercial promotions like spreading spam.

To maintain a sustainable system, policies to promote cooperation and contribution are essential. While there are many previous work [27, 28, 55, 109, 120, 121] on analyzing malicious user behaviors and enhancing security, in this thesis we focus on lazy and strategic users, and we study mainly the schemes to incentivize them to contribute. To date, there has been a great effort on proposing, analyzing, and improving contribution incentives. These mechanisms range from *barter schemes* to *monetary schemes*. Below, we will discuss them in turn.

### **1.4.2 Barter schemes and their limitations**

In human society, barter refers to a system of exchange by which goods or services are exchanged for other goods or services without using a medium of exchange such as money [76]. Barter can be bilateral or multilateral. Bilateral barter implies direct exchanges of goods between two participants and therefore represent *direct reciprocity*. Multilateral barter, on the other hand, involve multiple participants in *indirect reciprocity*. Barter has several limitations, for instance, the need for the coincidence of wants from participants and the lack of a standard value of the exchanged good.

A number of online networks utilize barter schemes, or reciprocity, to incentivize users to contribute. For example, in P2P systems with no central service providers, users barter their contributions with others so as to be served. As in human society, barter

schemes in online networks also have a number of limitations. BitTorrent incorporates an incentive mechanism named Tit-For-Tat (TFT) that is based on bilateral barter-like direct reciprocity. Under TFT, users prefer uploading to users who have contributed to them in the past at the highest speeds. TFT works reasonably well in fostering barter, or cooperations, among users that are downloading the same file. Nevertheless, TFT does not provide incentives for users to remain in the system and to serve others after their downloads are finished, since there is no coincidence of wants anymore. Therefore, users are free to engage in “Hit and Run” behavior, leaving immediately upon completing their downloads. Further, TFT in BitTorrent is not an exact bilateral barter strictly with a tit for a tat. Instead, users barter with those that have reciprocated the most. Therefore, the value for one “tit” is constantly changing, which leads to possible manipulations from strategic users through, for example, only providing upload bandwidth high enough to be uploaded to, but at the same time low enough to be the lowest among all the chosen users [62, 65, 78]. Last but not least, it has been shown that in BitTorrent systems, it is unlikely that two users encounter each other often enough, and therefore TFT-like direct reciprocity is not effective for long-term use [79].

To tackle these issues, various remedies have been proposed and the primary idea is to include indirect reciprocity [60, 71, 79]. Indirect reciprocity occurs when, for example, after user A contributes to user B who further contributes to user C, user C rewards user A based on their indirect relationship. A major problem arising from indirect reciprocity is the *trust* issue [1, 35]. It is obvious that, in the former example, if user A and user B collude and exaggerate the maybe-not-existing contribution from A to B, user A can potentially get a reward from user C with no actual contributions. Similarly, if user B is malicious and deliberately disguises user A’s contribution, user A may not get a reward from user C even if he has contributed. As a consequence, contribution incentives with indirect reciprocity often rely on instantaneous communications with a secure third party [79]. However, the reliability of requiring a secure third party is questionable, and moreover, it runs counter to the open membership that underlies the success of these systems in the first place. Another approach to tackle the trust issue is to diminish the amount of potentially false information [71], however, this often relies on algorithms with high complexity that are unrealistic to be used in real systems.

In this thesis, we provide a theoretical model for BitTorrent’s TFT incentive policy and its variations, aiming to provide some insights into how users in BitTorrent allocate their bandwidth, i.e., make barter, with others.

### **1.4.3 Monetary schemes and their risks**

In human society, a monetary system refers to a system of exchange by which goods or services are exchanged using money as a medium [31]. In contrast to barter, monetary

schemes do not require a coincidence of wants, and therefore tackle the problems of barter schemes in both human society and online networks. Under a monetary scheme, through contributions users earn credits, or more precisely, virtual money, that can be used in their later consumptions. Thus, users are incentivized to contribute, so as to keep some reserve for their future needs.

Monetary schemes are becoming prevalent in nowadays online networks, ranging from the currencies used in online virtual worlds such as Second Life, to methods for resource allocation in wireless sensor network [17] and file sharing [43]. Several monetary schemes for contribution incentives have been proposed and analyzed in the literature [37, 39, 56, 73, 89, 107]. As in human society, measures are taken to avoid monetary risks including inflation, deflation, and systemic risk that may cause the collapse of the entire economic system [58]. For instance, Vishnumurthy *et al.* [107] have presented a system involving a virtual currency named Karma, in which sets of bank nodes keep transaction balance of users. Karma captures the amounts of resources a user has contributed and consumed. To avoid inflation and deflation, the level of per-capita Karma in the system, i.e., the total Karma divided by the number of active users, is constantly monitored and maintained. Several researchers [37, 56, 89] have further shown that, in a similar scrip system where users can consume and produce services, both an abundance of money supply and a shortage of it lead to inefficiency. An oversupply of money leads to a *crash* in which users hold abundant money and are not willing to contribute. Conversely, an undersupply of money leads to a *crunch* in which users go broke and cannot afford to consume any services.

A good example of utilizing monetary schemes in distributed online networks is private BitTorrent communities that aim at providing incentives beyond BitTorrent's original Tit-For-Tat. To do so, these communities employ private trackers that maintain centralized accounts and record the download and upload activity of each user. They apply community-level policies to incentivize good overall upload / download behavior. One such policy is credit-based, in which each user is required to maintain a positive credit (its upload amount minus its download amount). Another such policy, as we introduced earlier, is Sharing Ratio Enforcement (SRE), in which each user is required to keep his sharing ratio at least equal to a threshold. Under both policies, community members who cannot fulfill the requirements are banned from downloading or even expelled from the community. Thus, it is guaranteed that each user performs some contribution.

The credit-based policy is an obvious monetary scheme. SRE, though not quite at first glance, only differs in that it allows users to have negative credit (i.e., their download amounts larger than their upload amounts) and that when users do have negative credit, the amount of credit circulated in the system is dynamic. Under both schemes, contributions are strongly incentivized. Nevertheless, monetary schemes induce systemic risk that may cause the collapse of the entire system. In this thesis, we explore both the system-level



dynamics and the user-level performance in communities adopting such policies.

#### 1.4.4 User interactions and their implications

User interaction is a key aspect of user behavior and is the underpinning of any contribution incentive in any online network, for without which the users cannot be “connected” and the network cannot be “netted”.

Online networks provide various mechanisms for users to interact. For instance, in Facebook, users maintain friendships, post messages on their friends’ walls, and comment on their friends’ photos. In Wikipedia, users collectively edit articles in their areas of expertise, and in BitTorrent, users upload to and download from each other to share the contents of their common interests. The patterns and strengths of user interactions are prominent, and they are useful for a variety of applications.

Previously, a number of applications [102, 103, 120, 121] leveraged *online friendships* to incentivize users to contribute, to enhance security, to promote cooperation, to improve item recommendation, etc. Nevertheless, it has long been observed that low-interaction friendships, as exemplified by the “Familiar Stranger” [106], are prevalent, and that the dynamics of user interactions is more representative for inferring user relationships [108, 114] than simple, statically established “binary” friendships. Therefore, in these applications, users will be much better off by estimating their interaction strengths with others and by giving high ranks to the ones with whom they have interacted frequently. As another example, in P2P systems, user interactions form the foundation for designing incentive policies. Through estimating the interaction strengths between users in terms of the amounts or durations of contributions, system designers can make users favor the highly ranked users for future consumptions.

The importance of user interactions in online networks leads to the question: *How can we estimate user interaction strength?* Previous work addressing this issue [18, 108, 114, 116] has focused only on online social networks like Facebook, and has only considered *binary* user interactions, simply indicating whether a user has interacted with another user or not. In contrast, we propose a User Interaction Strength Estimation scheme called UISE that has a much more fine-grained notion of user interaction, that is applicable to a more general category of online networks, and that can be easily applied to distributed systems and therefore achieves a scalable design. In the end, UISE serves as a general framework for expressing user interactions and their strengths that is both generic and can be applied to a wide range of online networks.

## 1.5 Problem statement

The research problem that we address in this thesis is how user behaviors and contribution incentives in online networks are correlated. The research questions we answer are as follows:

**What are the characteristics and principles behind BitTorrent’s Tit-For-Tat incentive policy and its variations?** Tit-For-Tat was originally designed for sharing files in resource-constrained scenarios. The current *BitTorrent ecosystem*, however, is becoming *over-provisioned* and is enriched with new applications like video streaming. When the resource is abundant or when there are real time constraints, always enhancing reciprocity is system-wide inefficient and therefore a balance between reciprocity and inequality is necessary. To this end, it is important to understand the characteristics and the principles behind BitTorrent’s Tit-For-Tat and its variations.

**Is there a theoretical validation for the effectiveness of monetary schemes such as Sharing Ratio Enforcement in incentivizing users to contribute?** Several measurement studies have shown that SRE is very effective in boosting cooperation [13, 64, 69, 124]. For instance, [69] reports seeder-to-leecher ratios that are at least 9 times higher than in public BT communities, while download speeds are measured to be 3–5 higher. Therefore, it would be beneficial to analyze *how* SRE actually provides seeding incentives and to quantify the expected performance improvement in terms of the download speed.

**What are the risks of using monetary schemes as contribution incentives?** Monetary schemes such as SRE and credit-based policies have been proven in the real world to be effective in incentivizing users to contribute. Nevertheless, they require delicate designs without which, as in any monetary system, they induce systemic risks that may cause the collapse of the entire system. To maintain a sustainable system, it is important to analyze the system-level dynamics and the potential risks under such schemes. Moreover, even when systemic risks do not occur, this only ensures that the system is able to function, but not how well it functions. Thus, a further analysis of user-level performance under such schemes is essential.

**How do users behave under contribution incentives?** The dedication of users is not the only behavioral change observed in communities with contribution incentives. The same incentive policy may trigger different or even opposite user behaviors. The various reactions from users are good indicators for further improvements of incentive policies. Therefore, it is necessary to explore user behaviors, to argue the reasons for these behaviors, and to demonstrate both the positive and negative effects of these behaviors.

**Can we find a generic framework for estimating user interaction strength?** User interaction is a key aspect of user behavior and is the underpinning for contribution incentives and many other applications. Therefore, it is beneficial to design a generic

---

framework for estimating user interaction strength. A promising design should have a fine-grained notion of user interactions and should be general enough to be applicable in a large range of systems. Further, with the astounding growth of online networks—with Facebook exceeding a billion users and BitTorrent serving hundreds of millions of users—a scalable design in a distributed manner is preferred.

## 1.6 Contribution and thesis outline

The contributions of this thesis are as follows:

**Balancing reciprocity and inequality in BitTorrent (Chapter 2)** We present a fluid model of BitTorrent’s Tit-For-Tat incentive policy and its variations. Our model effectively captures the bandwidth allocation between users under different incentive policies, based on which we explore strategies that influence the balance between reciprocity and equality among users. Our study shows that (i) reducing inequality leads to a better overall system performance, and (ii) the behavior of seeders influences whether reciprocity is enhanced or inequality is reduced. This chapter is largely based on our paper [48]:

A.L. Jia, L. D’Acunto, M. Meulpolder, J.A. Pouwelse, and D.H.J. Epema. BitTorrent’s dilemma: Enhancing reciprocity or reducing inequity. In *Proceedings of the 7th IEEE Consumer Communications and Networking Conference (CCNC’11)*, 2011.

**Modeling and analysis of Sharing Ratio Enforcement in private BitTorrent communities (Chapter 3)** We provide a theoretical model to analyze how Sharing Ratio Enforcement (SRE) provides seeding incentives and how SRE influences the download performance of users. Specifically, we study the influence of the SRE threshold and the bandwidth heterogeneity of the users. Under the assumption that users are rational, i.e., they seed only the minimum amount required by SRE, we show that the download speed as predicted by our model represents a lower bound for the actual speed that can be reached in the real world. This chapter is largely based on our paper [47]:

A.L. Jia, L. D’Acunto, M. Meulpolder, and J.A. Pouwelse. Modeling and analysis of sharing ratio enforcement in private BitTorrent communities. In *Proceedings of the IEEE International Communications Conference (ICC’11)*, 2011.

**Monetary schemes as contribution incentives: systemic risk and user-level performance (Chapter 4)** We analyze the performance of online networks adopting monetary schemes as contribution incentives from both the system-level and the user-level perspectives. We show that both credit-based and sharing ratio enforcement policies can lead to system-wide *crunches* or *crashes* where the system seizes completely due to too little or to too much credit, respectively. We explore the conditions that lead to these system pathologies, we present a theoretical model that predicts if a community will eventually crunch or crash, and we design an adaptive credit system that automatically adjusts credit policies to maintain the sustainability. We further analyze the user-level performance by

studying the effects of oversupply. We show that although achieving an increase in the average downloading speed, the phenomenon of oversupply has three undesired effects: long seeding times, low upload capacity utilizations, and an unfair playing field for late entrants into swarms. To alleviate these problems, we propose four different strategies that have been inspired by ideas in social sciences and economics. We demonstrate their effectiveness through simulations. This chapter is largely based on our papers [49, 51]:

A.L. Jia, R. Rahman, T. Vinkó, J.A. Pouwelse, and D.H.J. Epema. Systemic risk and user-Level performance in private P2P communities. In *IEEE Transactions on Parallel and Distributed Systems*, online, 2012.

A.L. Jia, R. Rahman, T. Vinkó, J.A. Pouwelse, and D.H.J. Epema. Fast download but eternal seeding: the reward and punishment of sharing ratio enforcement. In *Proceedings of the 11th International Conference on Peer-to-Peer Computing (P2P'11)*, 2011.

**User behaviors under contribution incentives: a measurement study (Chapter 5)** Taking private BitTorrent communities as an example, we explore user behaviors under contribution incentives. We argue the reasons for these behaviors and we demonstrate both the positive and negative effects of these behaviors. We show that, as predicted by our model, under SRE users seed for excessively long times to maintain required sharing ratios, though their seedings are often not very productive. And as users evolve in the community, some become more attached in terms of higher ratios of the seeding and the leeching time, and some game the system by keeping risky low sharing ratios while leeching more often than seeding. Based on these observations, we analyze strategies that alleviate the negative effects of these user behaviors from both the user's and the community administrator's perspective. This chapter is largely based on our papers [46, 50]:

A.L. Jia, X. Chen, X. Chu, J.A. Pouwelse, and D.H.J. Epema. How to survive and thrive in a private BitTorrent community. In *The 14th International Conference on Distributed Computing and Networking (ICDCN'13)*, 2013.

A.L. Jia, X. Chen, X. Chu, J.A. Pouwelse, and D.H.J. Epema. User behaviors in private BitTorrent communities. *Under review*.

**Estimating user interaction strength in online networks (Chapter 6)** To date, several theoretical, centralized schemes for estimating user interaction strength have been proposed. Here we present the design, deployment, and analysis of the UISE scheme for User Interaction Strength Estimation for both centralized and decentralized online networks. Among the strong points of UISE is that it captures direct and indirect user interactions, that it scales with only partial information dissemination in decentralized systems, and that it provides disincentives for malicious user behaviors. We apply UISE to detect user interaction patterns based on wall posts in Facebook and we derive patterns that resemble those observed in the offline human societies. We further apply UISE to online time estimation based on rendezvous as user interactions in Tribler. We demonstrate

the accuracy and scalability of UISE with different information dissemination protocols and user behaviors using simulations, emulations, and a real-world deployment. This chapter is largely based on our paper [52]:

A.L. Jia, B. Schoon, J.A. Pouwelse, and D.H.J. Epema. Estimating user interaction strength in online networks. *Under review*.

**Conclusions and future work (Chapter 7)** In this chapter, we summarize our key contributions and we provide suggestions for future work.



## Chapter 2

# Balancing reciprocity and inequality in BitTorrent

Enhancing reciprocity has been one of the primary motivations for the design of incentive policies in BitTorrent-like P2P systems. Reciprocity implies that users need to contribute their bandwidth to other users if they want to receive bandwidth in return. As we introduced earlier, BitTorrent incorporates an incentive mechanism, Tit-For-Tat (TFT), based on *direct reciprocity* to incentivize contributions, where users prefer uploading to others who have contributed to them in the past at the highest speeds. This incentive mechanism was designed to allow users to obtain their file of interest even in *resource-constrained* scenarios, e.g., when only a few users exist that hold a complete copy of the file (*seeders*, in BitTorrent terminology), or during flash-crowds.

However, the *BitTorrent ecosystem* is nowadays extremely diverse. For example, a recent measurement study [69] has shown that most BitTorrent communities are over-provisioned, i.e., there are significantly more seeders than downloaders. Also, the design of many next-generation P2P systems, such as those for the distribution of live and on-demand streaming [53, 87, 110], has been inspired by the BitTorrent paradigm. The real-time constraints of these systems require that all peers are provided with a certain minimum download speed (in order to support the bitrate of the video) and that peers do not earn more utility in downloading at rates much faster than that. These observations suggest that it is not necessary to always enhance reciprocity; in some cases it is more advisable to reduce *inequality* among peers, instead. One of the first studies of this trade-off in BitTorrent-like systems was provided by Fan *et al.* [30].

In this chapter, we propose a theoretical model for BitTorrent and we analyze *how* the incentive mechanism of the BitTorrent protocol can be tuned to enhance reciprocity or reduce inequality. Furthermore, in our study we consider the implications of exchanging BitTorrent's standard incentive mechanism with one that is based on effort rather than speed. Finally, we also analyze the role of the seeders. Hence, we provide significant

insights in the implications of the trade-off between enhancing reciprocity and reducing inequality. Our contributions can be summarized as follows:

1. We provide an analytical model that characterizes the inherent relationship between a peer's performance and the design parameters of the BitTorrent protocol that are responsible for its incentive mechanism (Section 2.2).
2. We use this model to analyze different strategies to enhance reciprocity or to reduce inequality, and to understand the role of the seeders (Section 2.3).
3. We evaluate the impact of these strategies on the overall system performance (Section 2.3).

Overall, our work in this chapter aids in informing the design choices that best fit the requirements of a BitTorrent-like P2P system.

## 2.1 BitTorrent overview

In this section we provide an overview of the BitTorrent protocol with specific focus on its Tit-For-Tat incentive policy.

### 2.1.1 The swarm

In BitTorrent, a *swarm* is consisted by peers<sup>1</sup> who are interested in the same file and sharing it with each other. Peers who partially hold the entire file are called *leechers*: they upload while downloading from each other. Peers who hold the entire file and only stay to upload are called *seeders*. In a BitTorrent swarm, peers are usually with various upload and download bandwidths, i.e., the so-called *bandwidth-heterogeneous swarm*

### 2.1.2 The Tit-For-Tat incentive policy

Incentive policies play a key role in BitTorrent-like systems, as they determine how peers distribute their limited upload bandwidth to other peers. BitTorrent's original incentive policy is *Tit-For-Tat* (TFT), in which a peer favors other peers that have recently reciprocated at the highest rate. More specifically, every peer has a number of upload slots available, which are divided into two categories, *regular unchoke slots* and *optimistic unchoke slots*. Leechers choose which peers will be allocated to regular unchoke slots according to TFT. On the contrary, peers to be allocated to optimistic unchoke slots are chosen randomly from the neighbors set. While regular unchoke slots are used to enhance

---

<sup>1</sup>From here, we use *users* and *peers* alternatively to refer to the individuals that participate in BitTorrent.



reciprocity, optimistic unchoke slots serve the purpose of (1) potentially discovering new faster peers and (2) allowing new peers to bootstrap (i.e., obtain their first pieces of the file).

Due to the special role of seeders, i.e., they have a complete copy of the file and share it without any direct benefit to do so, their uploading (or seeding) policies are different from that of the leechers. In general, there are two popular seeding policies: (1) *favoring fast peers* (FF): seeders allocate their regular upload slots to peers that downloaded at the fastest rates and optimistic unchoke slots randomly; (2) *random seeding* (RS): seeders have no preference and just choose peers randomly.

## 2.2 A Fluid Model for BitTorrent

In this section we introduce our model for a BitTorrent swarm in which peers have heterogeneous bandwidths. We present a system of differential equations that describe the evolution of a BitTorrent swarm, based on which we analyze the performance of TFT incentive policy. We illustrate the validation of our model by means of a discrete-event simulator.

### 2.2.1 The basic idea

In our approach we group peers into  $N$  different classes according to their upload capacities, with the peers in the same class having (roughly) the same upload capacities. We follow a similar fluid modeling approach as in [72, 85], where a Markov model is used to describe the arrival and departure of peers. The average download time is then derived based on a continuous fluid approximation of the Markov model under the assumption that there exists a steady state. By considering a bandwidth heterogeneous swarm, we analyze the dynamics of bandwidth allocation (1) within and between classes and (2) among regular and optimistic unchoke slots, according to the BitTorrent TFT policy.

### 2.2.2 Model description

The notation we use is shown in Table 2.1. In our model, a peer in class  $i$  has the upload capacity of  $\mu_i$ , the download capacity of  $d_i$ , and the number of unchoke slots of  $u_i$ , with  $u_i^{(reg)}$  and  $u_i^{(op)}$  for regular and optimistic unchoke slots respectively. For each class  $i$ , the number of downloads completed within a unit of time (e.g., second) is determined by the total upload bandwidth that class  $i$  receives from all classes in the swarm. Thus, the evolution of the number of leechers,  $x_i(t)$ , and the number of seeders,  $y_i(t)$ , can be described as follows:

Notation	Definition
$F$	the size of the file shared in the swarm.
$x_i$	number of leechers in class $i$ .
$\pi_i$	fraction of leechers in class $i$ , $\pi_i = x_i / \sum_i x_i$ .
$y_i$	number of seeders in class $i$ .
$\lambda_i$	the arrival rate of leechers in class $i$ .
$\gamma_i$	the rate at which seeders in class $i$ leave the system.
$\mu_i$	the upload capacity of a peer in class $i$ .
$d_i$	the download capacity of a peer in class $i$ .
$d'_i$	the per connection download capacity of a peer in class $i$ .
$u_i$	number of unchoke slots opened by a peer in class $i$ , $u_i^{(reg)}$ and $u_i^{(op)}$ for regular and optimistic unchoke slot.
$n_i$	the number of download slots opened by a class $i$ leecher
$\alpha_{ij}$	the number of upload slots allocated by a leecher in class $i$ to a leecher in class $j$ .
$\beta_{ij}$	the number of upload slots allocated by a seeder in class $i$ to a leecher in class $j$ .
$\omega_{ij}$	the fraction of upload capacity of leechers in class $i$ allocated to leechers in class $j$ .
$\sigma_{ij}$	the fraction of upload capacity of seeders in class $i$ allocated to leechers in class $j$ .
$U_{ij}$	the total upload capacity allocated from class $i$ to class $j$ .
$T_i$	the average download time of peers in class $i$ .
$D_i$	the average download speed of peers in class $i$ .

Table 2.1: Notation of our BitTorrent model

$$\begin{aligned}\frac{dx_i(t)}{dt} &= \lambda_i - \frac{\sum_j U_{ji}(t)}{F}, \\ \frac{dy_i(t)}{dt} &= \frac{\sum_j U_{ji}(t)}{F} - \gamma_i y_i(t).\end{aligned}\tag{2.1}$$

We assume that there exists a steady state in the system, i.e., while peers are arriving and departing, the total system population stays constant. In such a steady state, it holds that:

$$\frac{dx_i(t)}{dt} = \frac{dy_i(t)}{dt} \equiv 0,$$

together with Eq.(2.1) we have:

$$\lambda_i F = \sum_j U_{ji} = \gamma_i y_i F.\tag{2.2}$$

Here, we denote the equilibrium values of the quantities in steady state by  $U_{ji}$  for  $U_{ji}(t)$ , etc. As the arrival rate of leechers in a steady state is equal to the departure rate, i.e., the rate they finish their download, let  $T_i$  represent the average download time for peers in class  $i$ , we can apply the Little's Law to the number of leechers, i.e.:

$$x_i = \lambda_i T_i.\tag{2.3}$$

The total upload bandwidth of class  $j$  allocated to class  $i$  consists of upload bandwidth allocated by seeders and by leechers. Therefore, we have:

$$\sum_j U_{ji} = \sum_j (\omega_{ji} x_j + \sigma_{ji} y_j) \mu_j.\tag{2.4}$$

Combining Eqs. 2.2, 2.3 and 2.4, the average download speed for leechers in class  $i$  can be calculated as:

$$D_i = \frac{F}{T_i} = \frac{F \lambda_i}{x_i} = \frac{1}{x_i} \sum_j (\omega_{ji} x_j + \sigma_{ji} y_j) \mu_j.\tag{2.5}$$

We discuss how to derive the upload bandwidth allocation ( $\omega_{ji}$  and  $\sigma_{ji}$  respectively) in the following subsection.

### 2.2.3 Bandwidth allocation

Without loss of generality, we assume that  $\mu_1 < \mu_2 < \dots < \mu_N$ .

In BitTorrent, a leecher opens a number of regular and optimistic unchoke slots to upload to other peers. In our model, we assume a leecher can always find enough other leechers to be allocated into his unchoke slots. According to TFT, regular unchoke slots are allocated to peers with the best reciprocity and optimistic unchoke slots are allocated randomly. Therefore, high capacity peers (e.g., peers in class  $i$ ) only unchoke low capacity peers (e.g., peers in class  $j$ ,  $i > j$ ) in optimistic unchoke slots. Consequently, the number of upload slots allocated from a peer in class  $i$  to a peer in class  $j$  is equal to:

$$\alpha_{ij} = u_i^{(op)} \pi_j \quad i, j = 1, 2, \dots, N, i > j.$$

Due to their faster upload speed, higher-capacity leechers will get reciprocated when they upload to lower-capacity leechers. As long as there are enough upload slots, on average, the number of leechers in class  $i$  that a leecher in class  $j$  should reciprocate equals to:

$$\frac{\alpha_{ij} x_i}{x_j} = \frac{u_i^{(op)} \pi_j x_i}{x_j} = \frac{u_i^{(op)} x_i}{\sum_j x_j} = u_i^{(op)} \pi_i \quad i, j = 1, 2, \dots, N, i > j.$$

In case there are not enough upload slots, leechers in higher classes are reciprocated first, i.e.:

$$\alpha_{ji} = \min\{u_i^{(op)} \pi_i, u_j^{(reg)} - \sum_{i < p \leq N} \alpha_{jp}\} + u_j^{(op)} \pi_i \quad i, j = 1, 2, \dots, N, i > j..$$

Seeders adopt different upload strategies than leechers since they already have the entire file and do not need to be reciprocated. For seeders who adopt the FF (favoring fast) policy, its regular unchoke slots are allocated to the fastest leechers, i.e., peers in class  $N$ . And its optimistic unchoke slots are shared by leechers in all classes. Therefore, we have:

$$\begin{aligned} \beta_{iN} &= u_i^{(reg)} + u_i^{(op)} \pi_N, \\ \beta_{ij} &= u_i^{(op)} \pi_j \quad \forall i, j \text{ and } j < N, \end{aligned}$$

For seeders who adopt the RS (random seeding) policy, all unchoke slots are shared by leechers in all classes, i.e.:

$$\beta_{ij} = u_i \pi_j.$$

As peers are unchoked based on their upload capacities, i.e., their classes, peers in the same class receive similar services. Therefore, on average, the number of upload slots a peer in class  $i$  receives, i.e., the number of download slots it opens, is equal to:

$$n_i = \frac{\sum_j \alpha_{ji} x_j + \beta_{ji} y_j}{x_i}.$$

Consequently, the per connection download capacity of a peer in class  $i$  equals to:

$$d'_i = \frac{d_i}{n_i}.$$

BitTorrent uses TCP as transport layer protocol. TCP specifies that a peer's upload (download) capacity is equally divided over all connections, unless some of the connections have a bottleneck. When such a bottleneck exists, the leftover bandwidth is equally divided over other connections with higher link capacities. Following this basic rule, we calculate the fraction of the upload capacity of leechers in class  $i$  allocated to leechers in class  $j$ , i.e.,  $\omega_{ij}$ , as follows.

First, we reorder the leechers according to their per connection download bottleneck  $d'_i$  in such a way that  $d'_1 < d'_2 < \dots < d'_N$ . Then, we assume that the fraction of the upload capacity of leechers in class  $i$  allocated to leechers in class  $p$ , i.e.,  $\omega_{ip}$  where  $p < j$ , is known. Therefore, the left upload bandwidth, i.e.,  $\mu_i(1 - \sum_{p < j} \omega_{ip})$ , should be equally allocated among  $\sum_{k \geq j} \alpha_{ik}$  leechers in class  $k$  where  $k \geq j$ , as long as their per connection download capacity is not saturated. Finally, the fraction of the upload capacity of leechers in class  $i$  allocated to leechers in class  $j$  can be calculated as:

$$\omega_{ij} = \frac{\min\left\{\frac{\mu_i(1 - \sum_{p < j} \omega_{ip})}{\sum_{k \geq j} \alpha_{ik}}, d'_j\right\} \cdot \alpha_{ij}}{\mu_i}. \quad (2.6)$$

By calculating  $\omega_{ij}$  in sequence, e.g.,  $\omega_{i1}$ ,  $\omega_{i2}$ ,  $\omega_{i3}$  and so on, we can derive the upload capacity allocation between any two classes. Replacing  $\omega_{ij}$ ,  $\alpha_{ij}$  with  $\sigma_{ij}$ ,  $\beta_{ij}$  respectively, a seeder's upload bandwidth allocation can be derived in a similar way.

## 2.2.4 Model Validation

We have validated our model by means of simulations using a discrete-event simulator that simulates the behavior of BitTorrent at the level of piece transfers [72]. Fig. 2.1 illustrates the simulation results against the model predictions for a system with two classes of peers, fast and slow, from which we can make the following observations:

- the model predictions are close to the simulation results;
- the average download speed of both fast and slow peers increases when there are more seeders;
- the model predictions become less accurate as the fraction of seeders grows. This can be explained considering that, when a high fraction of peers are seeders (above

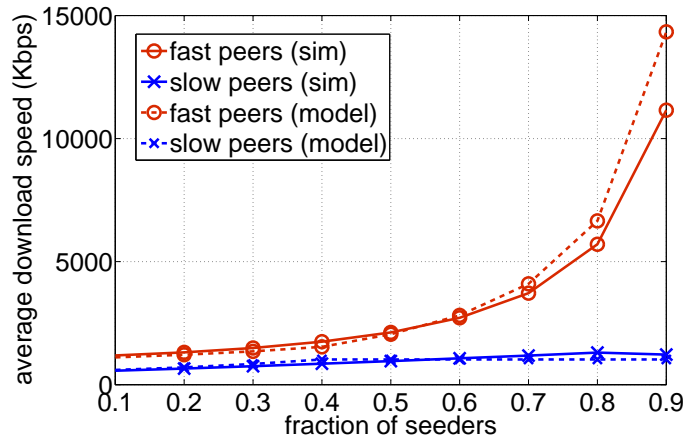


Figure 2.1: The average download speeds of fast and slow peers in a system with 50 fast peers and 50 slow peers, for different fraction of seeders. Fast peers are with infinite download capacity and upload capacity of 1024 Kbps; and slow peers are with upload and download capacities of 512 Kbps and 1024 Kbps, respectively. Seeders use the FF policy.

70 % in this case), fast leechers have a hard time in finding other fast leechers to reciprocate with. While in our model we assume that, in a steady state, leechers can always find enough other leechers.

## 2.3 Analysis

In this section, we analyze the balance between enhancing reciprocity and reducing inequality in BitTorrent. Specifically, we introduce four strategies: (1) fast peers opening more regular unchoke slots, (2) all peers opening more optimistic unchoke slots, (3) replacing TFT with an effort-based incentive policy, and (4) seeders favoring fast peers versus seeding randomly.

We evaluate the performance of the above strategies based on the following three performance metrics:

- *Download speed* is used to characterize performance.
- *Sharing ratio* is defined as the ratio between the total amount of data uploaded and downloaded. It represents a fairness in relation to contribution to the system—a sharing ratio equal to 1 for all peers means that all peers have contributed as much data as they have consumed.
- *Inequality coefficient* is defined as the largest download speed divided by the smallest download speed found among all peers. It indicates a fairness in relation to the

bandwidth capacity that peers receive from the system.

Unless stated otherwise, we consider a system with two classes of peers, i.e., fast peers with infinite download capacity and upload capacity of 1024 Kbps; and slow peers with upload and download capacities of 512 Kbps and 1024 Kbps, respectively. And the default values for the number of optimistic and regular unchoke slots are set to 1 and 4, respectively.

### 2.3.1 Enhancing reciprocity

A straightforward way to enhance reciprocity is to open more regular unchoke slots. Regardless of a peer's class, opening more upload slots can help a peer to (1) find more potential fast peers, or to (2) weaken another peer's potential monopoly on its uploading bandwidth since less bandwidth will be allocated to each upload slot. On the other hand, opening too many slots is neither realistic nor reasonable, since too many TCP connections could deteriorate link performance. Also it would become harder for slow peers to succeed in competing for reciprocity with faster peers.

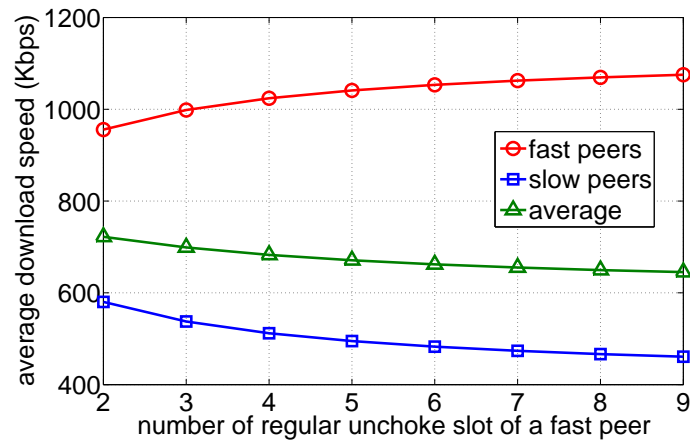
Given the above considerations, fast peers have a stronger motivation to open more slots than slow peers, since they may benefit from more extensive exploration, while remaining competitive in TFT. Having fast peers open more regular upload slots is a way to enhance reciprocity, as more bandwidth will be allocated to the regular unchoke slots.

In this experiment, we vary the number of regular unchoke slots of fast peers. Fig. 2.2(a) shows that as the number of regular unchoke slots of fast peers increases, their download speed improves (we can observe a growth of 10% when the number of regular unchoke slots goes from 2 to 9), while the average download speed of all peers decreases (10% with the number of regular unchoke slots from 2 to 9). This is due to the increasing inequality (almost 50%) between the two classes of peers, as shown in Fig. 2.2(c). On the other hand, we notice that the sharing ratio of fast peers decreases as they open more regular unchoke slots, and that of slow peers increases (Fig. 2.2(b)). The *perfect reciprocity*, i.e., sharing ratio equal to 1 for both fast and slow peers, is achieved when fast peers open 4 regular unchoke slots.

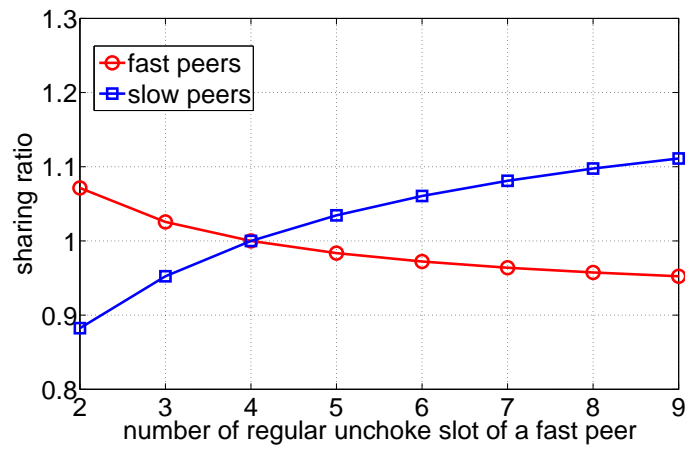
In the following theorem we state the conditions necessary to achieve the perfect reciprocity.

**Theorem 2.1** *In a BitTorrent system with two classes of peers, no seeders, and no download bottleneck, where peers can fully utilize their upload capacities, perfect reciprocity is achieved if and only if:*

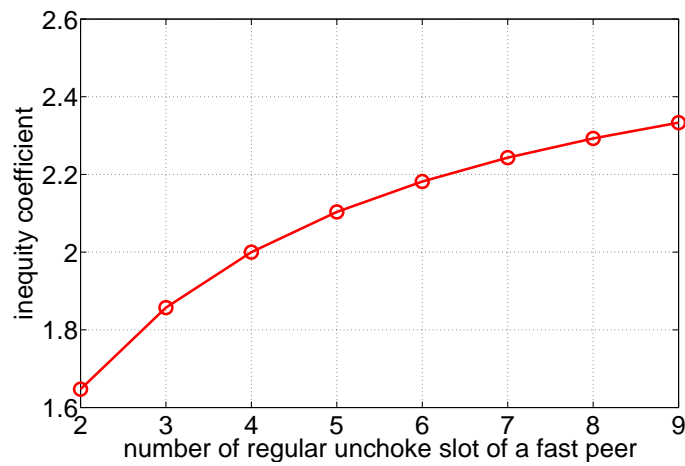
$$\frac{\mu_f u_s}{\mu_s u_f} = \frac{u_f^{(op)} + u_s^{(op)}}{u_f^{(op)}}. \quad (2.7)$$



(a) Average download speed



(b) Sharing ratio



(c) Inequality coefficient

Figure 2.2: The influence of the number of regular unchoke slot of fast leechers in a system with 100 leechers and no seeders.



**Proof:** We first show that for a system with perfect reciprocity, Eq.(2.7) holds. The sharing ratio of a leecher in class  $i$  in a steady state is equal to the ratio of its upload and download speed, i.e.:

$$\frac{\mu_i}{D_i} = \frac{\mu_i x_i}{F \sum_{j \in \{f,s\}} \omega_{ji} x_j \mu_j}. \quad (2.8)$$

where the download speed  $D_i$  is calculated from Eq.(2.5).

Perfect reciprocity implies that leechers in different classes achieve the same sharing ratio, i.e.:

$$\frac{\mu_f x_f}{\sum_{j \in \{f,s\}} \omega_{jf} x_j \mu_j} = \frac{\mu_s x_s}{\sum_{j \in \{f,s\}} \omega_{js} x_j \mu_j}. \quad (2.9)$$

Following the model proposed in Section 2.2, fast peers only upload to slow peers in optimistic unchoke slots and slow peers reciprocate those fast peers in their regular unchoke slots. Therefore:

$$\begin{aligned} \omega_{fs} &= \frac{u_f^{(op)} \pi_s}{u_f} = 1 - \omega_{ff}, \\ \omega_{sf} &= \frac{u_f^{(op)} \pi_s x_f}{x_s u_s} = 1 - \omega_{ss}. \end{aligned} \quad (2.10)$$

Taking Eq.(2.10) back to Eq.(2.9), it follows that Eq.(2.7) holds.

Next, we show that when Eq.(2.7) holds, perfect reciprocity is achieved. Substituting Eq.(2.7) into Eq.(2.8), we get Eq.(2.9), which implies that fast and slow leechers have the same sharing ratio. It follows that perfect reciprocity is achieved.

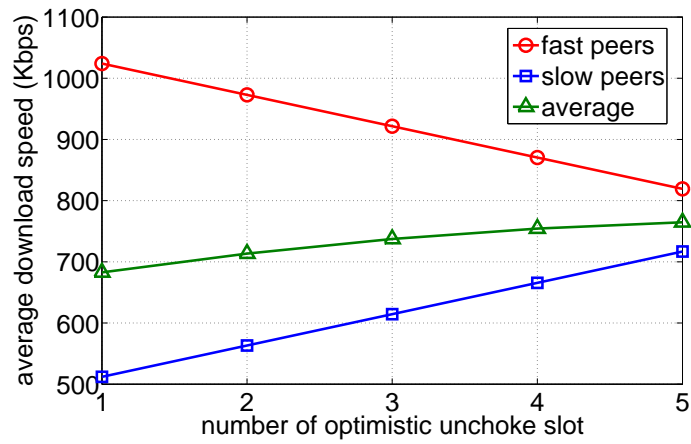
From the above theorem it follows that, when we use  $\mu_f = 1024, \mu_s = 512$  and  $u_s^{(op)} = u_f^{(op)} = 1$ , a perfect reciprocity is obtained for  $u_f^{(reg)} = u_s^{(reg)} = 4$ .

### 2.3.2 Reducing inequality

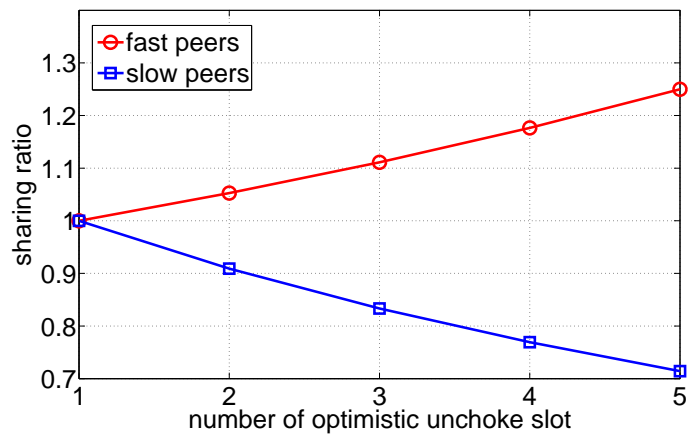
In this section we evaluate two strategies for reducing inequality, namely TFT with more optimistic unchoke slots and effort-based policy.

#### TFT with more optimistic unchoke slots

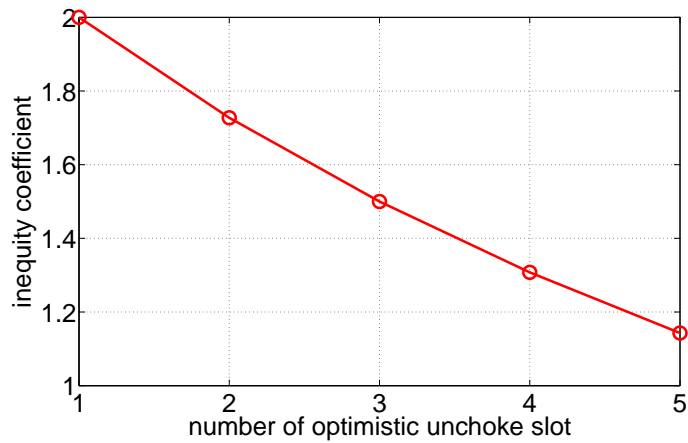
A straightforward way to reduce inequality is to open more optimistic unchoke slots. In this section, we analyze the influence of having all peers open more optimistic unchoke



(a) Average download speed



(b) Sharing ratio



(c) Inequality coefficient

Figure 2.3: The influence of the number of optimistic upload slot of peers in a system with 100 leechers and no seeders.

slots. While peers always open 5 unchoke slots in total, we let the number of their optimistic unchoke slots vary from 1 to 5. As we can see in Fig. 2.3(a), in this way the download speed of slow leechers is improved by 40%, at the expense of the fast leechers. Interestingly, the average download speed of the whole population increases of 15%. Moreover, we observe a 45% decrease of the inequality coefficient (Fig. 2.3(c)).

However, it should be noted that by having peers open more optimistic unchoke slots, the effectiveness of TFT is reduced, as a peer with no contributions is chosen with the same probability as a cooperative peer.

### Effort-based incentives

Rahman *et al.* [90] have recently proposed a novel incentive mechanism based on effort, rather than speed. More specifically, under this effort-based mechanism, peers are not rewarded based on the absolute amount of data they provided, but based on the relative amount of bandwidth they make available (utilized or not). With this approach, a slow peer offering all its bandwidth to the system is preferred over a fast peer offering 0.9 of its total bandwidth.<sup>2</sup>

Consider that there are two types of peers in the system, i.e., *fully cooperative* peers that contribute all their upload bandwidth and *partially cooperative* peers that only contribute a fraction of their upload bandwidth. Let  $n_p$  represent the number of partially cooperative peers, and  $n_{ff}$  and  $n_{fs}$  represent the number of fully cooperative peers that have a high and low upload capacity, respectively. Based on the effort-based mechanism, each peer reciprocates fully cooperative peers by allocating regular unchoke slots to them, and punishes partially cooperative peers by only optimistically unchoking them. Then, the slot allocation for each class of peers can be calculated as:

$$\begin{aligned}\alpha_{i(p)} &= \frac{u_i^{(op)} n_p}{n_{ff} + n_{fs} + n_p} \\ \alpha_{i(ff)} &= \frac{(u_i - \alpha_{i(p)}) n_{ff}}{n_{ff} + n_{fs}} \\ \alpha_{i(fs)} &= \frac{(u_i - \alpha_{i(p)}) n_{fs}}{n_{ff} + n_{fs}} \quad \forall i \in \{p, ff, fs\}.\end{aligned}\tag{2.11}$$

Given Eq.(2.11), the upload bandwidth allocation can be calculated in a similar way as in our earlier analysis.

The idea of this effort-based incentive scheme is to reduce inequality among the fully cooperative peers while still punishing the partially cooperative peers. We first evaluate its performance by considering a system with both fully and partially cooperative peers,

<sup>2</sup> [97] provides a possible method to measure upload capacity of users in P2P systems.

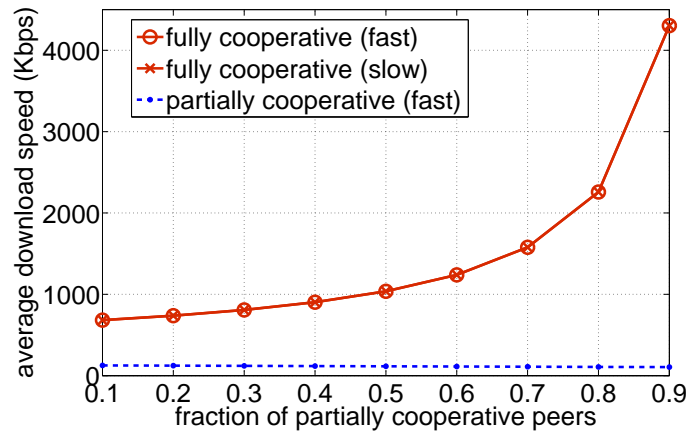
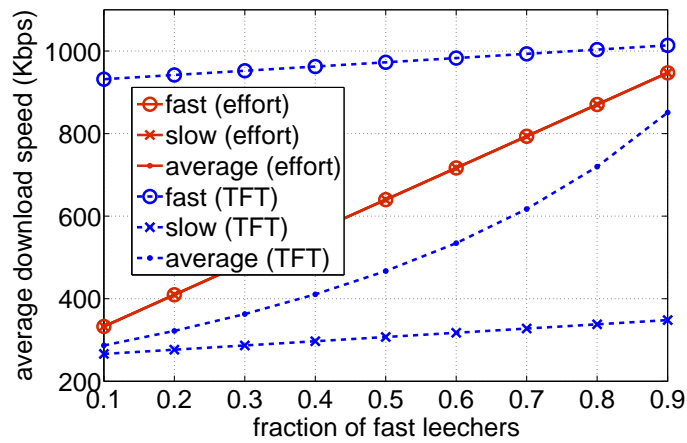
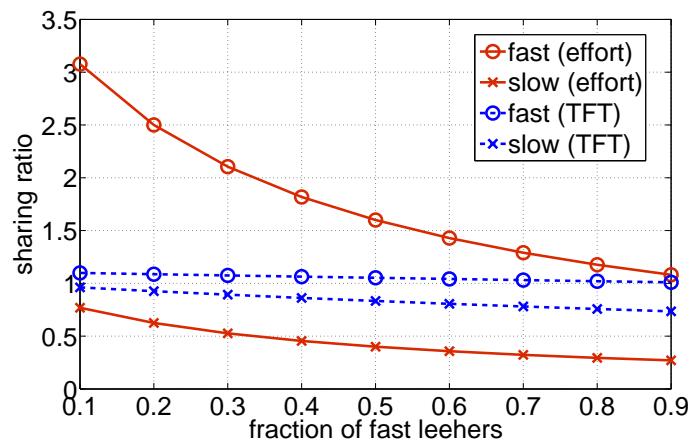


Figure 2.4: The effectiveness of effort-based mechanism.



(a) Average download speed



(b) Sharing ratio

Figure 2.5: The comparison between effort-based mechanism and TFT. The upload capacities of fast and slow peers are 1024 and 256 Kbps, respectively.

---

where we vary the fraction of partially cooperative peers from 0.1 to 0.9. Particularly, we set partially cooperative peers to be fast peers, and we set half of the fully cooperative peers to be fast and the other half to be slow. Results are shown in Fig. 2.3.2. We see that this effort-based policy effectively eliminates the inequality in the system and the partially cooperative peers are punished by achieving lower download speeds than fully cooperative peers.

Next, we compare the effort-based policy with TFT. We consider a system in which all peers are fully cooperative and we vary the fraction of fast peers among them. As shown in Fig. 2.5(a), the effort-based scheme eliminates the system's inequality and achieves a better overall performance, in terms of a higher download speed averaged over all peers. As a consequence, it reduces the reciprocity and leads to a higher disparity in sharing ratios achieved by fast and slow peers (see Fig. 2.5(b)).

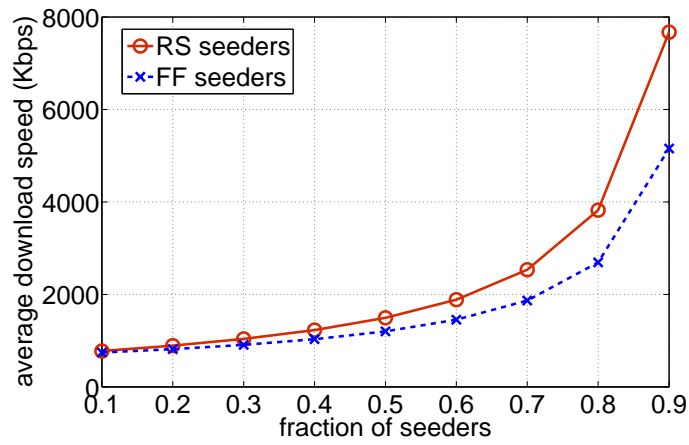
### 2.3.3 Seeding policies

The mainline BitTorrent client has been implemented with two different seeding strategies in different releases. One is the favoring of fast peers. This strategy accelerates a fast leecher's ability to finish downloading, thereby potentially having it serve as fast seeder in the system sooner. The other strategy is seeding randomly. The first strategy is resource-constrained oriented, as it aims at increasing the serving capacity quickly. The second strategy is more equality oriented, as all peers are treated in the same way.

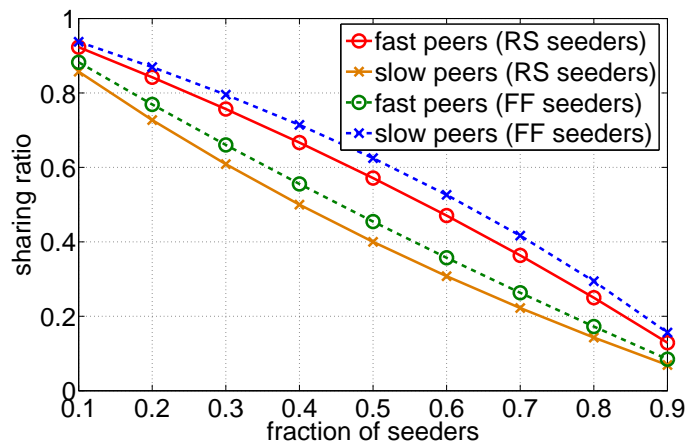
We have applied our model to analyze and compare these two strategies, where we use the default parameter settings as specified at the beginning of Section 2.3. Fig. 2.6(a) and Fig. 2.6(c) show that if seeders seed randomly, the system achieves a better overall performance (in terms of a higher average download speed) and the inequality is reduced. On the contrary, if seeders favor fast peers, the reciprocity is enhanced. As shown in Fig. 2.6(b), both fast and slow peers achieve sharing ratios higher than in a system where seeders adopt random seeding.

## 2.4 Related work

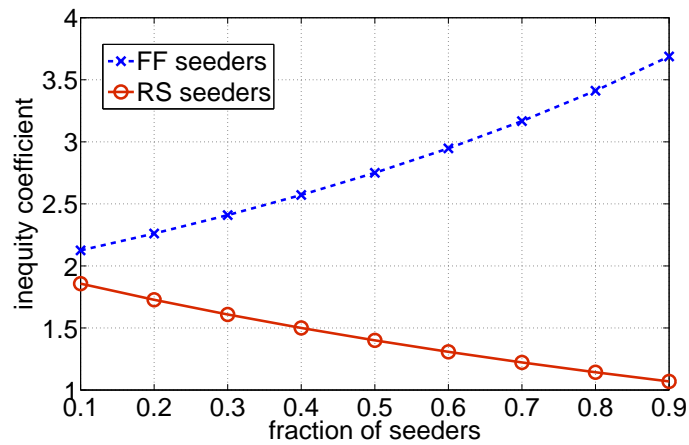
There are a number of studies on modeling and improving BitTorrent's incentive policies. Some earlier work focuses only on homogeneous systems [36, 85, 117]. Liao *et al.* [112] have considered heterogeneous BitTorrent systems but only with two classes of peers. Fan *et al.* [30] have developed a general heterogeneous model to evaluate the tradeoff between performance and fairness. Meulpolder *et al.* [72] and Chow *et al.* [16] also provide models for heterogeneous BitTorrent systems, with which they analyze the clustering and data distribution in BitTorrent swarms. While these works all focus on a particular design, we analyze the performance of different incentive policies from a higher level: we consider



(a) Average download speed



(b) Sharing ratio



(c) Inequality coefficient

Figure 2.6: The influence of seeding strategies: favoring fast peers (FF) and randomly seeding (RS).

different BitTorrent applications and stress that merely enhancing reciprocity is not sufficient in the design of a good incentive policy. We furthermore identify several strategies that can be used to enhance reciprocity or reduce inequality.

## **2.5 Conclusions**

Inspired by the observations of nowadays BitTorrent applications such as over-provisioned BitTorrent communities and P2P streaming with lowest rate constrained, in this chapter, we have analyzed why an incentive policy should not only enhance reciprocity but also reduce inequality. We have provided an analytical model for heterogeneous BitTorrent systems that captures the essence of BitTorrent's incentive policy. Based on our model, we have analyzed how TFT could enhance reciprocity or reduce inequality by carefully tuning the number of regular and optimistic unchoke slots. We have also compared TFT to an effort-based incentive policy, and have showed that a policy that focuses on reducing inequality leads to a better overall performance in terms of a higher upload speed averaged over all peers. Finally, we have analyzed different seeding policies and our results show that, although seeders do not need to be reciprocated, they can still be used to further enhance reciprocity or reduce inequality among leechers.





## Chapter 3

# Modeling and analysis of Sharing Ratio Enforcement

Sharing Ratio Enforcement (SRE) was first adopted in private BitTorrent communities, aiming at providing contribution incentives beyond BitTorrent's original Tit-For-Tat. It is a general contribution enforcement policy that can be easily applied to many online networks besides BitTorrent-like file sharing systems, by simply generalizing the metrics for determining the sharing ratio from the upload and download amounts to any metrics representing contribution and consumption. In this chapter, we propose a theoretical model, based on which we demonstrate the effectiveness of SRE.

BitTorrent's TFT incentive policy works reasonably well in fostering cooperation among leechers. However, similar to a bilateral barter in human society, TFT requires the presence of the coincidence of wants of the participants, and therefore it does not provide any incentive for users to remain in the system after the download is complete, in order to *seed* the entire file to others. Furthermore, it has been shown that TFT is vulnerable to attacks such as the *large view exploit* [66], by means of which a user succeeds in achieving a good download speed without uploading any data in return.

To overcome the above issues, in recent years, there has been a large proliferation of so-called *private* BitTorrent communities. These sites typically require users to register accounts and then demand that their members maintain a *sharing ratio*, i.e., the ratio between a peer's total upload and download amounts, above a particular threshold. This mechanism is known under the name of *Sharing Ratio Enforcement (SRE)*<sup>1</sup>. Community members whose sharing ratio drops below the threshold are warned and then banned from downloading, or even expelled from the community. In this way, it is guaranteed that each participant provides a certain level of contribution to the community. Furthermore, since it is normally difficult to obtain membership of a private BitTorrent community, the threat

---

<sup>1</sup>The sharing ratio is calculated and recorded by the trackers deployed by each community.

of the *white-washing attack*<sup>2</sup> is very low. On the other hand, to make SRE realistic and feasible, most private communities adopt some special rules. For example, new members normally are provided with a bonus to get started (e.g., in HDChina [64] the first download is for free).

Several measurement studies show that SRE is very effective in boosting cooperation [13, 64, 69, 124]. For instance, [69] reports seeder-to-leecher ratios that are at least 9 times higher than in public BT communities, while download speeds are measured to be 3–5 higher. Hence it would be beneficial to analyze *how* SRE actually provides seeding incentives and quantify the expected performance improvement, in terms of user download speeds. In this chapter we focus on these aspects. Specifically, our contributions are as follows:

1. We provide an analytical model for bandwidth-heterogeneous private communities, which characterizes the inherent relationship between a peer's performance and the parameters of SRE. We apply our model both to a single swarm and across multiple swarms, and we quantify the performance improvement/deterioration for peers with different capacities, assuming rational user behavior (Section 3.1).
2. We analyze the factors that build up SRE's influence, i.e., the SRE threshold, and the bandwidth heterogeneity of the peers in the system (Section 3.2).
3. We show that, due to the influence of irrational user behavior observed in real private communities, i.e., some peers seed more than they need and achieve sharing ratios (much) higher than the threshold [64], the expected download speeds derived in our model represent a *lower bound* for the actual download speeds achievable by peers. Hence, following our model, administrators of private communities can predict the minimum performance level their systems will be able to reach. (Section 3.2).

## 3.1 A simple model for Sharing Ratio Enforcement

In this section, we introduce the assumptions and other details of our model.

### 3.1.1 Preliminary: rational user behavior

For the purpose of our analysis, we consider a user to be rational if it tries to maximize its download speed and minimize its seeding work. This means that when SRE is adopted,

---

<sup>2</sup>White-washing refers to the action of a peer who repeatedly rejoins a system with a new identity in order to get rid of a negative history.

rational peers seed the minimum amount to meet the threshold and, when SRE is not adopted, peers immediately leave the system once their downloads are complete.

Furthermore, we assume that rational peers always upload at their full capacity, since a recent study [64] has shown that when using TFT, uploading at the full capacity is the best strategy for peers to maximize their download speed. Finally, in accordance with previous work [85], we do not consider the piece availability problem and assume that peers can always find pieces that they are interested in at other peers.

### 3.1.2 The basic model

We consider a BitTorrent swarm in which peers are downloading and uploading pieces of the same file. Similar to the model proposed in Chapter 2, we assume that the system is in steady state: while peers are arriving and departing, the total size of the population is constant. We group peers into different classes according to their upload capacities, and assume that the upload bandwidth allocated to each class is equally shared by all the leechers in that class. The notation used in our model is illustrated in Table 3.1.

Given the above assumptions, we can derive that within a time interval  $\mathcal{T}$ , there will be  $\mathcal{T}x_i/t_i$  leechers in class  $i$  who will have completed their downloads, where  $t_i$  is their average download time and  $x_i$  is the number of leechers in class  $i$  in a steady state at any given time. We assume that  $\mathcal{T}$  is long enough so that we can get an average performance for peers in each class. The conservation law applied to the bandwidths implies that, in this interval, the total download amount must be equal to the total upload amount, i.e.:

$$\sum_i \frac{\mathcal{T}}{t_i} x_i F = \sum_i \frac{\mathcal{T}}{t_i} x_i \mu_i T_i, \quad (3.1)$$

where  $T_i$  denotes the total length of time a peer has spent in the swarm (both as a leecher and as a seeder).

For a swarm where peers do not seed (i.e.,  $T_i = t_i$ ), and considering that the download speed  $d_i$  of a peer in class  $i$  equals  $F/t_i$ , Eq.(3.1) becomes:

$$\sum_i x_i d_i = \sum_i x_i \mu_i. \quad (3.2)$$

### 3.1.3 Within one swarm

We first analyze how SRE influences the system performance when it is applied within one swarm. This mechanism is adopted in several private BitTorrent communities such as BitHQ [9] and PolishTracker [81]. Under this situation, once a leecher's sharing ratio

Notation	Definition
$\alpha$	the sharing ratio threshold of SRE.
$F$	the size of the file being downloaded.
$N$	the total number of classes of peers.
$x_i$	number of leechers in class $i$ in a steady state.
$\mu_i$	the upload capacity of a peer in class $i$ .
$\omega_{ij}$	the fraction of upload capacity of leechers in class $i$ allocated to leechers in class $j$ in a steady state.
$\sigma_{ij}$	the fraction of upload capacity of seeders in class $i$ allocated to leechers in class $j$ in a steady state.
$d_i$	the average download speed of a peer in class $i$ .
$t_i$	the average download time of a peer in class $i$ .
$T_i$	the time a peer in class $i$ spent in the swarm,
$\varphi_i$	the sharing ratio achieved by a peer in class $i$ as a leecher.

Table 3.1: Notation of our BitTorrent model

drops below the threshold, its download process is halted immediately: it needs to upload for a while until it gains enough sharing ratio to perform further downloading. When it completes the download, the peer leaves the swarm immediately (hence  $T_i = t_i$ ), with a sharing ratio no less than the SRE threshold  $\alpha$ .

We call  $\varphi_i^\circ$  the sharing ratio a leecher in class  $i$  would obtain in a swarm where SRE is not applied. When SRE is applied, if  $\varphi_i^\circ < \alpha$ , peers in class  $i$  will be banned for some time during their downloads, so that their final sharing ratio  $\varphi_i$  will be exactly equal to the threshold. While these peers are banned, their upload capacities will be allocated to the other peers in class  $j$ , where  $j \in \{j, \varphi_j^\circ \geq \alpha\}$ . Given the definition of sharing ratio as  $\varphi_i = \mu_i t_i / F = \mu_i / d_i$ , from the conservation law in Eq.(3.2), we have:

$$\sum_{i, \varphi_i^\circ < \alpha} x_i \frac{\mu_i}{\alpha} + \sum_{i, \varphi_i^\circ \geq \alpha} x_i \frac{\mu_i}{\varphi_i} = \sum_i x_i \mu_i. \quad (3.3)$$

As an illustrative example, let us consider a swarm that consists of a class of slow peers with upload bandwidth  $\mu_s$  and a class of fast peers with upload bandwidth  $\mu_f$ . We assume that originally slow peers cannot achieve the SRE threshold when using only TFT. Eq.(3.3) implies that:

$$x_s \frac{\mu_s}{\alpha} + x_f \frac{\mu_f}{\varphi_f} = x_s \mu_s + x_f \mu_f. \quad (3.4)$$

It is easy to verify that when  $\alpha < 1$  (which is the case in most private communities), it will always hold that  $\varphi_f > 1$ . For example, when  $x_s = x_f$ ,  $\mu_f = 4\mu_s$ , and  $\alpha = 0.9$ , we

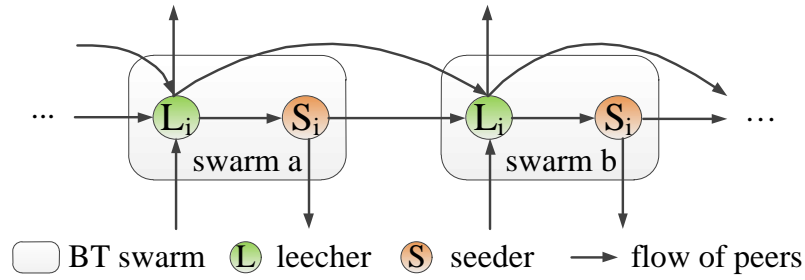


Figure 3.1: The fluid model for multiple swarms.

have  $d_s = 10\mu_s/9$ ,  $d_f = 35\mu_s/9$ , and  $\varphi_f = 36/35 > 1$ . This implies that fast peers who originally could achieve the SRE threshold will still be able to achieve it, though their download speed is increased due to the extra bandwidth allocated to them.

### 3.1.4 Within multiple swarms

To avoid getting a worse performance, it is reasonable for a peer who cannot achieve the SRE threshold while leeching, to seed for a while before starting another download. In this way, it will gain some sharing ratio as deposit, and spend it for its next download.

We assume that in a BitTorrent community containing a number of different files (each associated with a different swarm), peers that are interested in multiple files will download them one after another. We do not consider parallel downloads, since a peer who downloads  $n$  files simultaneously can be considered as being  $n$  different peers, each having  $1/n$  of the original upload capacity. For the same reason, we do not consider parallel seeding either.

The flow of peers within multiple swarms is shown in Fig. 3.1: if, after completing its download in swarm  $a$ , a leecher meets the SRE threshold (i.e.,  $\varphi_i \geq \alpha$ ), it can directly join another one (swarm  $b$  in the figure) if it wishes so. Otherwise ( $\varphi_i < \alpha$ ), to keep its community membership, the peer needs to turn into a seeder, and seed for a while in swarm  $a$ . Its seeding amount should compensate its upload deficiency, i.e., the required upload amount minus the actual upload amount:  $(\alpha - \varphi_i)F$ .

For simplicity of presentation, we assume that all swarms are of an identical configuration, i.e., the files are of the same size ( $F$ ), the compositions of peers are the same<sup>3</sup>. Within a long time period  $\mathcal{T}$ , in a particular swarm there will be  $\mathcal{T}x_i/t_i$  leechers in class  $i$  who have completed their downloads. If  $\varphi_i < \alpha$ , each of these leechers will turn into a seeder and seed  $(\alpha - \varphi_i)F$  amount of data. The conservation law implies that, in this interval, the total download amount must be equal to the total upload amount, i.e.:

<sup>3</sup>It should be noted that we can perform the same analysis for swarms of different configurations by simply adding a coefficient in our equations.

$$\sum_i \frac{T}{t_i} x_i F = \sum_i \frac{T}{t_i} x_i \mu_i t_i + \sum_{i, \varphi_i < \alpha} \frac{T}{t_i} x_i (\alpha - \varphi_i) F, \quad (3.5)$$

where the two terms on the right side account for the contributions of leechers and seeders, respectively.

Let  $\mu'_i = (\alpha - \varphi_i)F/t_i$ , then, similarly to Eq.(3.1), Eq.(3.5) can be simplified to:

$$\sum_i x_i d_i = \sum_i x_i \mu_i + \sum_{i, \varphi_i < \alpha} x_i \mu'_i. \quad (3.6)$$

To this end, given that the resource allocation is only determined by TFT, the average download speed of a peer in class  $i$  can be calculated according to the model proposed in Chapter 2, i.e.:

$$d_i = \frac{\sum_j \omega_{ji} x_j \mu_j + \sum_{j \in \{j, \varphi_j < \alpha\}} \sigma_{ji} x_j \mu'_j}{x_i}, \quad i = 1, 2, \dots, N, \quad (3.7)$$

where  $\omega_{ji}$  ( $\sigma_{ji}$ ) specifies the fraction of bandwidth allocated from a leecher (seeder) in class  $j$  to leechers in class  $i$  in the BitTorrent protocol. The accuracy of this model has been demonstrated through simulations [48].

Note that the term on the right side of Eq.(3.6) is equal to  $\sum_{i, \varphi_i < \alpha} x_i (\mu_i + \mu'_i) + \sum_{i, \varphi_i \geq \alpha} x_i \mu_i$ , which implies that the upload performed by a peer in class  $i$  (as a leecher and later as a seeder), where  $i \in \{i, \varphi_i < \alpha\}$ , is equivalent to it uploading (only as a leecher) at a speed equal to  $\mu_i + \mu'_i$ . In both cases it exactly achieves the SRE threshold. Hence,  $(\mu_i + \mu'_i)/d_i = \alpha$ , from which we can calculate  $\mu'_i$  as:

$$\mu'_i = \alpha d_i - \mu_i, \quad i \in \{i, \varphi_i < \alpha\}, \quad (3.8)$$

and solve the system of equations in Eq.(3.7).

As an illustrative example, let us consider again a system with two classes. Assuming that the sharing ratio obtained by slow peers when using only TFT is below the SRE threshold, from Eqs. 3.7 and 3.8 it follows that:

$$\begin{cases} d_s = \omega_{ss} \mu_s + \omega_{fs} \mu_f \frac{x_f}{x_s} + \sigma_{ss} \mu'_s \\ d_f = \omega_{sf} \mu_s \frac{x_s}{x_f} + \omega_{ff} \mu_f + \sigma_{sf} \mu'_s \frac{x_s}{x_f} \\ \mu'_s = \alpha d_s - \mu_s \end{cases}$$

Here  $\omega_{ij}$  and  $\sigma_{ij}$  are determined by the design of TFT and network settings. We use the default TFT settings as in the BitTorrent main client, i.e., peers open 5 upload slots, among which one is used for optimistic unchoke, and seeders seed randomly [72]. Then, for the following network settings:  $x_s = x_f$ ,  $\mu_f = 4\mu_s$ ,  $\alpha = 0.9$ , we have:  $\omega_{fs} = 1 - \omega_{ff} = 0.1$ ,  $\omega_{sf} = 1 - \omega_{ss} = 0.2$ ,  $\sigma_{ss} = 1 - \sigma_{sf} = 0.5$ . Solving the above

system equations, we have:  $d_s = 14\mu_s/11$ ,  $d_f = 213\mu_s/55$ , and  $\varphi_f = 220/213 > 1$ . We see that, despite the extra seed supply provided by slow peers, fast peers can still meet the SRE threshold.

## 3.2 Analysis

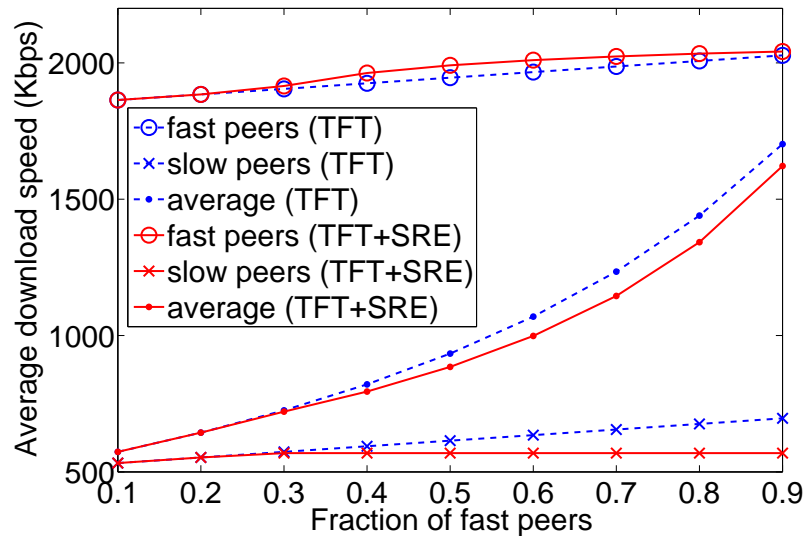
Based on our model, in this section we analyze the performance of SRE both when it is applied to a single swarm and to multiple swarms. Unless stated otherwise, we assume that each swarm in the steady state consists of two classes of peers, 50 fast peers with an upload capacity  $\mu_f$  equal to 2048 Kbps and 50 slow peers with an upload capacity  $\mu_s$  equal to 512 Kbps. We believe this simplified version of upload capacity setting is already enough for our analysis, though more complicated capacity distribution can also be used. By default, there is the same number of fast and slow peers, and the SRE threshold  $\alpha$  is set to 0.9.

We use two metrics to evaluate SRE's performance, i.e., the average download speed and the sharing ratio, and we consider three factors influencing SRE's performance, i.e., the SRE threshold  $\alpha$ , the fraction of fast peers, and the upload capacity ratio between fast and slow peers.

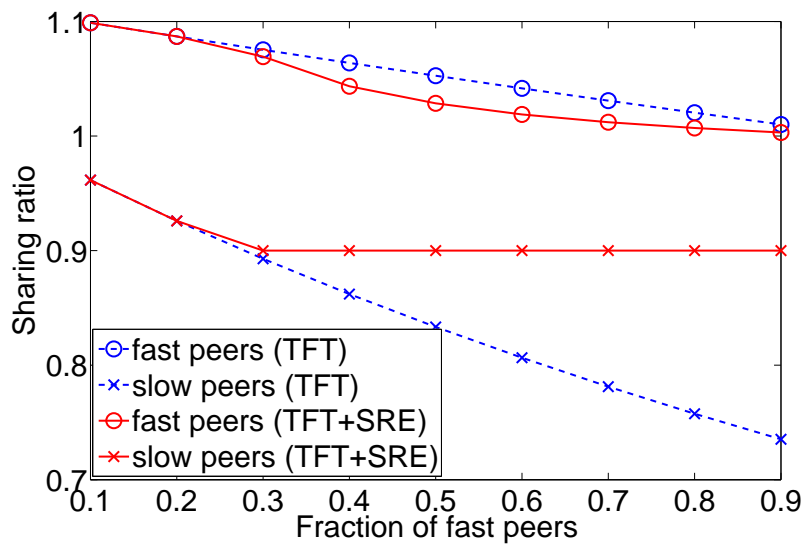
### 3.2.1 One swarm

We first analyze the effects of using SRE within one swarm only. In Figs. 3.2, 3.3, and 3.4, TFT alone is compared to TFT with SRE. As expected, SRE helps in enhancing reciprocity. In fact, the download speed of fast peers, who have a higher sharing ratio than slow peers under all considered scenarios, is higher when sharing ratio is enforced, as compared to the case when only TFT is applied. However we note that, for certain settings, i.e., when the threshold is too low (Fig. 3.3) or when the upload capacity ratio of the two classes of peers is small (Fig. 3.4), SRE has a limited influence on the performance. Hence, knowing the capacity distribution of peers is important to make the use of SRE most effective. Furthermore, in line with our previous results in Chapter 2, we note from Figs. 3.2(a), 3.3(a), and 3.4(a) that enhancing reciprocity deteriorates the overall download performance in the system.

These findings suggest that applying SRE within one swarm is a very strict way to enhance reciprocity: the performance of a peer depends only on his upload capacity. Hence, this method might be useful for administrators of private communities to exclude low-capacity peers.



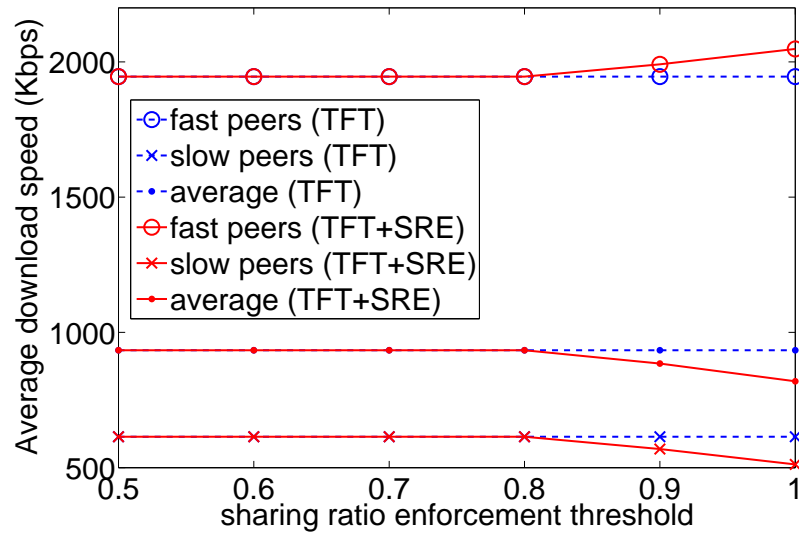
(a) Average download speed



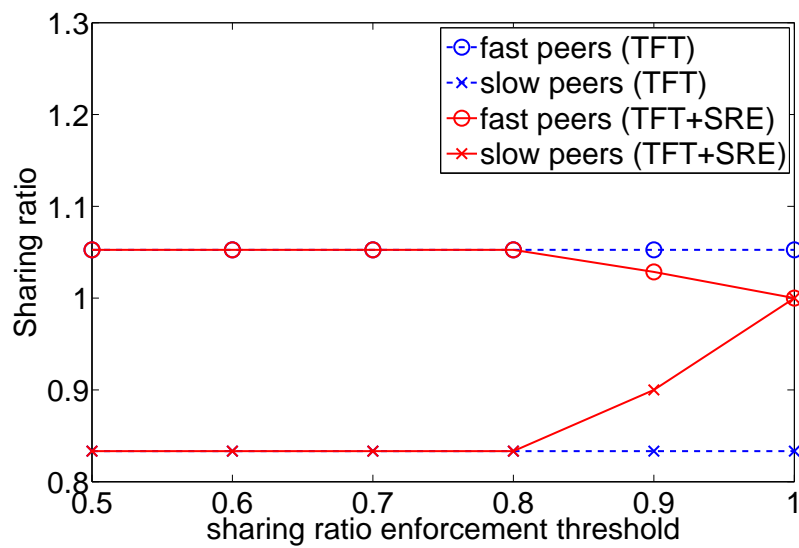
(b) Sharing ratio

Figure 3.2: SRE in one swarm: the influence of the fraction of fast peers.



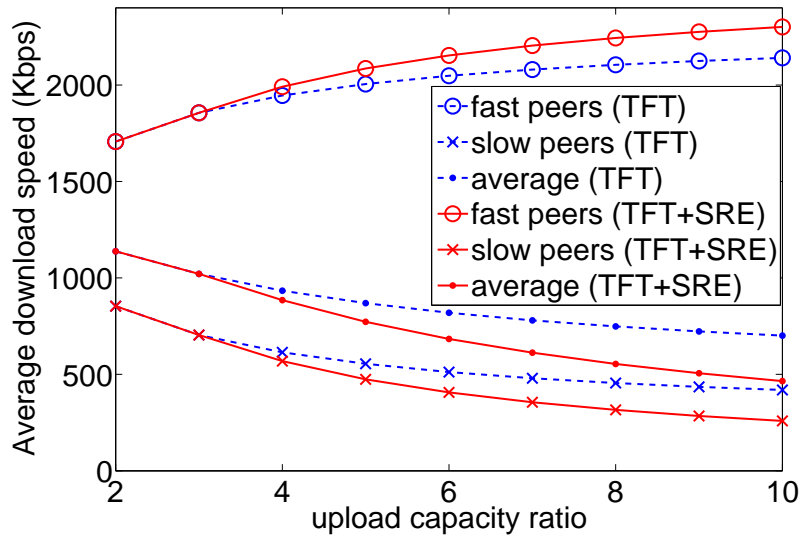


(a) Average download speed

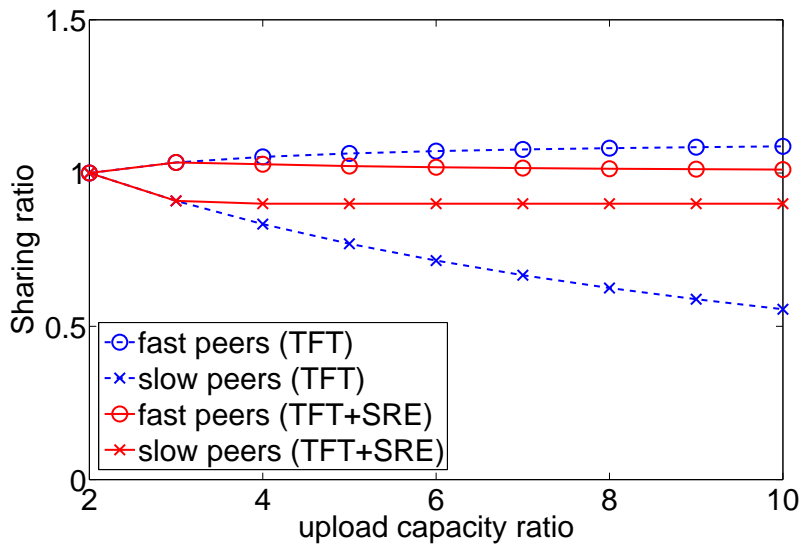


(b) Sharing ratio

Figure 3.3: SRE in one swarm: the influence of the threshold.



(a) Average download speed



(b) Sharing ratio

Figure 3.4: SRE in one swarm: the influence of the upload capacity ratio (where the ratio is expressed in terms of  $\mu_f/\mu_s$  and the average capacity of the population is constant).

### 3.2.2 Multiple swarms

In this section, we analyze the effects of SRE's when the sharing ratio of each peer is calculated across multiple swarms. As we can observe in Figs. 3.5, 3.6, and 3.7, the seeding work performed by peers in order to comply with the SRE threshold results in a higher download speed for peers in all classes, as well as in a better overall download performance in the system. Furthermore, this performance improvement is increased with a higher SRE threshold, a higher fraction of fast peers, or a larger upload capacity ratio of the two classes of peers.

Hence, applying SRE within multiple swarms translates into seeding incentives for those peers that cannot comply with the SRE threshold by only leeching. This represents a win-win situation where all peers are provided with a better service due to the increased bandwidth supply.

### 3.2.3 Comparison with the real world

In our model, we have only considered rational user behavior, i.e., peers only seed the minimum amount they are required to. In a real private community, this is not always the case. Several measurement studies [13,64] show that instead of following the enforcement rationally, many users prefer to seed more than they need, and thus achieve sharing ratios much higher than required. For example, in HDChina, a popular private community that has over 18,000 registered users, over 90% of the users have a sharing ratio higher than one [64]; while in another popular community, CHDBits, the top 250 users possess a sharing ratio higher than 10 [13]. This suggests that the potential risk of being expelled from a community due to an insufficient commitment psychologically manipulates users' behavior.

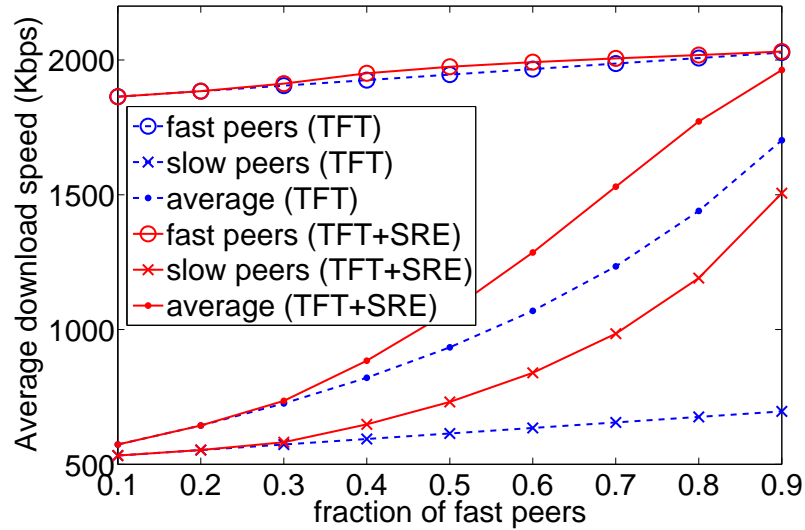
Given the existence of this irrational user behavior, the results derived in our model can be seen as a *lower bound* for the performance improvement provided by SRE, as we formally prove now.

**Theorem 3.1** *In a BitTorrent swarm where SRE is used and where peers upload at their full capacities, the average download speed  $d'_i$  of a peer in class  $i$  is not lower than  $d_i$ , regardless of the user behavior:*

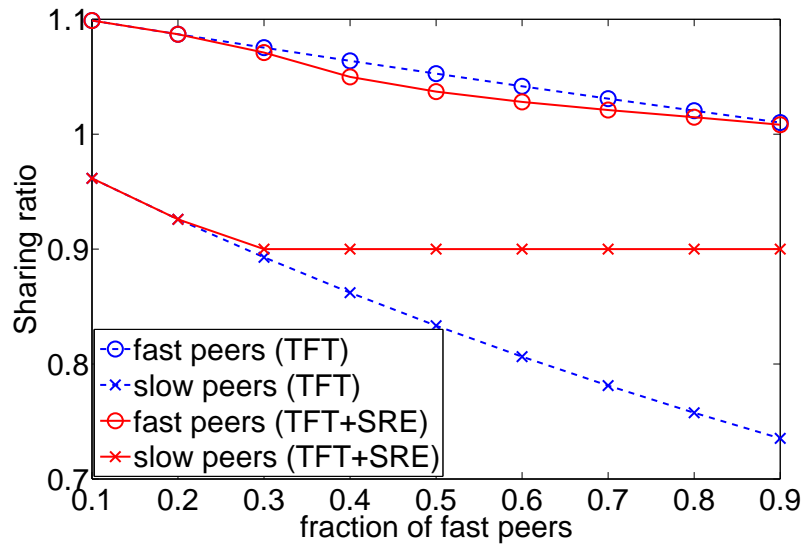
$$d'_i \geq d_i = \frac{\sum_j \omega_{ji} x_j \mu_j + \sum_{j \in \{j, \varphi_j < \alpha\}} \sigma_{ji} x_j \mu'_j}{x_i}. \quad (3.9)$$

(The expression for  $d_i$  comes from Eq.(3.7)).

**Proof:** Given the rational user behavior assumed in our model, the second term in the numerator of Eq.(3.9) specifies the minimum bandwidth allocated from class  $j$  seeders to



(a) Average download speed



(b) Sharing ratio

Figure 3.5: SRE in multiple swarms: the influence of the fraction of fast peers.

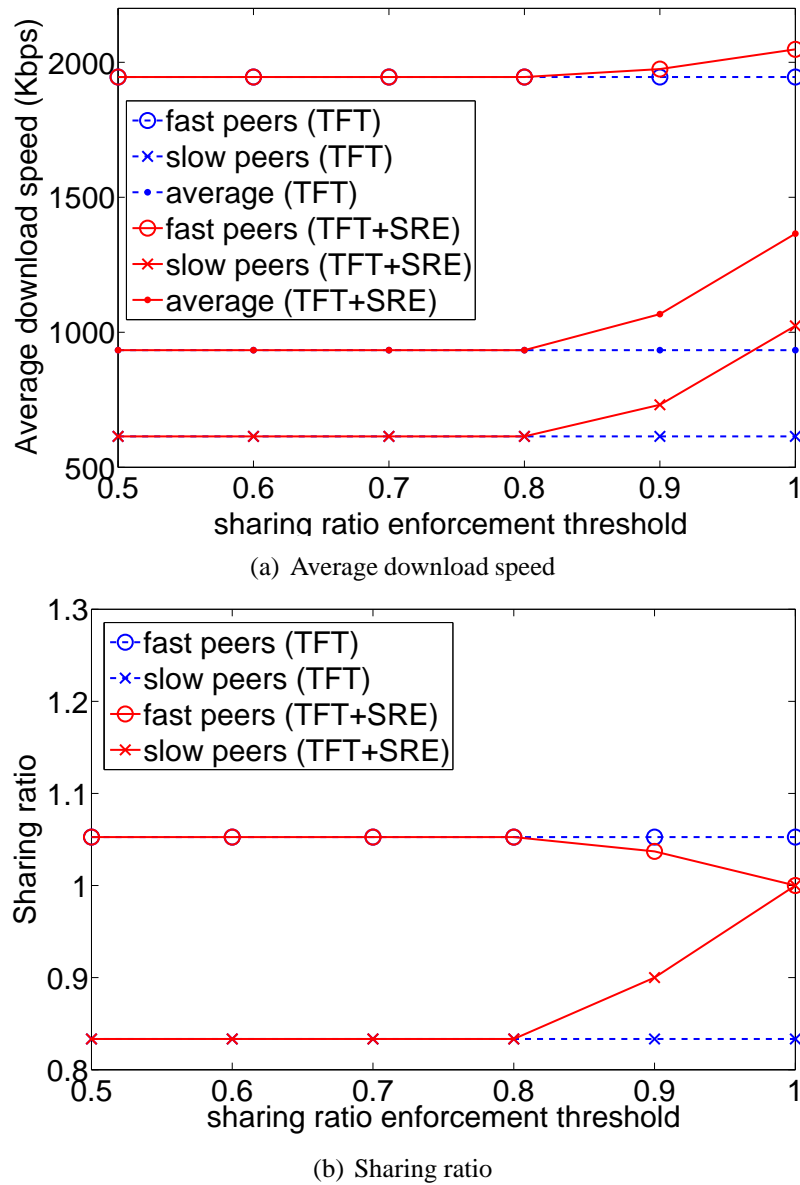
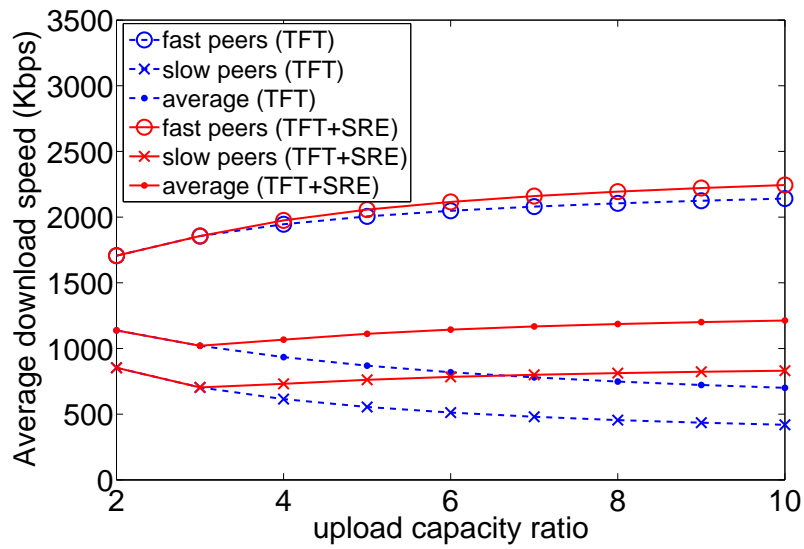
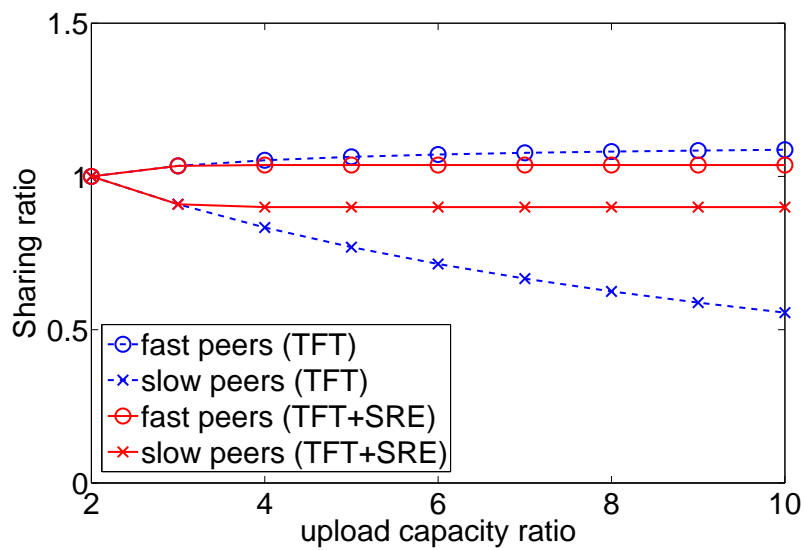


Figure 3.6: SRE in multiple swarms: the influence of the threshold.



(a) Average download speed



(b) Sharing ratio

Figure 3.7: SRE in multiple swarms: the influence of the upload capacity ratio (where the ratio is expressed in terms of  $\mu_f/\mu_s$  and the average capacity of the population is constant).

class  $i$  leechers, where  $\mu'_j = (\alpha - \varphi_j)F/t_j$ . Due to possible irrational user behavior, the sharing ratios achieved by peers in a private community swarm might be much diverse but, nevertheless, not less than  $\alpha$ . Let  $\mathcal{B}_i = (\mathcal{A}_i - \varphi_i)F/t_i$ , where  $\mathcal{A}_i$  is the average sharing ratio achieved by a peer in class  $i$ , i.e.,  $\mathcal{A}_i \geq \alpha$ . Similarly to our analysis in Section 3.1.4, we calculate the average download speed  $d'_i$  in a swarm with possible irrational user behavior as follows:

$$d'_i = \frac{\sum_j \omega_{ji} x_j \mu_j + \sum_j \sigma_{ji} x_j \mathcal{B}_j}{x_i}. \quad (3.10)$$

Because  $\mathcal{A}_i \geq \alpha$ , we have  $\mathcal{B}_i \geq \mu'_i \geq 0$ . Hence it follows that  $d'_i \geq d_i$ .

### 3.3 Related work

Most existing studies on BitTorrent incentive policies focus on TFT and its variations [30], [72], [90], [48]. To date, only few works analyze private BitTorrent communities. Andrade *et al.* [5] focus on the dynamics of resource demand and supply, and one of their most interesting findings is that a small set of users contributes most of the resources, but the users that provide more resources are also those that demand more. Rahman *et al.* [89] introduced and studied the credit crunch and crash problem, and they provide a novel credit intervention mechanism that proactively stops the system seizing. Zhang *et al.* [124] investigated hundreds of private trackers and depicted a broad and clear picture of the private community landscape. Chen *et al.* [13] compared system behaviors among 13 private trackers and 2 public trackers, and they showed their differences regarding user viscosity, single torrent evolution, user behaviors, and content distribution. While these studies all focus on demonstrating the properties of private communities based on measurements or simulations, we provide a theoretical model to analyze SRE's influence on the system performance. Liu *et al.* [64] developed a model to analyze SRE as well, based on a game theory approach. While they show the existence of a Nash Equilibrium and the conditions to achieve it, our model quantifies the lower bound of the performance improvement when using SRE, and we further study the influence of the SRE threshold and the bandwidth heterogeneity of the peers in the system.

### 3.4 Conclusions and future work

In this work, we have provided an analytical model that captures the essence of SRE adopted by private BitTorrent communities. Due to the existence of “irrational” seeding behavior observed in real private communities, our model represents a lower bound for the average download speed that peers can achieve. Based on our model, we show that apply-

ing SRE within a single swarm is a possible way for administrators of private communities to enhance reciprocity, or to exclude low-capacity peers. On the other hand, applying SRE across multiple swarms provides seeding incentives, and these seeding resources lead to a better overall download performance. We furthermore show that the performance of SRE is increased by 1) a higher fraction of high-capacity peers, 2) a higher SRE threshold, and 3) a larger disparity in the upload capacity of peers.



## Chapter 4

# Monetary schemes as contribution incentives: systemic risk and user-level performance

Online networks often require the cooperation and the contribution of their users. Effective contribution incentives are essential for the sustainability of the system. Barter schemes like Tit-For-Tat are based on reciprocity and have a number of limitations in, for example, the need for the coincidence of wants of the participants and the ineffectiveness in the long-term use. To tackle these problems, monetary schemes in economics are borrowed, modified, and utilized to design incentive policies. Under monetary schemes, virtual currencies are issued and used as an exchange media for the contributions and consumptions of users, implicitly or explicitly. In this way, users are incentivized to contribute, so as to keep some reserve of the virtual currency for their future needs.

Monetary schemes are becoming prevalent in nowadays online networks, ranging from the currencies used in online virtual worlds such as Second Life, to methods for resource allocation in wireless sensor network [17] and file sharing [43]. As one example, in recent years there has been a proliferation of so-called *private* BitTorrent communities aiming at providing contribution incentives beyond BitTorrent's original TFT. These communities employ private *trackers* that maintain centralized accounts and record the download and upload activity of each user. They apply policies to incentivize good overall upload / download behavior. One such well-known policy is the *credit-based* policy, which requires each member to maintain a positive *credit* (its total amount of upload minus its total amount of download). Another such policy is *Sharing Ratio Enforcement* (SRE), in which each member is required to keep its *sharing ratio* (the ratio between its total amounts of upload and download) at least equal to a threshold called the *SRE threshold*, which is set by the community administrator. The credit-based policy is an obvious monetary scheme. SRE, though not quite at the first glance, only differs in that it allows

a user to have negative credit (i.e., its download amount can be larger than its upload amount) and that when users do have negative credit, the amount of credit circulated in the system is dynamic. Under both policies, community members that cannot fulfill the requirements are first warned and then banned from downloading, or even expelled from the community.

In this chapter, we explore both the system-level dynamics and the user-level performance in communities adopting monetary schemes as contribution incentives. We use private BitTorrent communities as an example, but our analysis is applicable to any on-line network that adopts contribution enforcement policies, by generalizing the metrics for determining the credit and the sharing ratio from the upload and download amounts in a P2P file sharing system to any metrics representing contribution and consumption.

Considering a private community as an economic system, we analyze its system-level dynamics by studying its potential systemic risk. In economics, systemic risk is the risk of a collapse of an entire economic system or market [58]. We find that in private communities, too much credit distributed too evenly leads to a *crash* in which peers hold abundant credit and are not willing to contribute. Hence, the system seizes to zero throughput containing only leechers. Conversely, too little credit distributed over the peers leads to a *crunch* in which peers do not have enough credit to download, leading to a seized system containing only seeders.<sup>1</sup>

Even when crashes or crunches do not occur, i.e., when the system is *sustainable*, this only ensures that the system is able to function, but not how well it functions. Though many measurement studies [13, 64, 69, 124] have shown that the SRE-based and credit-based policies are very effective in boosting contribution levels in terms of high seeder-to-leecher ratios and the corresponding high downloading speeds, we argue that the abundant supply of bandwidth also has several negative effects such as excessively long seeding times that are often unproductive. To explore this, we analyze the user-level performance in sustainable private communities.

Our main contributions are as follows:

1. We demonstrate using simulations that in private communities credit crashes and crunches can occur, and we identify the conditions that lead to these extreme outcomes (Section 4.3);
2. We present a theoretical model that predicts whether a system will crash, crunch, or be sustainable over a defined time horizon. Based on this model we propose an

---

<sup>1</sup>A real world example of the crash and crunch is the story of the Capitol Hill Baby Sitting Co-op [56], which was a group of parents who agreed to cooperate to babysit. A crunch happened when most people wanted to save up coupons: they looked for an opportunity to babysit but there was little demand. Later when more coupons were issued a crash happened: most people felt they had enough coupons so they didn't want to babysit, leaving the system with huge demand but no supply.

adaptive credit policy that helps the system to avoid crashes and crunches (Section 4.3);

3. We show that the users in sustainable private communities, while achieving high system-wide downloading speeds, are forced to seed for excessively long times, during which their upload capacity utilizations are quite low (Section 4.4). Further, when the popularity of a swarm decreases over time, peers that join the swarm not early enough will have to seed for much longer durations than peers who join (strategically) at the beginning of the swarm (Section 4.5);
4. We propose and evaluate by means of simulations four new strategies that alleviate these problems while still maintaining a reasonable system-wide downloading speed (Sections 4.4 and 4.5).

## 4.1 Support from real world observations

To support our later analysis, we first present real world observations of two private communities, CHDBits.org [12] and Bitsoup.org [10]. CHDBits and Bitsoup both require the users to maintain sharing ratios larger than the threshold of 0.7. The trackers of CHDBits collect information that is periodically reported by the BitTorrent clients of its users, which is displayed in the form of HTML pages available to only its users. We crawled these trackers in May 2011. For each user in CHDBits, we collected the information on its user profile page including the upload and download amount, the seeding time, and the sharing ratio. For each torrent, we collected the information of its published date, and its numbers of seeders and leechers at the time of snapshot. In total, information on all the 31,547 registered users and 40,040 torrents was obtained. For Bitsoup, we use the traces published in [5] that report the user activity of 84,007 users in 13,741 torrents during a period of two months.

### 4.1.1 The existence of over-seeding behavior

A previous work from Tribler group [37] has shown that users who always seed can, counter-intuitively, lead private communities to poor performance due to a credit crunch, in which a few peers accumulate much of the credit and deprive others of the opportunity of downloading. This implies that the user behavior can significantly influence the system performance. Inspired by this finding, we first demonstrate the user behavior as observed in the real world, based on which we later analyze the system-level and user-level performance of private communities.

In CHDBits, maintaining a sharing ratio equal to the SRE threshold is sufficient for a user to start downloading a new file. However, we observe that not all the users behave

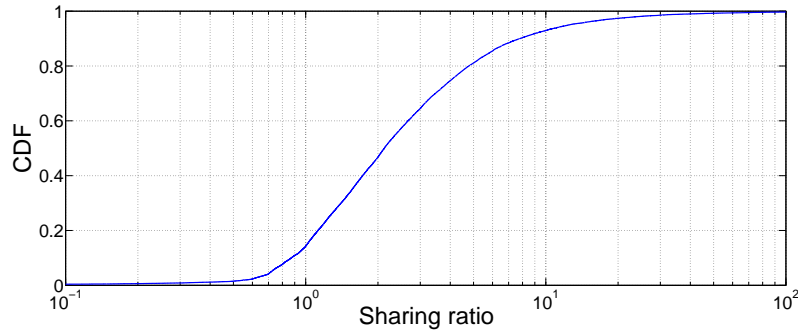


Figure 4.1: Over-seeding behavior: the CDF of the sharing ratios of peers in CHDBits.org.

like this. As shown in Fig. 4.1, more than 95% of the users in CHDBits keep sharing ratios higher than 0.7 and more than 50% of the users keep them higher than 2. This phenomenon of peers seeding more than required and achieving sharing ratios that are (much) higher than the SRE threshold has also been observed in many other communities [64].

From the above observation we abstract two user behaviors for our later analysis<sup>2</sup>, *lazy-seeding* and *over-seeding*. *Lazy-seeding* peers seed the minimum amount required by the enforcement policies. They represent the users who are download-oriented, i.e., who only seed enough to maintain adequate sharing ratios or credit to be able to start new downloads. On the other hand, *over-seeding* peers are deposit-oriented, and always maintain sharing ratios (much) higher than required. The behavior of such peers may be triggered by various motivations such as altruism, a desire to be part of the rich elite of the community, or a habit of storing credit for the future. In line with the terminology used in economics, *over-seeding* peers can be understood as *hoarders* as their behavior essentially amounts to hoarding credit.

### 4.1.2 The oversupply

The main motivation for implementing credit or SRE policies is to close the gap between bandwidth demand and supply as observed in public BitTorrent communities, where there is significantly more demand than supply [69]. However, the presence of *over-seeding* peers completely reverses the situation and in private communities, swarms tend to be extremely oversupplied.

At the time of the crawling, CHDBits had 33,041 active swarms (with at least one leecher or one seeder), among which 26,402 swarms (79.9%) had no leechers at all. As shown in Fig. 4.2(a), 40% of the swarms with no leechers still had at least 5 seeders, and

<sup>2</sup>We use the abstracted behaviors instead of the real trace because we intend to identify what behavior influences what performance.

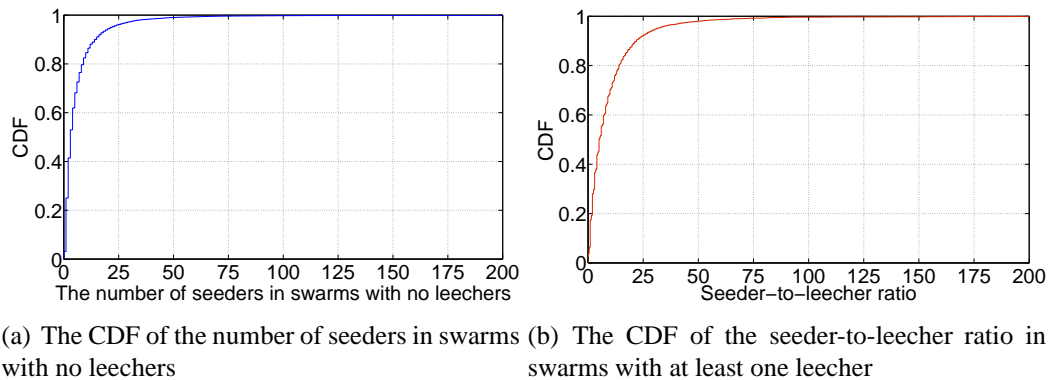


Figure 4.2: Oversupply in CHDBits swarms.

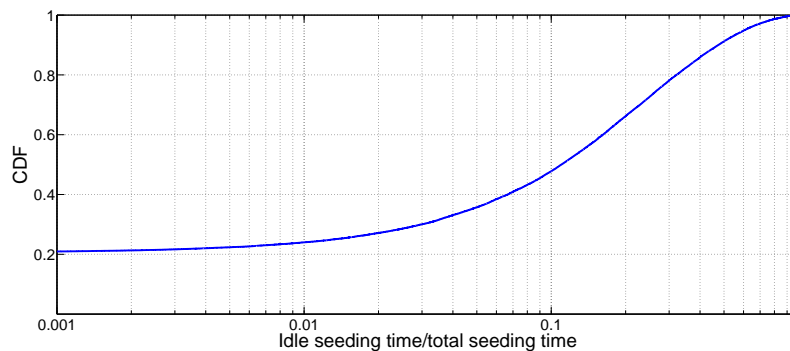


Figure 4.3: Unproductive seeding: The CDF of the fraction of idle seeding time of peers with sharing ratios smaller than 1 in BitSoup.org.

5% of these swarms even had more than 20 seeders. For swarms with at least 1 leecher, the *seeder-to-leecher ratio* ( $SLR$ ) is quite high: as shown in Fig. 4.2(b), 50% (5%) of these swarms had an  $SLR$  of at least 6 (30). We see clearly that a majority of the swarms are heavily oversupplied. In such swarms, intuitively it is difficult for seeders to perform any actual uploads due to the insufficient demand and unsatisfied supply. We validate our speculation through the following observation.

### 4.1.3 Unproductive seeding

It is clear that in order to achieve high sharing ratios, peers need to spend considerable amount of seeding time. In the case of over-seeding peers, long seeding times are to be expected. However, we observe that even many peers with small sharing ratios suffer from excessively long seeding times, and a significant part of their seeding time is spent idle without being able to upload anything to others. As a consequence, they have to wait for a long period until their sharing ratios are high enough to start new downloads.

Fig. 4.3 shows the CDF of the fraction of idle seeding time of peers with sharing ratios

smaller than 1 in BitSoup. We see that 10% of these peers spend at least half of their seeding time idle. Note that Fig. 4.3 only shows the fraction of idle seeding time. It can be conjectured that the fraction of seeding time that is not completely idle yet still yields very low upload speed, would be much higher. We term this situation as *unproductive seeding* and we hypothesize that it is due to the oversupply under credit-based or SRE-based schemes.

Based on these observations, in later sections we analyze the system-level credit dynamics and user-level performance in private communities. Before that, we first introduce the basic model in the following section.

## 4.2 Model description

In this section we will explain the credit-based and SRE-based incentive policies, and our model of communities that employ one of these policies.

### 4.2.1 Credit-based versus SRE-based policies

The credit-based and SRE-based policies are essentially very similar, in a way that they can be understood as variations of each other. The idea behind both policies is that every peer has to maintain at all times  $t$  a certain relation between the total amount  $u(t)$  it has uploaded and the total amount  $d(t)$  it has downloaded since it entered the community until time  $t$ . The credit-based policy requires users to keep non-negative credit, i.e., to ensure that  $u(t) - d(t) \geq 0$ , while the SRE-based policy requires users to keep a minimum sharing ratio  $SR(t) = u(t)/d(t)$ , i.e., to ensure that  $SR(t) \geq \alpha$ , where  $\alpha$  is the SRE threshold<sup>3</sup>. When  $\alpha = 1$  in the SRE policy, the SRE-based and credit-based policies coincide.

By enforcing non-negative credit in the credit-based policy, the exchanging of data by peers does not generate new credit, and the total amount of credit in the community is always equal to zero (or to the sum of the initial credits allocated to the peers by the community administrator). In contrast, an SRE-based policy allows users to have negative credit (i.e., to have  $u(t) - d(t) < 0$ , which means that  $SR(t) < 1$ ). Holding negative credit increases the amount of credit among the peers with positive credit in the system—in other words, by holding negative credit a user is essentially minting credit. More precisely, the total credit minted by a user in an SRE-based community with  $SR(t) < 1$  until time  $t$  is:

$$d(t) - u(t) = (1 - SR(t))d(t),$$

---

<sup>3</sup>Throughout this chapter we assume  $\alpha \leq 1$ , as most private communities do [10,40]. It is not reasonable to have  $\alpha > 1$  since, given the conservation law, it is not possible for all users to achieve sharing ratios larger than one simultaneously.

which is bounded by  $(1 - \alpha)d(t)$ . As the sharing ratios of peers fluctuate, SRE-based communities hold a dynamic amount of credit circulating in the system.

### 4.2.2 The basic model

We consider a community that is either credit-based or SRE-based. The community comprises a set of  $s$  swarms<sup>4</sup> each associated with a file of size  $F$  (expressed in number of pieces or units), and a set of  $N$  peers each with upload capacity  $U$ .<sup>5</sup> We assume no limit on the download capacity of peers. The download model follows the TFT mechanism in BitTorrent, with seeders uploading units to leechers and leechers exchanging units with each other. In reality, a peer can participate in multiple swarms simultaneously, with its bandwidth shared among all the swarms. However, since the sharing ratio is aggregated over all the swarms, we assume that at any time a peer only participates in one swarm, either as a leecher or a seeder.

The operation of the model is based on *cycles* representing units of time. In every cycle, a peer either uploads and/or downloads data or is idle, and at the end of every cycle, it may switch swarms. Peers attempt to download all  $s$  files in random order.

In a credit-based community, every peer  $p$  is initialized with an amount  $C_p$  of credit, and in an SRE-based community, every peer is initialized with a download amount equal to  $F$  and a sharing ratio that is a uniformly random number between 0 and 2. A peer can and will only start leeching its next file if its credit or its sharing ratio is at least equal to its *target threshold*, otherwise it continues seeding the current file.

Based on real-world observations, we implement two user behaviors: lazy-seeding and over-seeding (see details in Section 4.1.1). The target threshold of a lazy-seeding peer is an amount  $F$  of credit in a credit-based community (enough to start and complete leeching a new file) and a sharing ratio equal to the SRE threshold in an SRE-based community. The condition for a lazy-seeding peer  $p$  at time  $t$  to stop seeding is  $c_p(t) \geq 0$ , with:

$$c_p(t) = \begin{cases} u_p(t) - d_p(t) + C_p - F & \text{credit-based,} \\ u_p(t) - \alpha d_p(t) & \text{SRE-based,} \end{cases} \quad (4.1)$$

where  $u_p(t)$  and  $d_p(t)$  represent the total amounts of upload and download of peer  $p$ . Over-seeding peers behave in a similar way, but in both credit-based and SRE-based communities they aim at large sharing ratios. Throughout this chapter we choose a sharing ratio of 2 as the default target threshold for over-seeding peers<sup>6</sup>.

---

<sup>4</sup>We assume the number of swarms to be large enough that even with no injection of new swarms, users still have enough swarms to download from.

<sup>5</sup>Bandwidth heterogeneity does not change whether a system will crash or crunch, but it does influence the user-level performance, for which we examine both bandwidth homogeneous and heterogeneous systems in Section 4.4.2.

<sup>6</sup>We have run several tests using different values for the threshold and the results show that the tendency

## 4.3 System-level performance: the systemic risk

In this section, we explore the systemic risk of monetary schemes. We start with simulations to demonstrate the consequences of the systemic risk and we identify the conditions that lead to these extreme outcomes. Then, we present a theoretical model that predicts when it may happen. Finally, we propose an adaptive credit policy that helps the system to avoid the systemic risk and to be sustainable.

### 4.3.1 The crash and crunch

In this section, we perform a number of simulations to explore the credit dynamics in private communities, focusing on analyzing the conditions under which a community will crash, crunch, or be sustainable. We define a *crash* as a situation in which due to credit abundance, peers are not incentivized to contribute and the system completely seizes up, providing no upload or download to any peers. We define a *crunch* as a situation in which due to credit shortages, peers cannot afford new downloads and the system seizes providing no upload or download to any peers. We define a community to be *sustainable* if it does not crash or crunch.

#### Experimental setup

We consider a closed system without new peer arrivals. Peer arrivals bring credit into the system and make it difficult to identify whether the underlying credit dynamics is due to the enforcement policy or to the new credit. In fact, in reality many private communities are (nearly) closed [12, 40]. For example, CHDBits hardly has any open registration and new members can only be admitted by extremely restricted invitation<sup>7</sup>.

The simulation is based on the basic model introduced in Section 4.2, with  $N = 1000$ ,  $s = 100$ ,  $F = 10$  units<sup>8</sup>, and  $U = 4$  units per cycle. We choose  $\alpha = 0.7$  as the default value of the SRE threshold<sup>9</sup>, as this value is used in many private communities, e.g., [10, 40]. For each experiment we perform 10 independent runs, and each run is executed for 2000 cycles.

---

of the problem is the same.

<sup>7</sup>As we will show later in this section and in Section 4.4.2, in a closed private community with over-seeding peers crunches easily happen and new peer arrivals bring credit, which alleviates the potential systemic risk. We conjecture this is the reason why CHDBits, as well as other private communities, from time to time (several times a year) accept a certain number of open registrations, instead of merely intending to share their resources with a larger user base.

<sup>8</sup>The small file size means the simulation runs produce results at a large scale of granularity. We also performed runs with  $F = 100$  and found no significant difference in results.

<sup>9</sup>We have run several tests using different values for  $\alpha$ . Results show that the tendency of the problem stays the same, but with different speeds of entering crash or crunch.



frac.of <b>rich</b> at start	avg. <b>throughput</b> (std.dev)	avg.frac.of <b>seeders</b> (std.dev)	final <b>state</b>
0.1	0.000 (0.000)	1.000 (0.000)	crunch
0.3	0.218 (0.001)	0.953 (0.005)	sustain
0.5	0.777 (0.002)	0.769 (0.018)	sustain
0.7	0.968 (0.004)	0.506 (0.018)	sustain
0.8	0.587 (0.478)	0.249 (0.204)	sustain/crash
0.9	0.001 (0.000)	0.000 (0.000)	crash

Table 4.1: Sustainability of the credit-based system.

We consider three performance metrics, namely the average throughput, the fraction of seeders, and the state of the system at the end of the simulation. The throughput is expressed as the total amounts of units of data exchanged in the system over an entire run, normalized to the highest one observed in all the experiments. The state of the system indicates whether the system crunches, crashes or sustains.

### Credit-based: constant credit

As discussed in Section 4.2, the amount of credit in a credit-based community is always equal to the initial credit allocated by the community administrators. In this experiment, we vary the fraction of peers who are given an initial credit of  $F$  (and other peers are given zero credit), which we call rich peers, thus generating different levels of credit in the system.

*Populations of lazy-seeding peers:* We first show the results of the system containing only lazy-seeding peers in Table 4.1. When the fraction of rich peers is initialized to 0.3, 0.5, and 0.7, we see sustainable outcomes<sup>10</sup> with increasing throughput and a smaller number of seeders at the end of the simulation. This is intuitive since as the amount of credit in the system increases, fewer peers are poor, and hence more exchange of data can occur.

In the crunch state, where only 10% of peers are initialized as rich, the system is composed of all seeders by the end of the run, and hence, no exchange of data can occur. Conversely, in the crash state, where 90% of peers are initialized as rich, all peers are leechers by the end of the run, which again means no exchange of data. Inspection of individual runs shows that crunches and crashes happen quickly—within the first ten cycles or so. This is reflected in the low (almost zero) throughput under crash and crunch states.

It is interesting to see that when the initial fraction of rich peers is set to 0.8, both sustain and crash outcomes can occur. This is reflected in the high variance of the throughput.

<sup>10</sup>We have run extended runs up to 20,000 cycles and find that the sustainable outcomes are maintained.

frac.of <b>over-seeding peers</b>	avg. <b>throughput</b> (std.dev)	avg.frac.of <b>seeders</b> (std.dev)	final <b>state</b>
0.1	0.0093 (0.0012)	0.0000 (0.0000)	crash
0.2	0.2046 (0.2103)	0.0037 (0.0082)	crash/sustain
0.3	0.8910 (0.0041)	0.1487 (0.0141)	sustain
0.4	0.9865 (0.0090)	0.4212 (0.0243)	sustain
0.5	0.1436 (0.0083)	1.0000 (0.0000)	crunch

Table 4.2: Sustainability of the SRE-based system.

Here we are very close to the threshold leading to a crash and we find path dependency based on initial random conditions leading to either a high sustainable throughput, or a sudden crash otherwise.

*Populations containing over-seeding peers:* We find that introducing *any* number of over-seeding peers into the system eventually leads to a crunch, and the speed of the crunch depends on the number of over-seeding peers. This is intuitive since in our experiments, over-seeding peers seed (to hoard credit) until they have a sharing ratio larger than 2. This means that as the simulation progresses, the over-seeding peers eventually hold all the credit in the system and a crunch is inevitable.

### **SRE-based: dynamic credit**

As discussed above, a credit-based community keeps a delicate constant amount of credit which, if not properly set, will lead the system to crunch or crash. On the other hand, as stated in Section 4.2, an SRE-based system keeps dynamic credit by allowing peers to have sharing ratios less than one, i.e., to have negative credits. Hence, essentially lazy-seeding peers in an SRE-based community inject credit into circulation, and as in a credit-based community, over-seeding peers absorb credit from circulation.

Intuitively, an SRE-based system cannot be sustainable if all peers are lazy-seeding: soon they will inject too much credit into circulation, which eventually leads the system to a crash. However, as we have shown in Section 4.1.1, in private communities over-seeding peers always exist and they absorb credit from circulation. Hence, in an SRE-based system with over-seeding peers, the effect of credit-injecting by lazy-seeding peers can be alleviated and the system might eventually be sustainable.

We run several simulations to validate the above hypotheses. We consider an SRE-based system in which we vary the fraction of over-seeding peers to assess their influence on the credit dynamics. Table 4.2 shows the simulation results. Consistent with our intuition, a certain fraction of over-seeding peers (0.3 and 0.4 in our experimental settings) does lead the SRE-based community to be sustainable. A too small or a too large fraction of over-seeding peers (0.1 and 0.5 in our experimental settings), on the other hand,

eventually leads the system to crash or crunch.

### 4.3.2 Predicting crashes and crunches

In this section, we will derive (approximate) conditions for predicting whether the system will crunch or crash.

In the model introduced in Section 4.2, suppose that at time  $t$ , a swarm  $\ell$  has  $x^\ell(t)$  leechers and  $y^\ell(t)$  seeders. Denoting the fraction of the file that a leecher  $x_i^\ell$  has already downloaded by  $p_i^\ell(t)$ ,  $x_i^\ell$  needs to spend an amount  $\alpha(1 - p_i^\ell(t))F$  of credit to finish its download. We define  $L_i^\ell(t)$  and  $R_i^\ell(t)$  as the sets of peers that have fewer or more pieces of the file than peer  $i$ , respectively (for a seeder  $y^\ell(t)$ ,  $L_i^\ell(t)$  consists of all leechers and  $R_i^\ell(t)$  is empty). We assume that peer  $i$  only downloads from peers in  $R_i^\ell(t)$ , and only uploads to peers in  $L_i^\ell(t)$  (this is not quite true in BitTorrent, which makes the conditions that we will derive approximations). We further assume that the credit paid by peer  $i$  is equally shared by all peers in  $R_i^\ell(t)$ . Hence, if the situation (in terms of  $L_i(t)$  and  $R_i(t)$ ) does not change from time  $t$  onward, peer  $i$  can earn an amount  $Q_i^\ell(t)$  of credit from the peers in  $L_i^\ell(t)$ , where

$$Q_i^\ell(t) := \sum_{j \in L_i^\ell(t)} \frac{(1 - p_j^\ell(t))F}{|R_j^\ell(t)|}.$$

Let  $X_\ell(t)$  and  $Y_\ell(t)$  respectively represent the sets of leechers and seeders that, assuming that the situation does not change from time  $t$ , are able to achieve their target thresholds and start new downloads. Together with Eq. (4.1), we have:

$$\begin{aligned} X_\ell(t) &:= \{x_i^\ell : c_{x_i^\ell}(t) + Q_i^\ell(t) - \alpha(1 - p_i^\ell(t))F \geq 0\}, \\ Y_\ell(t) &:= \{y_j^\ell : c_{y_j^\ell}(t) + Q_j^\ell(t) \geq 0\}. \end{aligned}$$

Now we estimate the remaining download time of leechers and the remaining seeding time of seeders for the current file. Here we assume that during the upload process, leechers and seeders alike upload with their full capacity  $U$  and distribute their upload capacity equally across all the leechers they are uploading to. The estimated remaining download time  $T_i^\ell(t)$  of leecher  $x_i^\ell$  can be expressed as

$$T_{x_i^\ell}(t) := \frac{(1 - p_i^\ell(t))F}{\sum_{k \in R_i^\ell(t)} \frac{U}{|L_k^\ell(t)|}}.$$

Similarly, the estimated remaining time for a seeder  $y_j^\ell$  to achieve its target threshold and stop seeding is:

$$T_{y_j^\ell}(t) := \max\{0, -c_{y_j^\ell}(t)\}/U,$$

where  $-c_{y_j^\ell}(t)$  (if positive) represents the credit  $y_j^\ell$  still needs to earn to achieve its target threshold.

We can now formulate the *condition for a crunch* to happen in the system as the condition that the sets  $X_\ell(t)$  and  $Y_\ell(t)$  are both empty for all swarms  $\ell$  at some time  $t$ , because that no leechers or seeders are able to earn enough credit to leave their swarms. As a consequence, by the time that the last leecher finishes its download, there will be no exchange of credit in the whole system.

In order to formulate the condition for a crash to happen, let  $P_\ell(t) := \{x_i^\ell : x_i^\ell \notin X_\ell(t)\}$  be the set of leechers in swarm  $\ell$  who will need to seed after finishing their current downloads in order to achieve their target thresholds. Then the *condition for a crash* to happen is

$$|Y_\ell(t)| = y^\ell(t) \text{ and } \min_{k \in P_\ell(t)} T_{x_k^\ell}(t) > \max_{j \in Y_\ell(t)} T_{y_j^\ell}(t),$$

for all swarms  $\ell$  at some time  $t$ . To see this, note that the system crashes if there are no seeders. The first part of the condition above says that all the seeders in the system will be able to earn enough credit to leave their swarms. If in addition none of the leechers in  $P_\ell(t)$  can finish its download before the last existing seeder leaves the swarm and if this happens to all the swarms (the second part of the condition), then the whole system will end up with no seeders and seize completely, i.e., a credit crash will occur.

### 4.3.3 Adaptive credit for sustainability

Based on the experimental and theoretical results of Sections 4.3.1 and 4.3.2, we have designed a novel *adaptive credit intervention mechanism* to avoid crashes and crunches. At each cycle, we check the conditions for crunches and crashes derived in Section 4.3.2, thus obtaining early warnings for potential crunches or crashes. When we find that the system is destined for a crunch, a new credit policy called *freeleech*<sup>11</sup> will be applied. As a consequence, leechers do not pay any credit for downloading, but seeders and other uploaders are still credited for uploading. Hence, new credit is injected into the system. Credit injection for stimulating the economy has often been used successfully in real world situations [98]. When we find that the system is destined for a crash, it applies a *freeseed* policy in which seeding peers (and uploading leechers) do not receive any credit for uploading, but leechers still pay credit for downloading. Hence, credit is removed from the system.

We use the credit-based system as an example to evaluate our strategies, but the same analysis can be applied to an SRE-based system. The experimental setup is the same as in Section 4.3.1.

---

<sup>11</sup>Freeleech is sometimes also used in existing private communities such as CHDBits, but in a more empirical manner.

frac.of rich at start	avg.throughput (std.dev)	avg.frac.of seeders (std.dev)	final state
0.1	0.234 (0.026)	0.948 (0.012)	sustain
0.3	0.311 (0.004)	0.941 (0.005)	sustain
0.5	0.782 (0.002)	0.769 (0.009)	sustain
0.7	0.968 (0.001)	0.512 (0.024)	sustain
0.8	0.976 (0.001)	0.535 (0.015)	sustain
0.9	0.995 (0.002)	0.575 (0.027)	sustain

Table 4.3: System sustainability with adaptive credit.

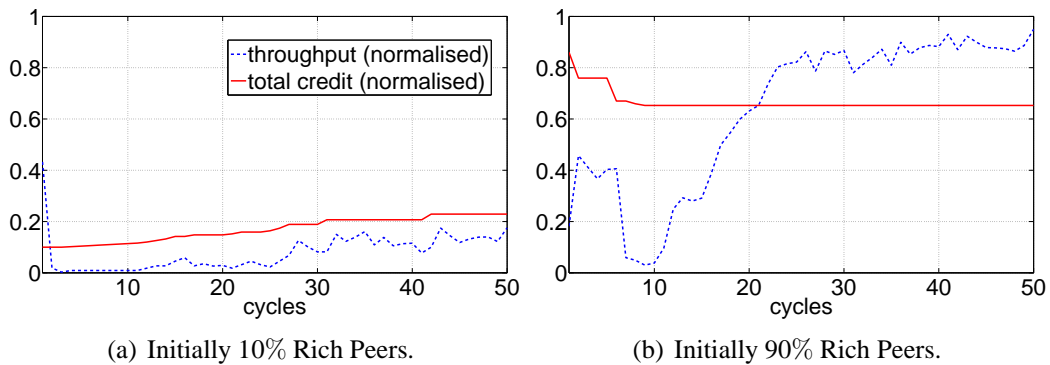


Figure 4.4: The normalized throughput and credit in the system with the adaptive credit intervention mechanism.

### Populations of lazy-seeding peers

Table 4.3 shows the performance of our adaptive credit intervention mechanism in a credit-based system containing only lazy-seeding peers. All runs produce a sustainable outcome, including those initialized with fractions of 0.1 and 0.9 of rich peers, which previously led to crunches and crashes when the mechanism is not applied (see Section 4.3.1). This indicates that the model in Section 4.3.2 gives early enough warning for the adaptive credit policy to avoid crashes and crunches.

Fig. 4.4 shows the results of two runs initialized with fractions of 0.1 and 0.9 of rich peers. A crunch is avoided in Fig. 4.4.(a) via the activation of freeleech at several cycles—note the increase in credit over time. A crash is avoided in Fig. 4.4.(b) via the activation of freeseed in the initial cycles—note the decreasing credit over time.

### Populations containing over-seeding peers

As stated in Section 4.3.1, any number of over-seeding peers in a credit-based community will eventually lead to a crunch, due to the increasing amounts of credit they hoard. In or-

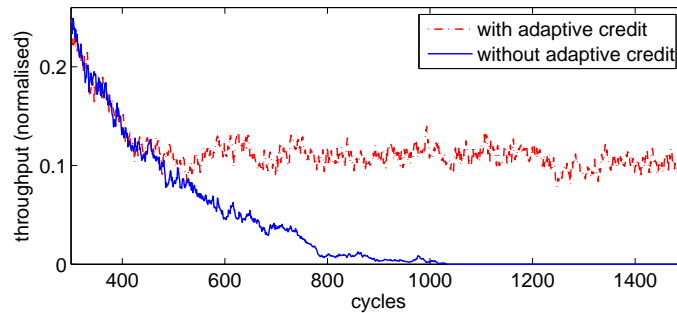


Figure 4.5: The normalized throughput with and without the adaptive credit intervention mechanism (50% rich peers and 1% over-seeding peers).

der to test whether our adaptive credit intervention mechanism can deal with this extreme condition, we ran several simulations in which a small subset (1%) of the population are over-seeding peers.

Fig. 4.5 shows the throughput in the system with and without the adaptive credit intervention mechanism. As can be seen, without the mechanism, the system eventually crunches, whereas with the mechanism, the system is sustainable. However, the throughput of the system in the latter case is still very low. Although new credit is injected each time a crunch is predicted, this additional credit is eventually collected by the over-seeding peers and the process repeats. We believe that this is due to the fact that the adaptive credit intervention mechanism does not attempt to optimize the system, but rather only to avoid a crunch. In later sections we provide a more thorough analysis on optimizing the system, i.e., improving the user-level performance.

## Discussion

The aim of adopting the freeleech and freeseed policies is to avoid crunches and crashes, which is actually achieved, but at the potential cost that the original incentive for contributions is temporarily suspended. It could be argued that this could lead to reduced performance if users learn to game the system by only downloading during freeleech periods and not seeding during freeseed periods.

A refinement that will help preserve incentives even during freeseed and freeleech periods, is to reduce the freeseed and freeleech “tax” amount. Then, rather than having leechers not pay anything at all for downloading and seeders not being credited for uploading, they can be charged or credited for a fraction, say 50%. Any value more than 0% still provides incentives for contribution. Furthermore, the taxation amount can also be variable, and can be applied in a continuous fashion, rather than getting triggered at the extreme conditions of crash and crunch. We explore this later in Section 4.4.3.

Until now, we have analyzed the sustainability of a P2P community that adopts a

credit-based or SRE-based policy. However, the sustainability of the system only ensures that the system is able to function, but does not guarantee it will function well (recall Fig. 4.5 for an example of a sustainable system with low performance). To explore this, in the following sections we analyze and improve the user-level performance in sustainable P2P communities. There, we take sharing ratio enforcement as an example, but our analysis is also applicable to the credit-based policy, since it is only a special case of SRE with a threshold equal to one.

## 4.4 User-level performance: the positive and negative effects of SRE

In this section, we analyze the user-level performance in private communities. First, we present a theoretical model that captures the bandwidth supply and demand in the community. Then, we show through simulations the positive and negative effects of SRE, followed by a proposal of four remedies. Finally, we evaluate and demonstrate the effectiveness of our strategies.

### 4.4.1 A simple model

From the two real world observations described in Section 4.1, in this section we propose a fluid model to analyze the user-level performance in private BitTorrent communities.

We follow a similar fluid modeling approach as in [48, 72, 85] and extend it by including SRE and considering multiple user behaviors derived from real world observations. Throughout this chapter we assume that the SRE threshold  $\alpha \leq 1$ , which is the case for most private communities. Remember that when  $\alpha = 1$ , the system also represents the one that adopts credit-based policy.

We divide peers into  $N$  different classes according to their upload capacities. Let  $U_i$  represent the upload capacity of peers in class  $i$ . Without loss of generality we assume that the download capacity of peers is not a bottleneck. The notation we use is shown in Table 4.4.

As mentioned earlier we consider two user behaviors: lazy-seeding and over-seeding. To better understand the effect of SRE and the over-seeding behavior, we consider an idealized scenario of a swarm, in which there are  $s_i$  over-seeding peers in class  $i$  with an infinite desired threshold for their sharing ratios. This implies that they stay in the swarm as seeders indefinitely. Lazy-seeding peers in class  $i$  join the swarm as leechers with an arrival rate equal to  $\lambda_i$  and sharing ratios equal to 0. After they finish their downloads, they calculate the sharing ratios they have achieved and, if necessary, they seed in this swarm until their sharing ratios reach the SRE threshold  $\alpha$ . Then they leave the swarm.

Notation	Definition
$F$	the size of the file shared in a swarm.
$\alpha$	the SRE threshold.
$\lambda_i$	the arrival rate of leechers in class $i$ .
$U_i$	the upload capacity of a peer in class $i$ .
$u_i$	the average upload speed of a peer in class $i$ .
$s_i$	the number of over-seeding seeders in class $i$ .
$T_i$	the average seeding time for lazy-seeding peers in class $i$ .

Table 4.4: Notation of our BitTorrent model

According to TFT, during the leeching process peers favor other peers who have recently reciprocated to them most. In this way, peers are roughly clustered according to their capacities, and peers with similar capacities have similar performance [48, 72]. Let  $x_i(t)$  and  $y_i(t)$  represent the number of leechers and lazy-seeding seeders in class  $i$  at a particular time  $t$ , and let  $T_i$  represent the average seeding time of lazy-seeding peers in class  $i$ , then the evolution of  $x_i(t)$  and  $y_i(t)$  can be described as:

$$\begin{aligned} \frac{dx_i(t)}{dt} &= \lambda_i - \frac{x_i(t)d_i(t)}{F}, \\ \frac{dy_i(t)}{dt} &= \frac{x_i(t)d_i(t)}{F} - \frac{y_i(t)}{T_i}, \quad i = 1, 2, \dots, N, \end{aligned} \quad (4.2)$$

where  $d_i(t)$  represents the average downloading speed of peers in class  $i$  at time  $t$ . The term  $x_i(t)d_i(t)/F$  specifies the rate at which leechers in class  $i$  turn into lazy-seeding seeders and  $y_i(t)/T_i$  specifies the leaving rate of lazy-seeding seeders in class  $i$ . In a steady state,  $dx_i(t)/dt = dy_i(t)/dt \equiv 0$ . Letting  $x_i$  and  $y_i$  represent the number of leechers and lazy-seeding seeders in class  $i$  in a steady state, and  $d_i$  represent the average downloading speed of leechers in class  $i$ , from Eq.(4.2) we have:

$$\lambda_i = \frac{y_i}{T_i} = \frac{d_i x_i}{F}. \quad (4.3)$$

Depending on the peers arrival rates ( $\lambda_i$ ) and the number of over-seeding peers ( $s_i$ ), a steady state will be either one of the two cases described separately below.

### Oversupplied

When there are a large number of seeders and a small number of leechers, and seeders cannot always fully utilize their upload capacities, we can characterize the swarm as *oversupplied*. Oversupplied is a typical phase for swarms in private BitTorrent communities. Given the abundance of seeders, it is realistic to assume that in an oversupplied



swarm seeders perform most of the uploads. Further considering the *piece availability problem*<sup>12</sup>, peers are more likely to download from seeders rather than other leechers. A previous measurement study [69] shows that in two private communities where the oversupplied situation exists, over 90% of the data comes from seeders. Accordingly in our model we assume that in an oversupplied steady state leechers do not contribute upload capacities. The condition for a swarm to be in an oversupplied steady state is:

$$\sum_i \lambda_i \Delta F < \sum_i (y_i + s_i) U_i \Delta = \sum_i (y_i + s_i) \gamma_i U_N \Delta, \quad (4.4)$$

which specifies that within a time interval  $\Delta$  the total upload volume that *can* be provided by all seeders is larger than the total download volume required by the  $\sum_i \lambda_i \Delta$  new leechers arriving in the same interval. Here  $U_i = \gamma_i U_N$  is the upload capacity of peers in class  $i$ .

In such a steady state, once a new peer joins, seeders upload to it with their full upload capacities and its download will be finished quickly. After that, seeders will be idle and wait for the next upload opportunity. Hence, on average seeders cannot fully utilize their upload capacities, and because of the operation of TCP<sup>13</sup> their average upload speeds will be proportional to their upload capacities. Let  $u_i$  represent the average upload speed of a peer in class  $i$ , then we have  $u_i = \gamma_i u_N$  and the total actual upload volume provided by all seeders should be equal to the total download volume required by all leechers, i.e.:

$$\sum_i \lambda_i \Delta F = \sum_i (y_i + s_i) \gamma_i u_N \Delta. \quad (4.5)$$

After the download is finished, a lazy-seeding peer in class  $i$  seeds for a period of length  $T_i$  until it achieves the SRE threshold, i.e., until  $\alpha = T_i u_i / F$ . Substituting  $y_i = \lambda_i T_i$  from Eq.(4.3) into Eq.(4.5) we find

$$u_N = \frac{(1 - \alpha) \sum_i \lambda_i F}{\sum_i s_i \gamma_i}. \quad (4.6)$$

Then,

$$T_i = \frac{\alpha F}{u_i} = \frac{\alpha F}{\gamma_i u_N} = \frac{\alpha \sum_i s_i \gamma_i}{(1 - \alpha) \gamma_i \sum_i \lambda_i}. \quad (4.7)$$

Given Eqs. (4.4) and (4.7), we can rephrase the condition for a swarm to be in an oversupplied steady state as:

<sup>12</sup>The piece availability problem specifies that two connected leechers may not perform actual download/upload, because they cannot find interesting pieces at each other.

<sup>13</sup>BitTorrent uses TCP as the transport layer protocol.

$$\frac{\sum_i s_i \gamma_i U_N}{\sum_i \lambda_i F} > 1 - \alpha. \quad (4.8)$$

In a private community where the SRE threshold equals  $\alpha$ , when  $s_i$ ,  $U_i$ ,  $\lambda_i$ , and  $F$  fulfill the condition of Eq.(4.8), we say that the swarm is in an oversupplied steady state. With a relatively small peer arrival rate ( $\lambda_i$ ) and a large number of over-seeding seeders ( $s_i$ ), peers will experience very low upload capacity utilizations and extremely long seeding times. This situation gets even worse for peers with low capacities: the ratio between the seeding times of two peers is inversely proportional to the ratio of their upload capacities ( $\gamma_i$ ). Moreover, over-seeding peers with higher upload capacities have a stronger influence (proportional to  $\gamma_i$ ) on this situation: the seeding time incurred by one over-seeding seeder in class  $i$  ( $i < N$ ) is equal to that incurred by  $\gamma_i$  ( $\gamma_i > 1$ ) over-seeding seeders in class  $N$ .

### Undersupplied

We recognize a swarm to be undersupplied if it is not oversupplied. We assume that in an undersupplied swarm both leechers and seeders can fully utilize their upload capacities, i.e.,  $u_i = U_i$ . This assumption has been validated by previous studies [48, 72]. In this situation, within a time interval  $\Delta$  the total upload volume that *can* be provided by all peers should be no larger than the total download volume required by all  $\sum_i \lambda_i \Delta$  new peers, i.e.:

$$\sum_i \lambda_i \Delta F \geq \sum_i (x_i + y_i + s_i) \gamma_i U_N \Delta. \quad (4.9)$$

Peers contribute their upload capacities, hence gain sharing ratios both in the leeching and the seeding process. At the end, they achieve sharing ratios equal to the SRE threshold, i.e.:

$$\alpha = \frac{((F/d_i)U_i + T_i U_i)}{F} \Rightarrow T_i = \frac{\alpha F}{U_i} - \frac{F}{d_i}, \quad (4.10)$$

where  $(F/d_i)U_i$  represents the upload volume provided by a leecher in class  $i$  in its leeching process.

With  $x_i$  leechers,  $y_i$  lazy-seeding seeders, and  $s_i$  over-seeding seeders in class  $i$  in a steady state, the average downloading speed of a peer in class  $i$  can be calculated by solving the system of equations proposed in Chapter 2:

$$d_i = \frac{(\sum_j \omega_{ji} x_j + \sum_j \sigma_{ji} (y_j + s_j)) U_j}{x_i}, \quad (4.11)$$

where  $\omega_{ji}$  ( $\sigma_{ji}$ ) specifies the fraction of upload speed allocated from a leecher (seeder) in class  $j$  to leechers in class  $i$  in the BitTorrent protocol.

From Eqs. (4.3), (4.10), and (4.11), we can derive  $T_i$ ,  $x_i$ , and  $y_i$  accordingly. Simply from Eq.(4.10) we already have  $T_i = \alpha F/U_i - F/d_i < F/U_i$ . This implies that in an undersupplied steady state, an upper bound for the seeding time of a peer is the ratio between the size of the shared file and its upload capacity, which is much better than in an oversupplied steady state (Eq.(4.7)).

Further applying Eqs. (4.10) and (4.11) to Eq.(4.9), we rephrase the condition for a swarm to be in an undersupplied steady state as follows:

$$\frac{\sum_i s_i \gamma_i U_N}{\sum_i \lambda_i F} \leq 1 - \alpha, \quad (4.12)$$

which is exactly the reverse of Eq.(4.8).

## 4.4.2 The effects of SRE

In this section we show the user-level performance under SRE. Based on simulations we examine the influence of several parameters and we exhibit the main reasons for the positive and negative effects of SRE.

### Experimental setup

In Section 4.3.1 we have shown that in closed private communities crashes or crunches easily happen. It is not worthwhile to analyze the user-level performance in an unsustainable system. Hence, in this section we consider an open system with peer arrivals. As stated in Section 4.2.1 and 4.3.1, new peers bring credit into the system and the increase of the credit level alleviates the potential systemic risk. In reality, there are many private communities with open registration, e.g., BitSoup [10], and they can be considered as open systems.

We use the same simulator and consider the same initial settings as in Section 4.2, except that now we consider 100 initial peers and 5 swarms in the system. In each cycle, new peers arrive according to a certain arrival rate and they join a random swarm to download. After the first download, they maintain a sharing ratio above their target thresholds. Each peer (with upload capacity 1 unit per cycle) attempts to download all the 5 files (with size of 10 units) in the system, in random order. We consider a bandwidth-homogeneous BitTorrent system unless otherwise indicated. We run the simulation for 2000 cycles and keep a record of peers who finish downloading all the files by the end of the simulation. The results represent the average of 5 runs.

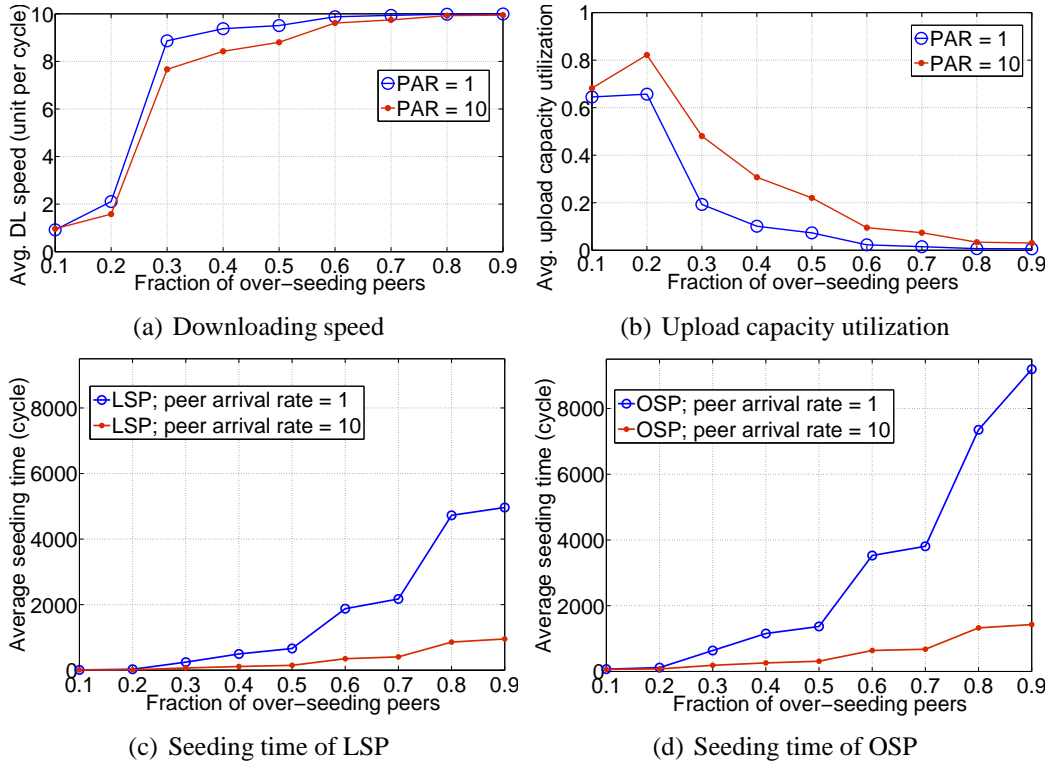


Figure 4.6: User-level performance under different fractions of lazy-seeding peers (LSP) and over-seeding peers (OSP), and different peer arrival rates.

### The imbalance of bandwidth supply and demand

In our first experiment we vary the fraction of over-seeding peers, thus generating different levels of oversupply. As shown in Fig. 4.6(a), with the fraction of over-seeding peers increasing from 0.1 to 0.9, the average downloading speed is increased nearly 10 times. However, the average upload capacity utilization is significantly deteriorated and the seeding time is increased dramatically. With 50% over-seeding peers, on average each peer can only utilize less than 20% of its upload capacity (Fig. 4.6(b)). With this low upload capacity utilization, all peers have to stay for extremely long times (compared to their downloading times) to achieve the sharing ratio required by SRE (Figs. 4.6(c) and 4.6(d)). In our experiment with 50% over-seeding peers, the seeding time of a lazy-seeding peer is nearly 200 times more than its downloading time, and for over-seeding peers, it even increases to over 400 times.

On the other hand, with a smaller peer arrival rate (which means a smaller demand) the imbalance and hence the performance, are even worse. As shown in Fig. 4.6, when the peer arrival rate decreases from 10 to 1 peer per cycle, with the same fraction of over-seeding peers, the average upload capacity utilization is decreased 2-3 times and the average seeding time is increased 2-5 times.

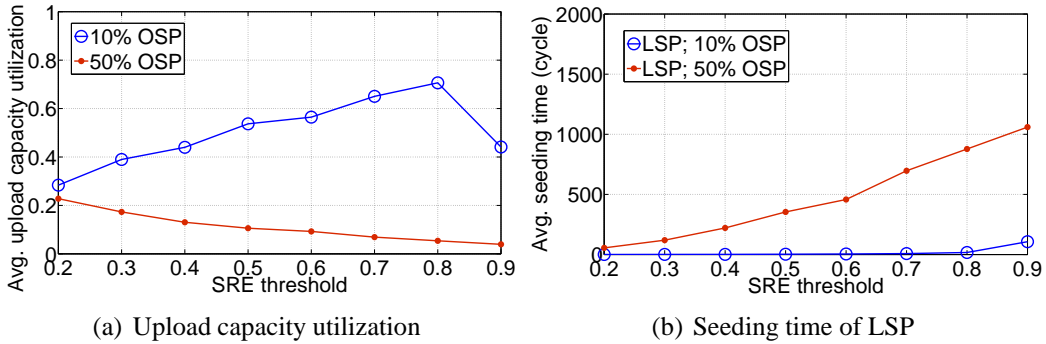


Figure 4.7: Influence of the SRE threshold under different fractions of lazy-seeding peers (LSP) and over-seeding peers (OSP).

Our simulation results show that, under SRE, the existence of over-seeding peers makes the swarms oversupplied. As a consequence, with a relatively large fraction of over-seeding peers and a small peer arrival rate, peers have to seed for extremely long times, though their seedings are not very productive. This is consistent with the theoretical model proposed in Section 4.4.1.

### The influence of the SRE threshold

Many communities [10, 40] use 0.7 as the default value of the SRE threshold, empirically or intuitively. This section complements the necessary analysis behind the choice.

Fig. 4.7 shows that, which is consistent with our intuition, when the SRE threshold is increased from 0.2 to 0.9, the upload capacity utilization decreases while the average seeding time is increases. Further, the effect SRE is limited when the fraction of over-seeding peers is small. Surprisingly, in Fig. 4.7(a) we see that when there are 10% over-seeding peers, the upload capacity utilization is increased when the SRE threshold increases from 0.2 to 0.8, and then drops when it further increases to 0.9. We believe this is due to, what we term as, the *seeder's dilemma*: with either a very small or a very large number of seeders, peers cannot well-utilize their upload capacities. The former case is due to the *piece availability problem*: When there are not enough seeders, leechers have to exchange data with each other, which is not always possible since they only hold a part of the entire file. The latter case is due to the insufficient download demand. Without enough demand, though seeders have the will, they cannot find enough leechers to upload to.

### The discrimination against peers with limited capacities

In this subsection, we analyze SRE's effects in bandwidth-heterogeneous systems. More specifically, we simulate a system with two classes of peers, namely slow and fast peers.

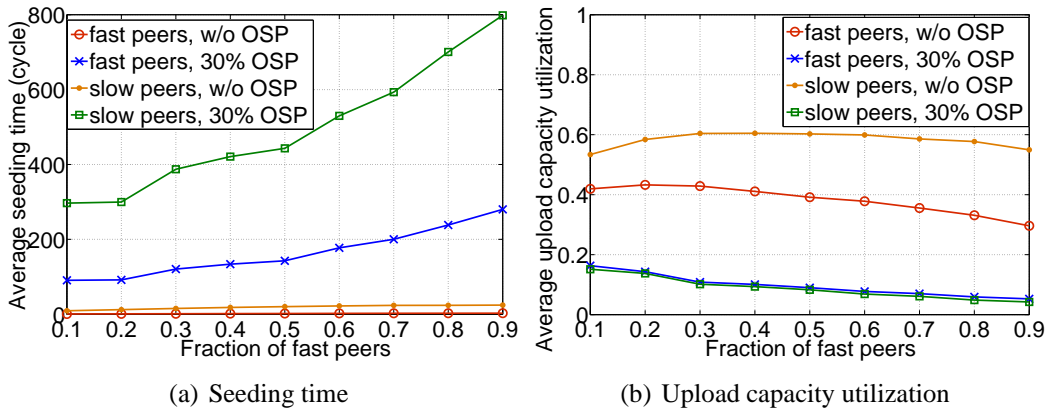


Figure 4.8: SRE's discrimination under different fractions of over-seeding peers (OSP).

All the other settings are the same as in previous experiments, except that the upload capacity of slow peers is 1 unit per cycle and for fast peers it is 4 units per cycle. We consider two scenarios in our simulation, i.e., without and with 30% over-seeding peers. As we show previously, 30% over-seeding peers is typical to demonstrate the effects of SRE. We change the fraction of fast peers from 0.1 to 0.9 and the results are shown in Fig. 4.8.

We see that when there is no over-seeding peers fast peers barely need to do any seeding work, but their existence increases the seeding times of slow peers (Fig. 4.8(a)). This result is consistent with the result derived from the theoretical model proposed in Chapter 3, where we show that high-capacity peers manage to upload considerably more during the leeching process, and thus need to seed for shorter times. When the fraction of over-seeding peers is increased from 0 to 30%, slow peers need to seed 200 to 500 cycles more than fast peers, while originally they only needed to seed 20 cycles more. In general, slow peers need to seed 4 times as long as fast peers, which is the same as the ratio between the upload capacity of a fast and a slow peer. This result is also consistent with our previous theoretical results in Section 4.4.1.

Meanwhile, Fig. 4.8(b) shows that the upload capacity utilizations of both fast and slow peers do not change much with the fraction of fast peers. However, when there is no over-seeding peer, slow peers have better upload capacity utilizations. We believe this is due to the fact that slow peers stay as seeders longer than fast peers. Normally seeders can achieve better upload capacity utilizations, since they are not influenced by the piece availability problem.

While fast and slow peers both put all their effort in participating in the community, slow peers need to seed longer. We term this as SRE's *discrimination* against low-capacity peers. Clearly, the long seeding time, the low upload capacity utilization, and the discrimination against low-capacity peers severely deteriorate the user-level performance in private communities. In the following sections, we propose several strategies to alleviate

these problems.

### 4.4.3 Description of proposed strategies

Inspired by ideas in social sciences and economics, in this section we propose four strategies aimed at alleviating the negative effects of incentive policies used in private BitTorrent communities, which require only a minor revision of those policies.

#### Negative taxation

The idea of *negative taxation* is that people earning below a certain amount receive supplemental pay from the government [32]. We take inspiration from the concept of negative taxation and devise a new strategy in which the upload amount of a peer is calculated as its actual upload amount multiplied by coefficient  $\mathcal{T}$  defined as:

$$\mathcal{T} = \max\{\min\{1/SR, \theta\}, 1\},$$

where  $SR$  represents the sharing ratio of a peer and  $\theta > 1$  represents the *maximum negative taxation degree*.

It is easy to see that a) when  $SR \geq 1$ ,  $\mathcal{T} = 1$ , b) when  $1/\theta \leq SR < 1$ ,  $\mathcal{T} = 1/SR > 1$ , and c) when  $SR \leq 1/\theta$ ,  $\mathcal{T} = \theta > 1$ . By using this new strategy, to gain the same sharing ratio, poor peers ( $SR < 1$ ) seed less and rich peers ( $SR \geq 1$ ) seed the same amount as when using the original SRE. The maximum negative taxation degree controls the maximum negative taxation a peer can get, which alleviates the threat of free-riding.

#### Welfare for the rich

The term *welfare for the rich* is used to describe the bestowal of grants and tax-breaks to the wealthy [63]. Taking inspiration from this concept, we devise another strategy to alleviate the long seeding time, i.e., accelerating the seeding process of an over-seeding peer by giving welfare to it. The upload amount of a peer is calculated as its actual upload amount multiplied by coefficient  $\mathcal{W}$  defined as:

$$\mathcal{W} = \max\{\min\{SR, \varphi\}, 1\},$$

where  $\varphi > 1$  represents the *maximum welfare degree*.

By using this strategy, to gain the same sharing ratio, poor peers ( $SR < 1$ ) seed the same amount and rich peers ( $SR \geq 1$ ) seed less than when using the original SRE. The maximum welfare degree controls the maximum welfare a peer can get, to prevent the over-seeding seeders from achieving their desired sharing ratios too quickly.

Community administrators can choose different values for  $\theta$  and  $\varphi$  to have different maximum negative taxation degrees and the maximum welfare degrees. In our simulation we choose  $\theta = \varphi = 2$ .

### **Remuneration according to effort**

In participatory economics, the *maxim of remuneration according to effort* has been introduced [3]. Under this scheme, people are paid according to the effort they put in rather than the amount of contribution. Taking inspiration from this concept, we propose the third strategy which takes into account the effort of users in terms of their seeding times. Previous studies have shown that the effort-based incentive policy applied in the leeching process improves the system-wide performance [90]. We expect the same improvement when this effort-based methodology is applied in a private community.

More specifically, by applying SRE with *counting seeding time*, a peer can start a new download when either it has achieved the SRE threshold or it has seeded for a sufficiently long time. In this way, peers that are stuck in long seeding process in oversupplied swarms can leave and perform further downloads. The new demand generated by these peers helps to balance the bandwidth demand and supply in the system.

Clearly, the definition of “a sufficiently long period” is quite vague. Community administrators may choose various values, like 4 hours, 10 hours, or one day. In our simulations, we simply assume that it equals the size of the shared file divided by the upload capacity of a peer. Note that since over-seeding peers are deposit-oriented, they still start new downloads only when they have achieved their desired sharing ratios.

### **Supply-based price**

According to the law of supply and demand, if the demand remains constant and the supply increases, the price of an item decreases and vice versa. For an insightful discussion of the relationship between supply, demand and price, we refer the reader to [7]. We take inspiration from this insight to devise our fourth strategy, i.e., SRE with *supply-based price*. The basic idea is that the price a downloader needs to pay for downloading one unit of data should be inversely correlated with the supply in the swarm, i.e., the higher the seeder-to-leecher ratio, the less a downloader should pay and vice versa. In this way, in an oversupplied swarm, a leecher pays less and potentially achieves a higher sharing ratio by the end of its leeching process. Hence it is less likely for it to have an insufficient sharing ratio and thus stay as a seeder, which indirectly solves the oversupply problem in this swarm. On the other hand, in an undersupplied swarm, a leecher pays more and potentially achieves a smaller sharing ratio, which makes it stay as a seeder with a higher possibility than using the original SRE. In this way, the undersupply problem is also alleviated indirectly.



Here, we use the *seeder-to-leecher ratio* ( $SLR$ ) as a metric to decide whether a swarm is oversupplied or undersupplied. Community administrators can set different  $SLR$  values as the threshold, but we simply assume that when  $SLR \geq 1$  the swarm is oversupplied and when  $SLR < 1$  the swarm is undersupplied. The download amount of a peer is calculated as its actual download amount multiplied by coefficient  $\mathcal{P}$  defined as:

$$\mathcal{P} = \max\{1/SLR, \phi\},$$

where  $\phi$  represents the *lowest price* for downloading one unit of data, which is used to alleviate the threat of free-riders. Community administrators can choose different values for the lowest prices. In our simulation we choose  $\phi = 0.1$ .

Note that the free-leech strategy we proposed in Section 4.3.3 to solve credit crunch is an extreme case of SRE with supply-based price, with price equal to 0.

#### 4.4.4 Strategy evaluation

In this section we evaluate the performance of the new strategies proposed in Section 4.4.3. The experimental setup is the same as in Section 4.4.2 and results are shown in Figs. 4.9 and 4.10.

##### Higher upload capacity utilization and shorter seeding time

From Fig. 4.9 we see that by using any of the new strategies, peers achieve higher upload capacity utilizations, as well as smaller seeding times. As shown in Fig. 4.9(a), when there are 40% over-seeding peers the upload capacity utilization is increased 2-3 times compared to using the original SRE. While all other strategies have decreasing upload capacity utilizations with an increasing fraction of over-seeding peers, SRE with supply-based price performs stably. Given any fraction of over-seeding peers, on average peers can utilize at least 40% of their total upload capacities while for the original SRE it drops to less than 1% when there are 90% over-seeding peers.

With the improved upload capacity utilization, the average seeding time is reduced significantly. As shown in Figs. 4.9(b) and 4.9(c), when there are 60% over-seeding peers, SRE with welfare for the rich reduces at least 10% of the original seeding time for both lazy-seeding and over-seeding peers. SRE with negative taxation deals with lazy-seeding peers directly, hence it achieves an even better performance in reducing the seeding time of lazy-seeding peers, which is a 50% improvement compared to that achieved by SRE with welfare for the rich.

SRE with counting seeding time further relieves lazy-seeding peers from the long seeding process in a more effective manner. As shown in Fig. 4.9(b), they only need to seed for a negligible time compared to when using the original SRE, or either of the

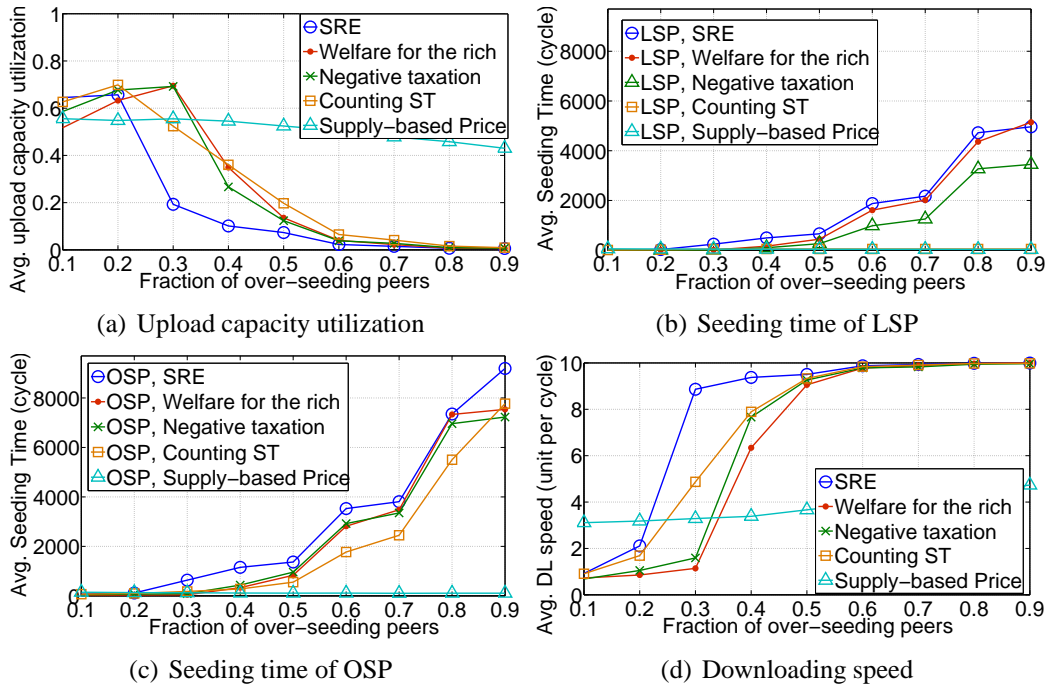


Figure 4.9: Strategy performance in alleviating the eternal seeding problem under different fractions of lazy-seeding peers (LSP) and over-seeding peers (OSP).

above two new strategies. Interestingly, by applying SRE with counting seeding time, the seeding time of over-seeding peers is also decreased (Fig. 4.9(c)), even though they still desire the high sharing ratios as when using the original SRE. We believe this is due to the fact that with lazy-seeding peers finishing their seedings sooner, the upload competition is reduced and over-seeding peers can achieve their desired threshold more quickly. Meanwhile, when the lazy-seeding peers are released from the seeding process, they join other swarms as new leechers, which indirectly alleviates the oversupply in those swarms.

Finally, the best performance in reducing the seeding time for all peers is achieved by SRE with supply-based price. The seeding time of both lazy-seeding and over-seeding peers is reduced by three orders of magnitude. In our view the main reason for the success of SRE with supply-based price is that it adaptively adjusts the supply and demand in a swarm. When the swarm is oversupplied, the price for downloading one unit of data is lower and peers can finish downloads at less expense, which directly reduces their consequent seeding amount and hence avoids adding more seeders in this oversupplied swarm. In this way, the imbalance of bandwidth supply and demand is mitigated, and the strategy gives a way to escape out of the seeder's dilemma as described in Section . A similar argument can also be applied to an undersupplied swarm.

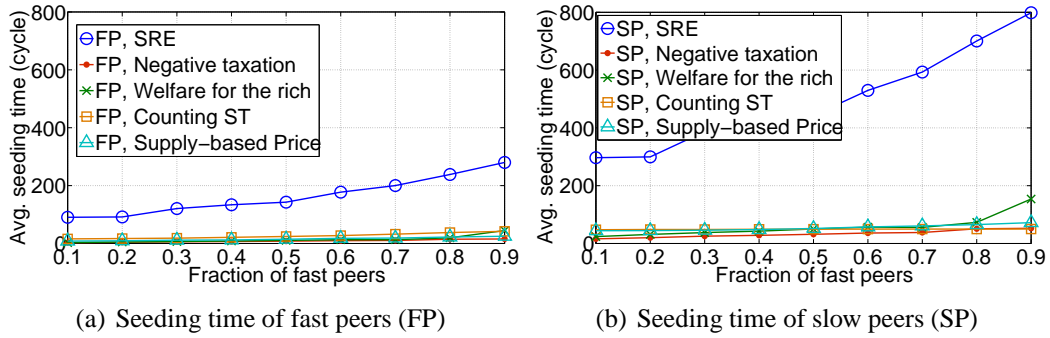


Figure 4.10: Strategy performance in alleviating discrimination with 30% over-seeding peers (OSP).

### Tradeoff: slightly decreased downloading speed

By adopting any of the new strategies, while the seeding time is dramatically reduced, as a trade-off, the average downloading speed is decreased (Fig. 4.9(d)), hence the downloading time is increased. However, given that in our simulations we consider files with size equal to 10 units, the increase of the downloading time (tens of cycles) is negligible compared to the decrease of the seeding time (hundreds or even thousands of cycles).

### Reduced discrimination

To examine the effectiveness of the proposed strategies in alleviating SRE's discrimination against peers with limited capacities, we repeat our experiments by further considering a bandwidth-heterogeneous system with two classes of peers, fast and slow. From Fig. 4.10 we see that *all* the proposed strategies effectively alleviate SRE's discrimination against low-capacity peers. With 30% over-seeding peers, originally slow peers need to seed 200-500 cycles more than fast peers do. By applying any of the new strategies, this difference is reduced to within tens of cycles.

## 4.5 Dynamic file popularity

So far, we have only considered scenarios in which all files have the same constant popularity. However, many measurement studies [5, 57] show that the popularity of a file decreases quickly after it is first published. In this section, we analyze the effects of SRE and evaluate our proposed strategies under dynamic file popularity.

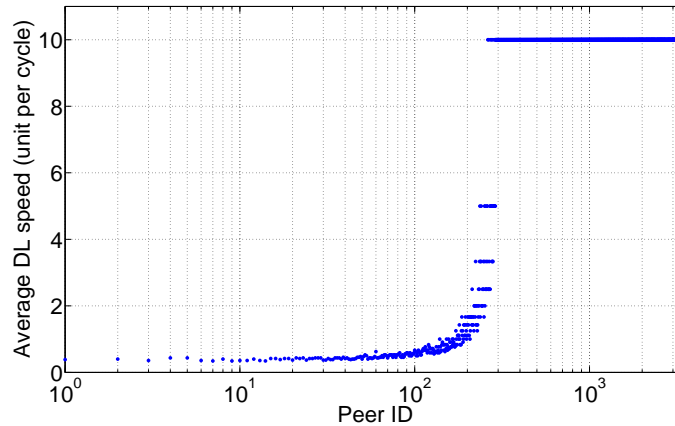


Figure 4.11: Individual average downloading speed.

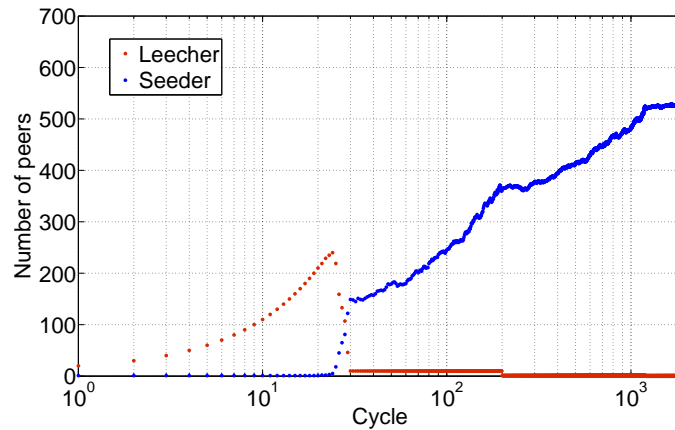


Figure 4.12: The number of peers in the system.

### 4.5.1 Experimental setup

We use the same simulator and consider the same initial settings as in Section 4.2, except that to better abstract the effects of dynamic file popularity, we only consider one swarm with decreasing popularity. The simulation starts with one injector, who stays in the swarm as a permanent seeder. In successive cycles, new peers arrive according to an exponentially decreasing arrival rate ( $\lambda(t) = \lambda_0 e^{-\frac{t}{\tau}}$ ), a peer arrival pattern that has been observed in many BitTorrent swarms [85]. Each peer joins the swarm with zero upload and download amounts. After a peer finishes its download, it seeds, if necessary, until it achieves its target threshold.

By default, we set 30% peers to be over-seeding. As shown in Section 4.4.2, this percentage is typical for showing the effects of SRE. We choose  $\lambda_0 = 10$  and  $\tau = 300$

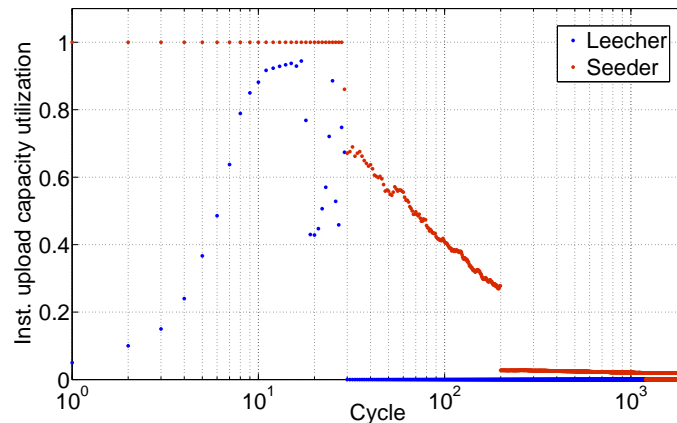


Figure 4.13: The average upload capacity utilization.

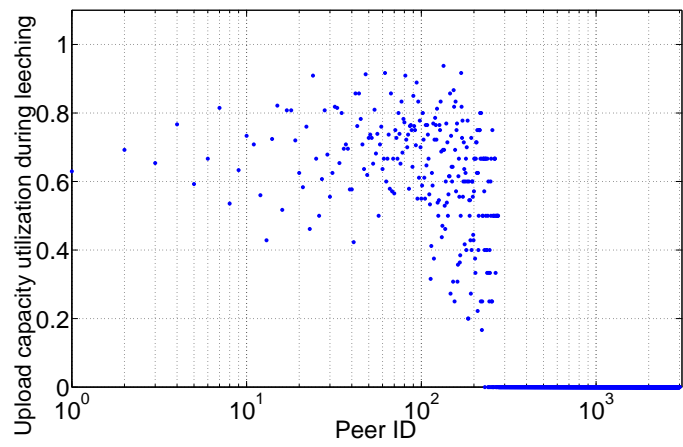
and we run the simulation for 2000 cycles. All together 3000 peers are included. We have also tested different values for  $\lambda_0$  and  $\tau$ , which give very similar results.

#### 4.5.2 The effects of SRE under dynamic file popularity

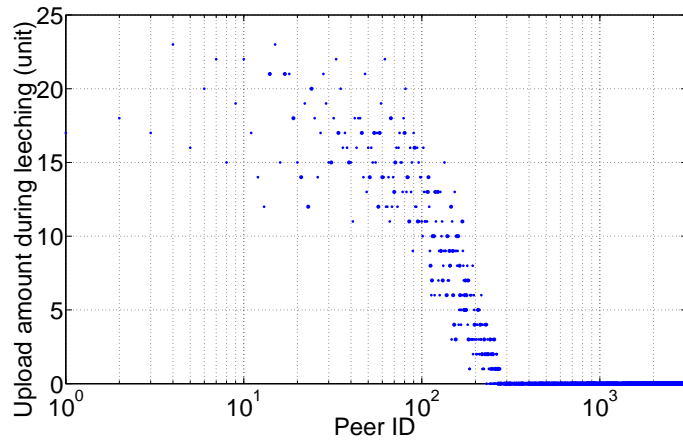
We first demonstrate the effects of SRE under dynamic file popularity. Fig. 4.11 shows each peer's average download speed, where a smaller peer ID means an earlier arrival time. We see that the first 200 peers experience download speeds similar to their upload capacities (1 unit per cycle). From peer 200, the download speed increases quickly: with file size equal to 10 units, it soon reaches the maximum, i.e., 10 units per cycle. This means that new peers can finish their downloads within the same cycle that they join the swarm.

The high downloading speed is due to the oversupply. As shown in Fig. 4.12, the number of seeders increases quickly and after the first 30 cycles, the swarm is occupied with hundreds of seeders but only with very few leechers. The presence of existing seeders increases the difficulty for a new seeder to achieve its target threshold and leave the swarm, and vice versa. We term this as *cumulative seeder effect*. As a consequence, the upload capacity utilization decreases severely. As shown in Fig. 4.13, within the first 500 cycles, both seeder's and leecher's upload capacity utilization decrease to less than 5%, with seeders performing a little bit better than leechers as they do not face the piece availability problem.

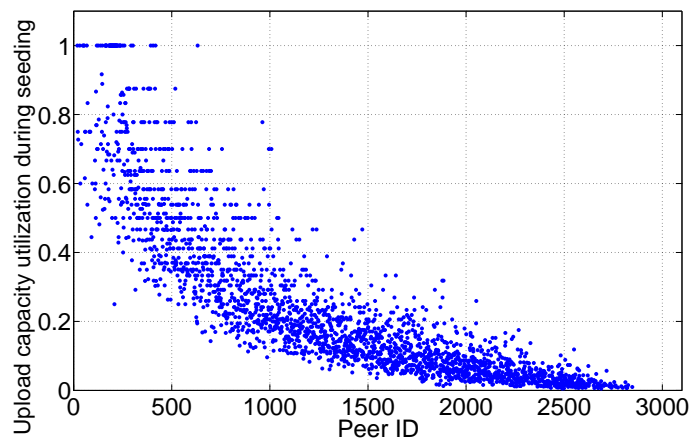
With the above differences in instantaneous system performance, peers arriving at different times achieve markedly different performance. As shown in Fig. 4.14, arriving earlier means higher upload speeds and hence larger upload amount during the leeching process (Figs. 4.14(a) and 4.14(b)), as well as better upload capacity utilization during



(a) Individual upload capacity utilization during leeching.



(b) Individual upload amount during leeching.



(c) Individual upload capacity utilization during seeding.

Figure 4.14: Individual upload activity.

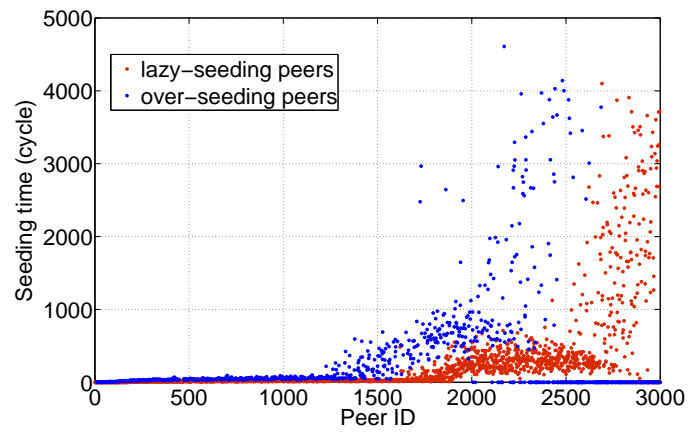
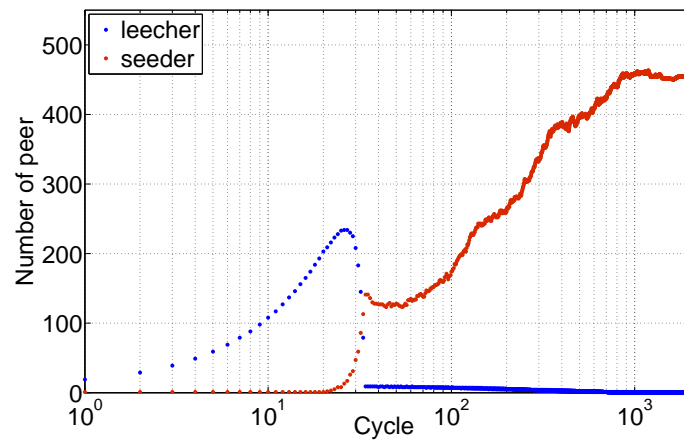
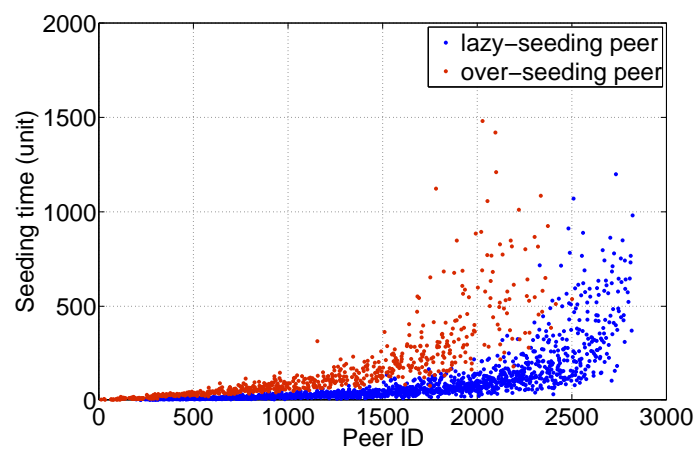


Figure 4.15: Individual seeding time.



(a) Instantaneous number of peers



(b) Individual seeding time

Figure 4.16: SRE with negative taxation under dynamic file popularity.

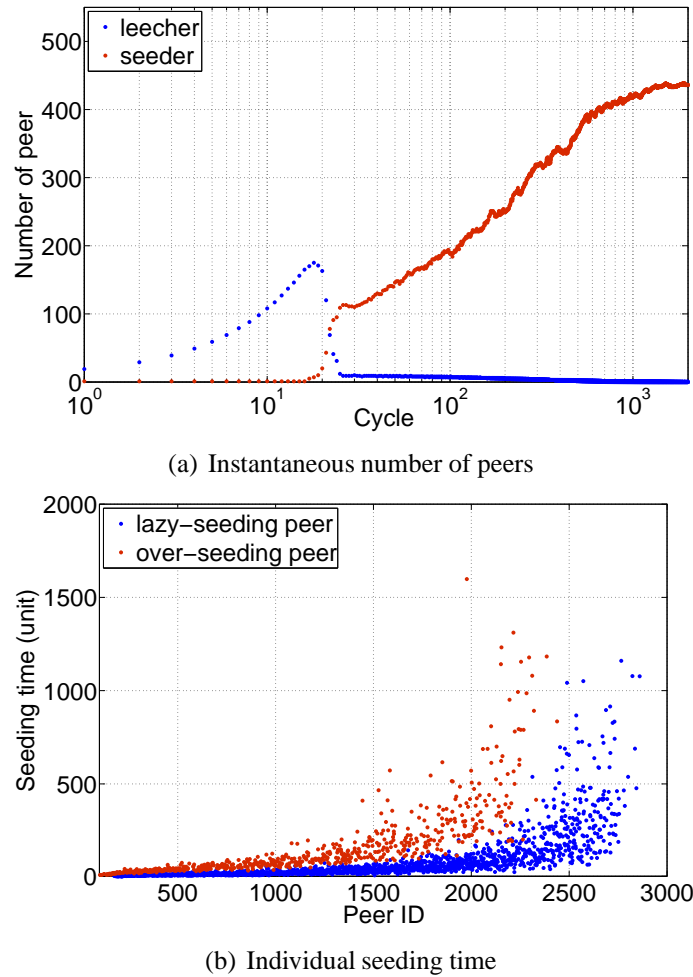


Figure 4.17: SRE with welfare for the rich under dynamic file popularity.

the seeding process (Fig. 4.14(c)). Thus, peers that arrive earlier experience much smaller seeding times. As shown in Fig. 4.15, to achieve the target thresholds, the first 500 peers only need to seed tens of cycles. After this, the seeding time increases quickly to hundreds or even thousands of cycles.

### 4.5.3 Proposed strategies under dynamic file popularity

We evaluate the performance of our proposed strategies under dynamic file popularity. The results are shown in Figs. 4.16, 4.17, 4.18, and 4.19.

1) *SRE with negative taxation* and 2) *SRE with welfare for the rich: limited effect for very low file popularity*: As in swarms with constant file popularity, applying SRE with negative taxation or SRE with welfare for the rich alleviates the oversupply. Comparing Figs. 4.12, 4.16(a), and 4.17(a), we see that these two new strategies reduce the instan-



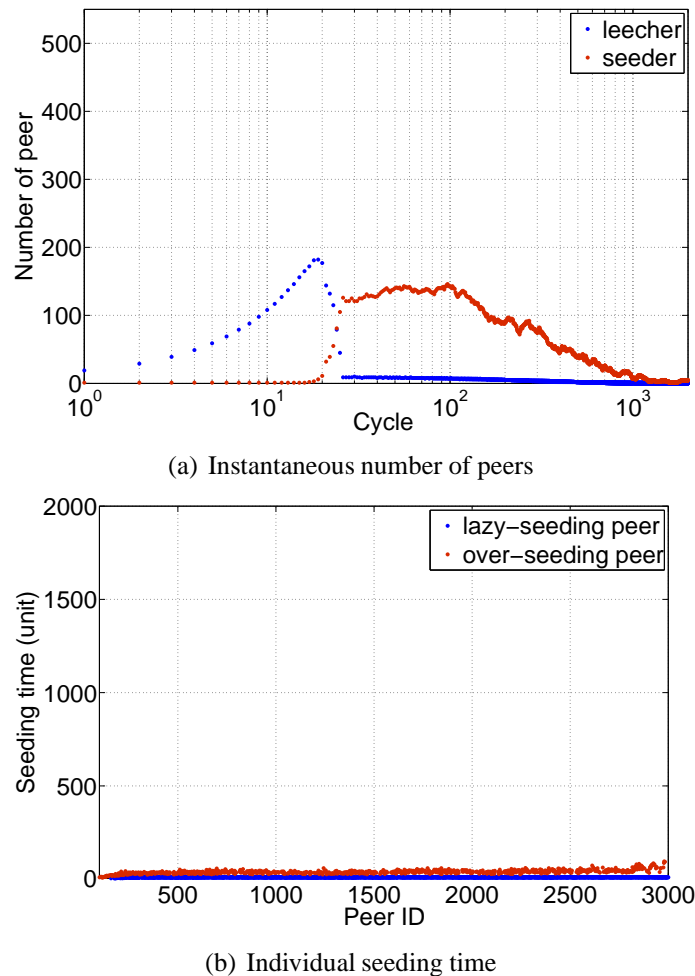
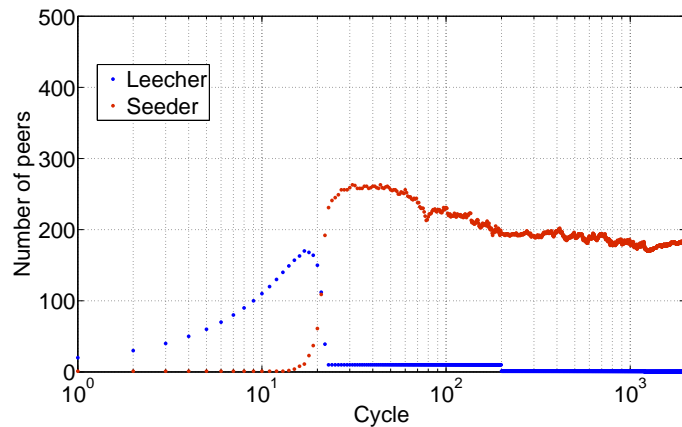


Figure 4.18: SRE with counting seeding time under dynamic file popularity.

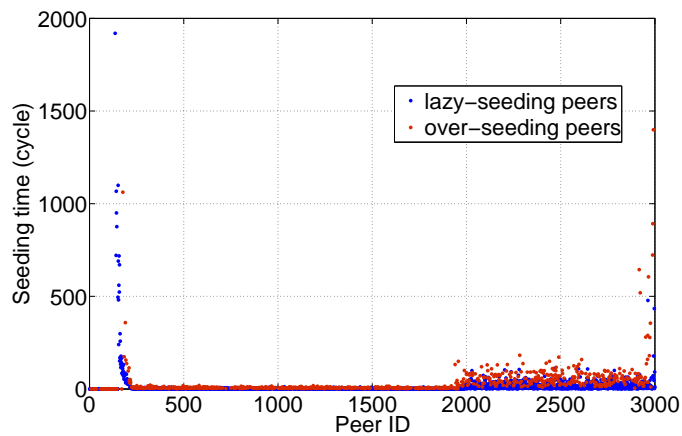
taneous number of seeders to around 80% of the case when the original SRE is adopted. But their effects are limited with very low file popularity. As shown in Figs. 4.16(b) and 4.17(b), peers that arrive late still experience relatively long seeding times.

3) *SRE with counting seeding time: strong effect in reducing the seeding time:* With the hard SRE requirement replaced by seeding for a particular period, SRE with counting seeding time dramatically alleviates the oversupply in the swarm. As shown in Fig. 4.18(a), except for the large number of seeders during the first 200 cycles, the instantaneous number of seeders is almost always under 100. Among these, we conjecture that most are over-seeding peers, since lazy-seeding peers can leave the swarm once they've seeded for a relatively short time, i.e., 10 cycles in our experiment. While these peers are released from the endless seeding process, the seeding time of over-seeding peers is also dramatically decreased to less than 100 cycles.

4) *SRE with supply-based price: effectively stabilize the supply:* Among all the four



(a) Instantaneous number of peers



(b) Individual seeding time

Figure 4.19: SRE with supply-based price under dynamic file popularity.

new strategies, SRE with supply-based price stabilizes the supply most effectively. As shown in Fig. 4.19(a), the instantaneous number of seeders stays stable after 200 cycles. With this constant supply (and not oversupply), peers experience relatively small seeding times (Fig. 4.19(b)). When the number of new peers decreases, as a matter of course, peers experience longer seeding times, but still much smaller compared to adopting the original SRE. We believe this constant supply and small seeding time are due to the fact that SRE with supply-based price is self-organized, and will adjust the demand and supply automatically.

It should also be noted that when adopting SRE with supply-based price, the first 200 peers have relatively longer seeding times than peers arriving later. We believe this is due to the fact that those peers arrive during the phase that the swarm is occupied with a large number of leechers and a small number of seeders. Hence, the price for downloading is higher and peers need to pay more to achieve their target thresholds.

## 4.6 Discussion: The change of user behavior?

It could be argued that the new strategies proposed in this chapter may trigger a change in user behavior. Specifically, users from the two classes that we defined, lazy-seeding and over-seeding, might be incentivized to switch their classes under the new strategies. It could be conjectured that as a consequence, the system performance might be adversely affected.

However, we note that there is only a very small fraction of strategic users in BitTorrent communities [4]. So the likelihood of peers changing behavior is quite small. Especially regarding over-seeding peers, we argue that in general the possibility of such peers changing to lazy-seeding peers is quite low. This is because we conjecture that the behavior of over-seeding peers is motivated by either one or a combination of the following three reasons: a) Over-seeding peers always want to be relatively more well-off in terms of sharing ratio as compared to average users so that they can be among the “rich elite” of the community and gain some potential benefits<sup>14</sup>; b) They are altruists who want to help the community as much as they possibly can; and c) They are hoarders who desire to conserve sharing ratio for “rainy days” i.e., those time periods when they feel they might engage in heavy downloading activity and might as a result be expelled from the community due to low sharing ratios.

Nevertheless, in this section we would like to analyze what happens if users do change their behavior. We consider each proposed strategy in turn and discuss the possible effects of the change in user behavior.

### **SRE with negative taxation and welfare for the rich:**

Under SRE with negative taxation, peers with lower sharing ratios gain sharing ratio more easily. Hence, lazy-seeding peers have no incentive to change their behaviors, while over-seeding peers may change to lazy-seeding peers. Similarly, when SRE with welfare for the rich is applied, strategic lazy-seeding peers could become over-seeding peers, while over-seeding peers will not change their behaviors.

The worst case scenario of applying either of these two new strategies is that all peers exhibit the same behavior, i.e., either all peers become lazy-seeding or all become over-seeding. In the former case every member of the community would still be forced to maintain a minimum sharing ratio required by SRE, i.e., every peer would continue to provide a certain level of contribution. This outcome would still be better than the situation in a public community where every peer has the option to leave immediately after downloading. In the latter case, i.e., when all peers are over-seeding, it is less likely for them to complain, since an over-seeding behavior, i.e., a desire for higher sharing ratios,

---

<sup>14</sup>Such as priority in downloading popular files, the possibility to send invitations to others, etc [10,40].

automatically implies a long seeding time.

#### **SRE with counting seeding time:**

Under SRE with counting seeding time, users may set their upload speeds to zero and pretend to be seeding. However, according to the TFT policy in BitTorrent, in an under-supplied swarm the upload speed of a peer directly influences its downloading speed. If a user sets its upload speed to zero, it would hardly be able to download at reasonable speeds in undersupplied swarms, which are normally swarms providing new and popular content.

Further, private community administrators could set another SRE threshold, which is smaller than the original one, and stipulate that users who cannot achieve the original SRE threshold must seed for a predefined time, as well as achieve the smaller SRE threshold. In this case, the potential threat of free-riders is alleviated.

#### **SRE with supply-based price:**

By adopting SRE with swarm-based price, the only advantage that users could gain is that they may opt to download files which have a lower price. However, this is unlikely to happen, since the choice of file to download mainly depends on user's interest in the content, rather than the price. Even if some users might be tempted to download a file simply based on its price, this would have little influence on the performance of SRE with swarm-based price, because this strategy is self-organizing and will adjust the balance of supply and demand automatically.

## **4.7 Related work**

This chapter is based on two previous papers [49, 89] with extensions including demonstrating detailed measurement results, unifying the analysis of SRE-based and credit-based private communities, analyzing the systemic risk of SRE-based private communities, as well as the influence of swarm popularity.

Many P2P incentive schemes based on credits have been proposed in the literature. Vishnumurthy *et al.* [107] present a system involving a virtual currency called *Karma*, which is defined as the value capturing the amount of resources a peer has contributed and consumed. The level of Karma (or credit) in the system is maintained and measures are taken to avoid inflation and deflation that can occur when peers leave the system. However, in avoiding inflation and deflation, the only aim of the paper is to maintain the per-capita Karma, i.e., the total Karma divided by the number of active users.

Kash *et al.* [56] show that in a scrip system, where agents can consume and produce services, both an overabundance of money supply and its shortage lead to inefficiency.

They also consider hoarders and how to optimize the credit supply. Our work is different in that we focus not on a generic service exchange scenario but on a file sharing scenario inspired by BitTorrent private communities. Also we apply multiple user behaviors rather than focusing only on one that optimizes the utility. In addition we focus on detecting and avoiding extreme crashes and crunches, where the entire system seizes. Last but not the least, we also study the effects of such credit-based schemes from the perspective of user level performance.

As stated, for grounding our work we chose the realm of private P2P file sharing communities. To date, only few works have analyzed private communities. Zhang *et al.* [124] investigate hundreds of private trackers and depict a broad and clear picture of the private community landscape. Chen *et al.* [13] compare system behaviors among 13 private trackers and 2 public trackers, and they show their differences regarding user viscosity, single torrent evolution, user behaviors, and content distribution. Liu *et al.* [64] also perform measurement studies and further develop a model to show that SRE indeed provides effective incentives, but is vulnerable to collusion.

While these studies all focus on demonstrating the high seeding level achieved by private communities, there have been a few preliminary works that show the adverse effects. Andrade *et al.* [5] focus on the dynamics of resource demand and supply, and they show that users typically try to increase their contribution levels by seeding for longer and not by providing more bandwidth to the system. However, our paper shows that providing limited bandwidth is not the will of users, but it is a consequence of the oversupply in private communities. Chen *et al.* [14] also notice the oversupply problem and provide a model to identify the optimal stable SLR range. However, they didn't propose strategies to solve the problem of oversupply. Kash *et al.* [57] demonstrate that there are significant disparities in the cost of new and old files in a private community named DIME, and users compensate for the high cost of older files by downloading more copies of newer files or by preferentially consuming older files during freeleech periods. Particularly, they have shown that after a period of freeleech, there are more download activities in the community. This is consistent with our theoretical result that during a pre-crunch state, injecting credit will increase the system throughput. While these papers mainly perform measurement-based studies to analyze the positive and adverse effects of SRE-like schemes on user-level performance, our paper is based on measurement, theoretical model, as well as extensive simulations. Further we propose new strategies to alleviate SRE's punishment, which are evaluated to be very effective through simulations.

## 4.8 Conclusion

In this chapter we have studied the effects of credit-based and SRE-based incentive policies employed in private P2P communities, from both the system-level and the user-level

performance perspective.

Based on two user behaviors abstracted from real world observations, i.e., lazy-seeding and over-seeding, we examine the system-level credit dynamics and show that crunches and crashes can easily happen in private communities. Crunches and crashes are due to credit shortage and credit abundance, respectively, and we apply a theoretical analysis to characterize the conditions that lead to these extreme outcomes. We apply the derived conditions to implement a novel *adaptive credit intervention mechanism* that proactively stops the system from seizing by temporarily changing the credit policies. A system that is predicted to crunch allows freeleech, and conversely, a system that is predicted to crash imposes freeseed. Simulation results show that our mechanism is very effective in avoiding crunches and crashes.

Given a private community that is sustainable, we further analyze its user-level performance by analyzing the positive and negative effects of SRE. Our simulation results show that with the existence of over-seeding peers, by adopting SRE, swarms tend to be extremely oversupplied. Although achieving an increase in the average downloading speed, the oversupply induces undesired effects, including low upload capacity utilizations, extremely long seeding times, and an unfair playing field for late entrants into swarms. To alleviate these problems, we propose four strategies and the simulation results show that they are all very effective. Particularly, *SRE with supply-based price*, while maintaining a system-wide high downloading speed, achieves very stable high upload capacity utilization and reduces seeding durations by three orders of magnitude as compared to the original SRE. When then the adaptive intervention mechanism is run in the background to check the extreme conditions for crunches and crashes, the system is ensured to have a high and sustainable performance.

## Chapter 5

# User behaviors under contribution incentives: a measurement study

The primary goal for adopting contribution incentives in online networks is to trigger user dedication, and they often work. Take private BitTorrent communities for example: it has been demonstrated that their users are more dedicated than users in public BitTorrent communities where credit-based or SRE-like incentives are not adopted [13, 64, 69, 124]. Nevertheless, one may wonder, besides the universal dedication, are there other types of user behavior triggered by contribution incentives? And besides an increased supply of contributions, are there any other positive or even negative effects of these user behaviors?

To answer these questions, in this chapter, we take private communities as an example to explore the user behaviors under contribution incentives. Private communities are ideal case studies, for they maintain user-level identifications and they keep a detailed tracking of user activities that can be obtained by crawling their sites. We perform a measurement study on three private communities. We classify their users into different groups based on their sharing ratios, their ages, their levels of consumption, and their *effort ratios*. The effort ratio of a user is defined as the ratio between his seeding and leeching time. We demonstrate the behavioral differences between users in different groups, we argue the reasons for these differences, and we show the positive and negative effects of these behaviors based on metrics including the seeding time, the upload speed, and the evolution of sharing ratio. In previous chapters we have analyzed the advantages and disadvantages of SRE schemes based on theoretical models, simulations, and measurements. This chapter complements these chapters by presenting observations from real world communities with a focus on the user behavior patterns.

The main contributions of this chapter are as follows:

1. We perform a measurement study of three private communities that provide user-level information including the upload amount, the download amount, the seeding time, the leeching time, and the sharing ratio of *each* individual user. Among the

dozens of existing communities we have examined, these are the only ones that provide such detailed information. We use one of the three communities as an example to explore the user behaviors (Section 5.1).

2. Based on the sharing ratio we classify users into the *rich*, the *middle class* and the *poor*. We show that, to maintain adequate sharing ratios, all users have to seed for excessively long times (compared to their downloading times), though most of the time their seedings are not very productive and their long seeding times do not necessarily lead to large upload amounts. For users who intend to increase their sharing ratios, we find that seeding for longer durations is not as effective as increasing the upload speed, which can be achieved by upgrading the internet access, or as joining swarms in their early stages to avoid situations of oversupply (Section 5.2).
3. Based on the age we classify users into the *new* and the *old*. We find that old users are more dedicated to the community, in terms of higher effort ratios, while new users in general seed more productively (Section 5.3).
4. Based on the download amount we classify users into the *big* and the *small consumers*. We find that big consumers are often at the same time big contributors, and are more active than small consumers in terms of both longer seeding times and longer leeching times (Section 5.4).
5. Based on the effort ratio we classify users into *gamers* (with high effort ratios) and *dedicators* (with low effort ratios). We find that gamers not only leech longer but also seed shorter than dedicators, and at the same time they maintain lower sharing ratios, which, however, are still high enough for them to stay in the community (Section 5.5).
6. Based on the user behaviors we defined, we analyze strategies that alleviate the negative effects of these user behaviors from both the users' and the community administrators' perspective (Section 5.6).

## 5.1 Methodology

In order to obtain a better understanding of private BitTorrent communities it is critical to be able to collect data on their operation. Over the years it has been proven to be a challenge to obtain detailed traces of user behavior, due to a combination of technical constraints and privacy concerns. For instance, prior work was never able to capture both detailed user profiles, content availability, and precise information on every user download.



---

To support our analysis, we have examined 38 elite private communities, out of which we selected three communities, CHDBits [12], HDStar [41]), and ChinaHDTV [15], for detailed regular deep crawling of HTML pages. These specific three communities were selected as they are the only ones that provide information detailed enough for our analysis. We have obtained the following three datasets for each community:

1. **Community-level user profile:** in this dataset, we crawl the profile page of *each* community user and obtain the information of his upload and download amount, his seeding and leeching time, his sharing ratio at the time of snapshot, and the time he joined the community.

It should be noted that the seeding time of each user recorded by the tracker is *swarm-based*, i.e., simultaneously seeding in multiple swarms counts separately. For instance, after a user has seeded in two swarms for 10 hours,  $2 \times 10 = 20$  hours will be added to his seeding time. Similarly, the leeching time recorded by the tracker is also swarm-based. In later sections, when we calculate the average upload speed of a user, we calculate his per-swarm average upload speed, i.e., the total upload amount divided by the swarm-based seeding time. In this way, we get a rough estimation of a user's seeding time and upload speed. Though more accurate calculation of the seeding time and upload speed would be better, to the best of our knowledge, until now no private communities provide this information and it is also impossible to deploy a client and contact every user individually to get this information.

2. **Community-level torrent profile:** in this dataset, we crawl the community trackers and collect information of *each* torrent, including the number of seeders and leechers, the number of finished downloads at the time of the snapshot, and the time the torrent was published.
3. **Torrent-level user activity:** the tracker records a user's torrent-level action times, such as the time of joining the swarm, the time of starting seeding, etc. The precision of the recorded action time decreases with time. For example, if a user started to seed 10 hours ago, its action time will be "10 hours ago". However, if a user started to seed one month, 23 days, and 10 hours ago, its action time will only be "one month and 23 days ago".

In order to obtain the action times with precision in hours, for each community, we examine all the torrents released within 24 hours. We follow these torrents for 7 days and record the activity of *each* user who has participated or is participating in one of them. The collected information includes each user's per-swarm upload amount, download amount, seeding time, and leeching time, as well as the time he joins and leaves the swarm.

Private communities often consist of tens of thousands of users and torrents, with total download amount in the order of petabytes. For instance, when we collected the data in May, 2011, CHDBits had 24,633 registered users, 40,040 torrents, and a total download amount of 24.3 PB. We have analyzed the measurements of the three communities in detail. As we show later in Section 5.2, they demonstrate similar performance. For simplicity of presentation, we only demonstrate the results of CHDBits in Sections 5.3 and 5.5 .

The detailed user behavior information allows us to explore the user behavior in private communities. User behavior directly decides a user's basic movements including (1) when to join the community, (2) how much to consume, and (3) how much to contribute, from which it further decides (4) a user's status in the community, in terms of the sharing ratio he achieves. Based on these four metrics we classify users into different groups, we demonstrate their behavioral differences, we argue the reasons for these differences, and we show the positive and negative effects of these behaviors. We start from the sharing ratio, since without a proper one a user cannot even stay in the community, which implies its fundamental importance.

## 5.2 The rich and the poor: positive and negative effects of SRE

In this section, we divide users into different groups based on their sharing ratios. We analyze the reasons for some users to achieve low sharing ratios, from which we demonstrate the positive and the negative effects of SRE.

### 5.2.1 A general view

We first show in Fig. 5.1 the cumulative distribution function (CDF) of the user sharing ratio in CHDBits, in HDStar, and in ChinaHDTV, respectively (dataset 1). We see that most users in these three communities achieve sharing ratios larger than the SRE threshold, i.e., 0.7. Take users in CHDBits for example, around 15% users have sharing ratios less than 1 (defined as the *poor*), while around 18% users have sharing ratios larger than 5 (defined as the *rich*). The rest that have sharing ratios between 1 and 5 are defined as the *middle class*. The behavior of accumulating a large sharing ratio may be triggered by various motivations, such as altruism, a desire to be part of the rich elite of the community, or a habit of saving sharing ratio for the future. The rich peers have little worry about staying in the community, since their sharing ratios are far beyond the SRE threshold. On the other hand, poor peers are at the risk of being expelled from the community. As a consequence, they need to be concerned a lot about their decisions: they may download

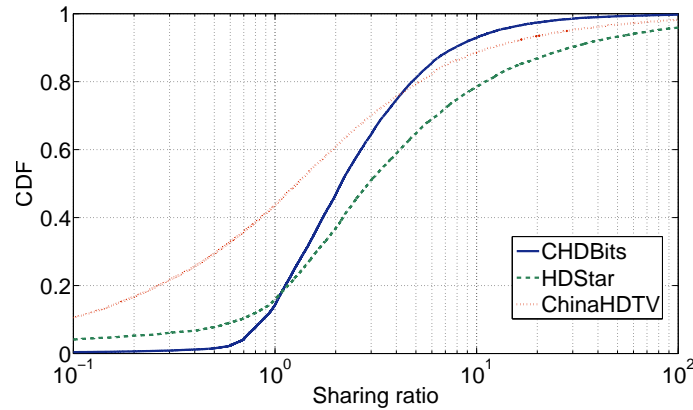


Figure 5.1: The CDF of the user sharing ratio in CHDBits, in HDStar, and in ChinaHDTV. The horizontal axis is in log scale.

new contents they really desire, but this might reduce their sharing ratios to a more risky level.

One may argue that the poor peers are free-riders, who intend to keep low and risky sharing ratios that are just enough to stay in the community. However, the highly restricted membership in private communities, especially in CHDBits and many other private communities where new members can only join by a limited number of invitations, makes it very difficult to get a new membership. Hence, we conjecture that not all poor peers are strategic and psychologically strong enough to face being expelled from the community due to insufficient sharing ratios. Interestingly, as we will show in the following sections, the poverty is partially induced by the fact that the poor peers are not strategic enough.

### 5.2.2 Long seeding time, even for the poor

Many previous studies have shown that under SRE, users seed for long durations [13, 64, 69, 124]. They consider this as a positive effect of SRE since long seeding durations lead to high download speeds. However, in this section we argue that the long seeding durations can also be seen as a negative effect, especially for poor peers.

Figs. 5.2, 5.3, and 5.4 show the CDFs of the seeding time and the leeching time in CHDBits, in HDStar, and in ChinaHDTV, respectively (dataset 1). Consistent with the theoretical results of our previous work [49,51], in general the seeding time is much longer than the leeching time in all these three communities. Take CHDBits for example, the median leeching time is 70 days while the median seeding time is 1,100 days. Remember that the seeding and the leeching time of users are swarm-based, leading to very high values.

Intuitively, longer seeding times than leeching times for rich peers are to be expected, since rich peers are saving sharing ratios by seeding. However, we observe from Figs. 5.2,

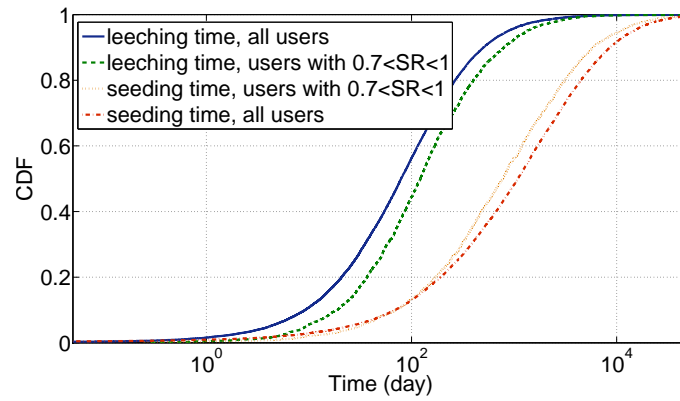


Figure 5.2: The CDF of the seeding and the leeching time in CHDBits. The horizontal axis is in log scale.

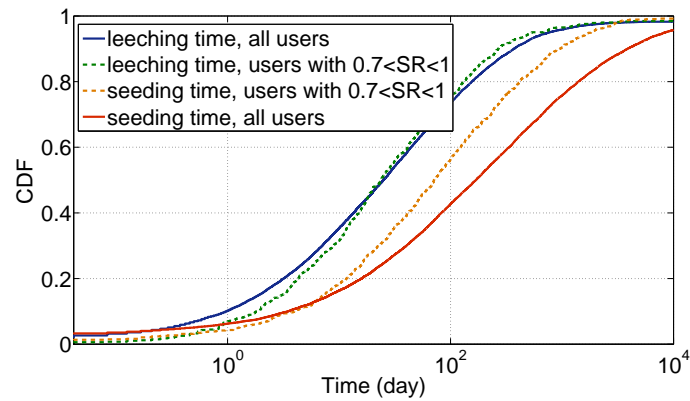


Figure 5.3: The CDF of the seeding and the leeching time in HDStar. The horizontal axis is in log scale.

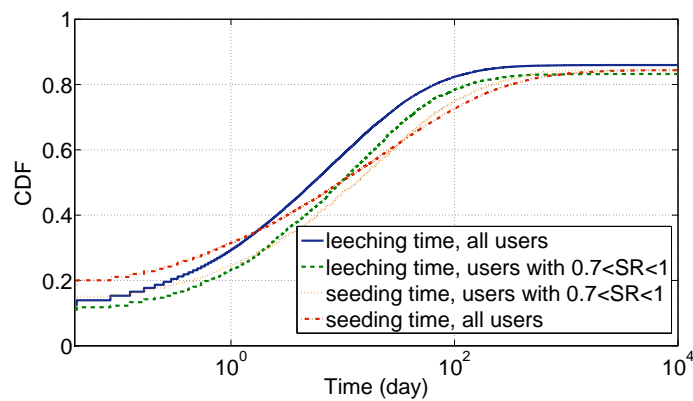


Figure 5.4: The CDF of the seeding and the leeching time in ChinaHDTV. The horizontal axis is in log scale.

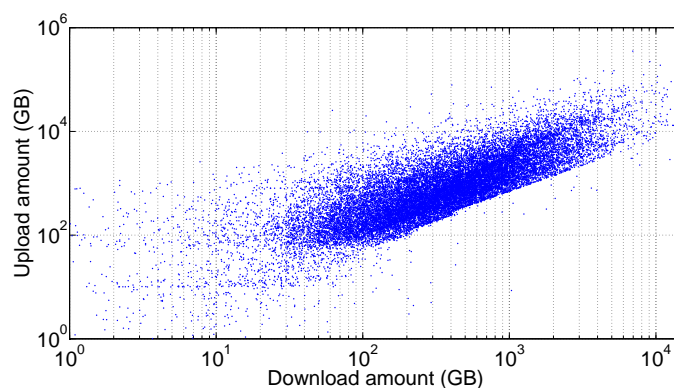


Figure 5.5: The upload amount versus the download amount (with Spearman’s rank correlation coefficient equal to 0.8110). The horizontal axis is in log scale.

5.3, and 5.4 that, even poor users seed much longer than they leech. While intuitively poor peers should be the ones that are not “hard-working” enough, why do some of them seed for long durations but still have low sharing ratios? In the following section, we explore the possible reasons.

As shown in Figs. 5.1 to 5.4, users in these three communities behave similarly. For simplicity of presentation, from now on we only show the results for CHDBits, without explicitly stating so. For the results shown in later sections, the datasets of the other two communities demonstrate similar performance as that of CHDBits [45].

### 5.2.3 Possible reasons?

One may argue that the long seeding times of poor peers are due to the fact that even though they contribute more, they also consume more. Hence, they seed for long durations but they still have low sharing ratios. This argument is partially true. Andrade *et al.* [5] have shown and we also observe from our measurement (Fig. 5.5, dataset 1) that the individual upload amount (contribution) increases with the corresponding download amount (consumption), with the Spearman’s rank correlation coefficient equal to 0.8110. Spearman’s rank correlation coefficient assesses how well the relationship between two variables can be described using a monotonic function [99]. However, this doesn’t necessarily mean that heavy contributions induce long seeding times, nor does it mean that long seeding times lead to heavy contributions.

Quite counter-intuitively, as shown in Fig. 5.6(a), a peer’s upload amount has little relation to its seeding time: many peers seed for long durations but only have uploaded relatively small amounts of data, while other peers seed for relatively short durations but have successfully achieved large upload amounts. The same argument is also applicable to poor peers (Fig. 5.6(b)). This interesting phenomenon implies that for poor peers who

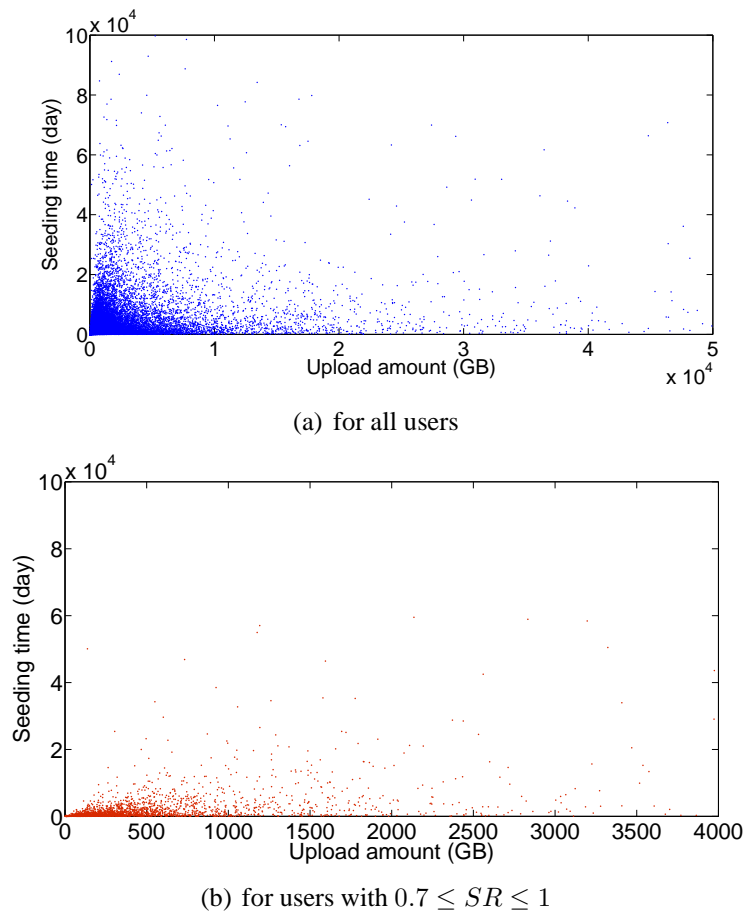


Figure 5.6: The seeding time versus the upload amount.

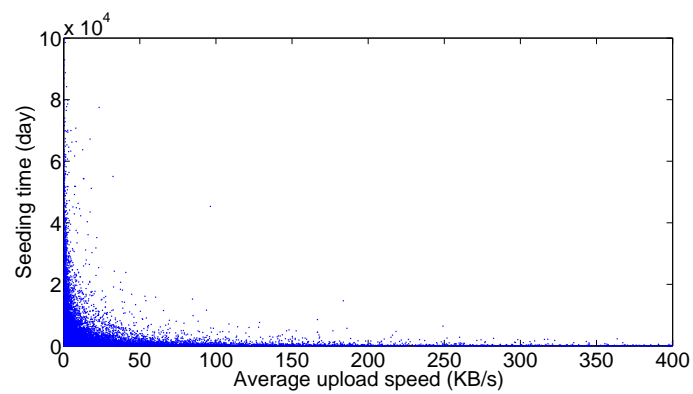


Figure 5.7: The seeding time versus the upload speed (with Spearman's rank correlation coefficient equal to  $-0.6318$ ).

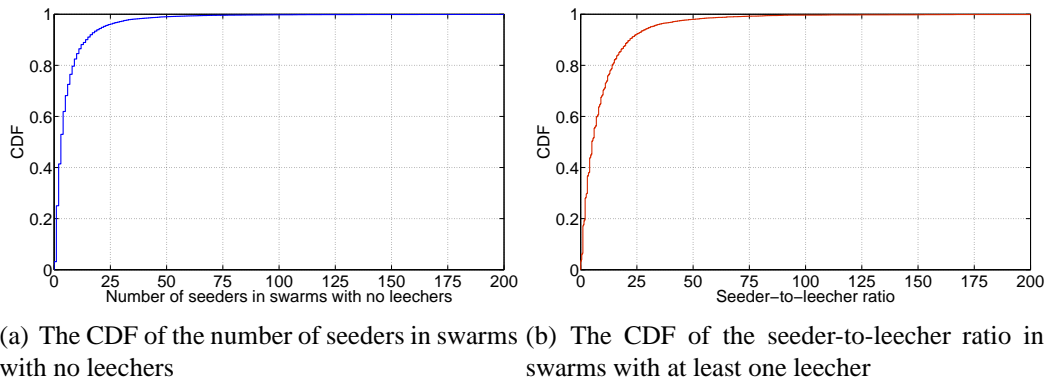


Figure 5.8: The oversupply in swarms.

intend to increase their upload amount to become rich, seeding for longer durations may not be an effective method, even if intuitively it seems so.

Though there is no strict relationship between a peer's seeding time and its upload amount, we do observe that a peer's seeding time is related to its average upload speed, regardless of its upload amount. As shown in Fig. 5.7, most of the long seeding durations happen to the peers with relatively small upload speeds, and for peers who have high upload speeds, the seeding times are normally short.

The most intuitive reason for a low upload speed is a limited internet access. However, we argue that this is not the only reason. From dataset 2, at the time when we crawled the site, CHDBits had 33,041 active swarms (with at least one leecher or one seeder), among which 26,402 swarms (79.9%) had no leechers at all. As shown in Fig. 5.8(a), 40% of the swarms with no leechers still have at least 5 seeders, and 5% of these swarms even have more than 20 seeders. For swarms with at least 1 leecher, the seeder-to-leecher ratio (SLR) is quite high: as shown in Fig. 5.8(b), 50% of these swarms have as SLR larger than 6, and 5% of these swarms even have as SLR larger than 30. We see clearly that a majority of the swarms in CHDBits are heavily oversupplied. In such swarms, seeders are not able to perform any actual uploads due to the insufficient demand and unsatisfied supply. We term this situation *unproductive seeding*. As a consequence, users have to seed for excessively long durations to achieve the sharing ratio required by SRE.

While a low upload speed mainly leads to a long seeding time, in the next section we show its influence on a user's status. We analyze the reasons for the poor being poor and discuss strategies for users to become rich efficiently.

#### 5.2.4 Why the poor are poor and how to become rich?

As the sharing ratio is defined as the ratio between a peer's upload and download amount, two possible reasons for a peer being poor are that it has downloaded too much or has uploaded not enough. The download amount depends on a user's interests in contents.

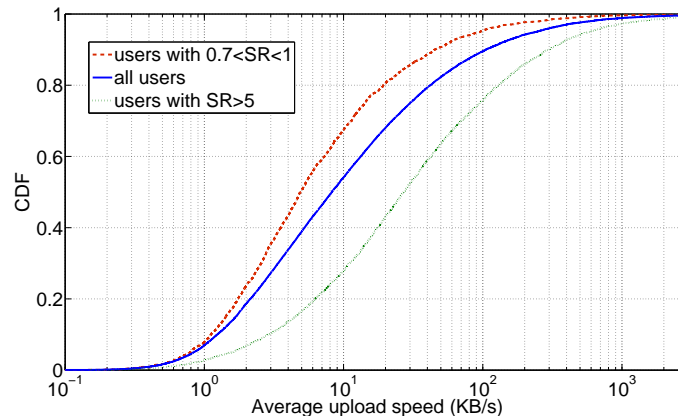


Figure 5.9: The CDF of the average upload speed. The horizontal axis is in log scale.

We do not suggest users to download less so as to become rich, since the fundamental user experience that should be guaranteed by communities is that users should not need to limit their download needs. Following this argument, in this section we focus on the user upload activity and analyze why some users have uploaded not enough (hence, are poor) and how they can improve it (to become rich).

### Community level

In Section 5.2.3 we have shown that the seeding time has little influence on the upload amount but the upload speed does. The upload speed further influences whether a user is rich or poor. As shown in Fig. 5.9 (dataset 1), in general rich peers ( $SR \geq 5$ ) have much higher upload speeds than poor peers ( $SR \leq 1$ ). For example, 80% of the poor peers upload at a speed less than 20 KB/s, while at least 40% rich peers can upload at a speed larger than 50 KB/s. Together with the result in Section 5.2.3, we conclude that instead of seeding for longer durations, peers who intend to become rich should seed with higher upload speeds. And to seed with a higher upload speed, a user could upgrade its internet access or choose a swarm that is less oversupplied.

One may argue that the above analysis is based on community-level activities, which only provide a macroscopic view that is not enough to show the underlying details. To explore this, in the following subsection we focus on a single swarm and demonstrate the torrent-level user performance, and we discuss possible strategies for users to become rich.

### Torrent level

Among all the CHDBits swarms in dataset 3, we choose the one with the largest number of participants as the example to show the torrent level user behaviors. In total, 3,776



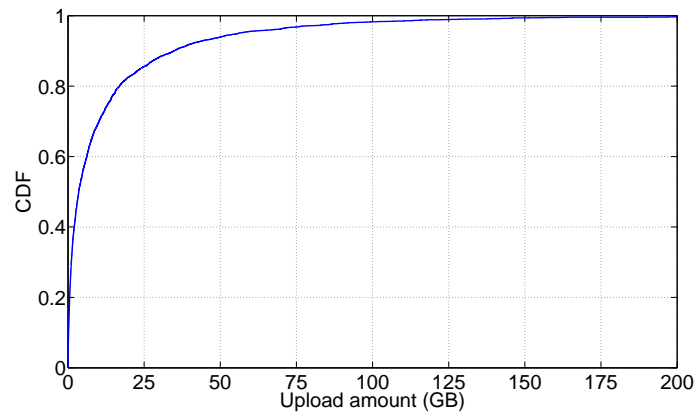


Figure 5.10: The CDF of the upload amount in one swarm.

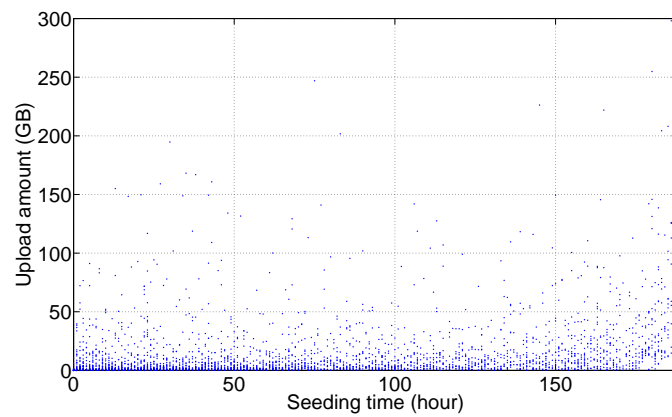


Figure 5.11: The upload amount versus the seeding time in one swarm.

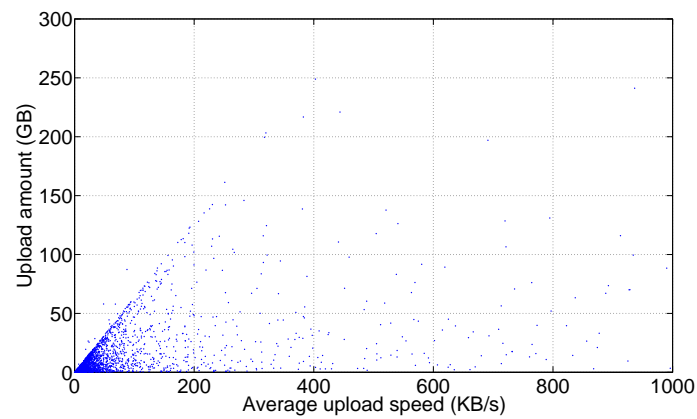


Figure 5.12: The upload amount versus the upload speed in one swarm (with Spearman's rank correlation coefficient equal to 0.7876).

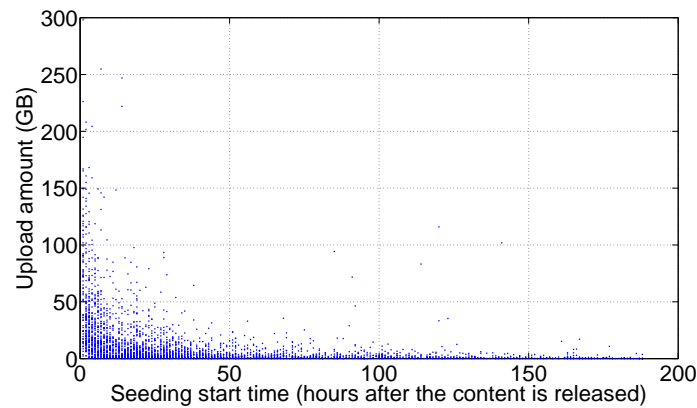


Figure 5.13: The upload amount versus the time of starting seeding in one swarm (with Spearman’s rank correlation coefficient equal to  $-0.6491$ ).

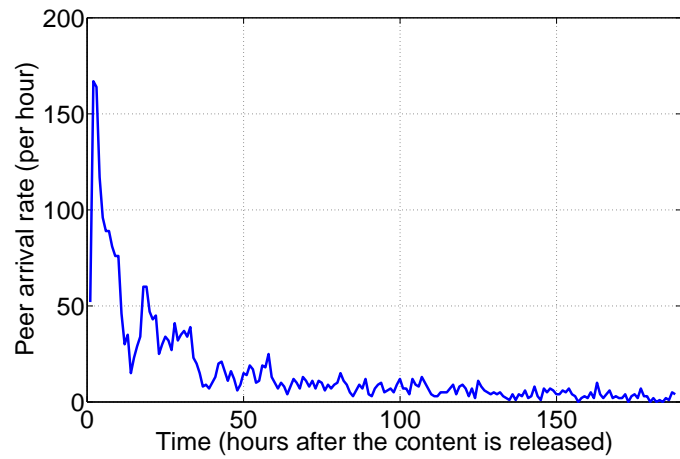
users are included.

**Different individual upload amount in one swarm:** Fig. 5.10 shows the CDF of the user upload amount in a single swarm (dataset 3), from which we observe that a small fraction of users have uploaded considerably more than the others. For example, 60% of the users have uploaded less than 10 GB, which is less than the amount they have downloaded (11.7 GB). On the other hand, 5% of the users have uploaded more than 50 GB. Of course, the users who managed to upload more will become richer. While these users have participated in the very same swarm, why did some manage to gain a lot while others didn’t?

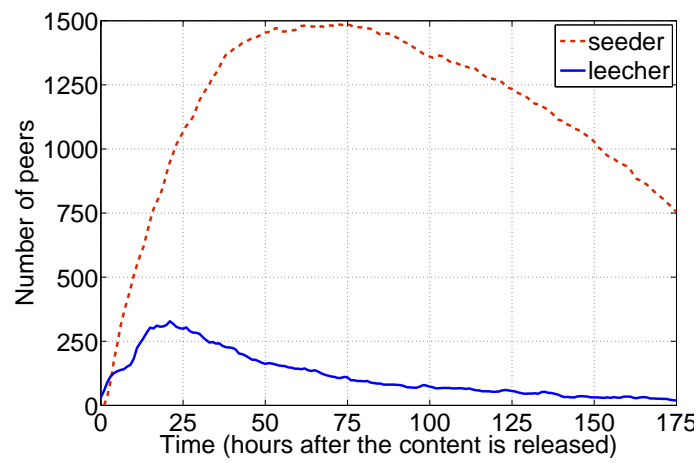
**Possible reasons and how to gain more:** One intuitive reason for a small upload amount is a short seeding time. However, similar to the analysis in Section 5.2.3, again we find the counter-intuitive result that in one swarm a peer’s upload amount is not related to its seeding time (Fig. 5.11). On the other hand, it is related to its upload speed. As shown in Fig. 5.12, most of the small upload amounts happen to the peers with relatively low upload speeds, and peers with high upload speeds normally have uploaded a large amount.

When we organize the peers according to the time they start to seed, we find another interesting phenomenon: peers that start to seed earlier normally have uploaded more (Fig. 5.13). The same phenomenon has also been observed by Kash *et al.* in [57]. One may argue that the peers who start to seed earlier can seed for longer durations, hence they upload more. However, in Fig. 5.11 we already show that the upload amount is not related to the seeding time. Then why do peers that start to seed earlier upload more?

As shown in Fig. 5.14(a), after the burst at the first two hours since the file was published, the peer arrival rate decreases dramatically. On the other hand, the number of seeders increases quickly at the first 60 hours, then decreases with a much smaller rate (Fig. 5.14(b)). In general, the number of leechers is negligible compared to the number



(a) The peer arrival rate (per hour)



(b) The evolution of the numbers of leechers and seeders

Figure 5.14: The performance in one swarm.

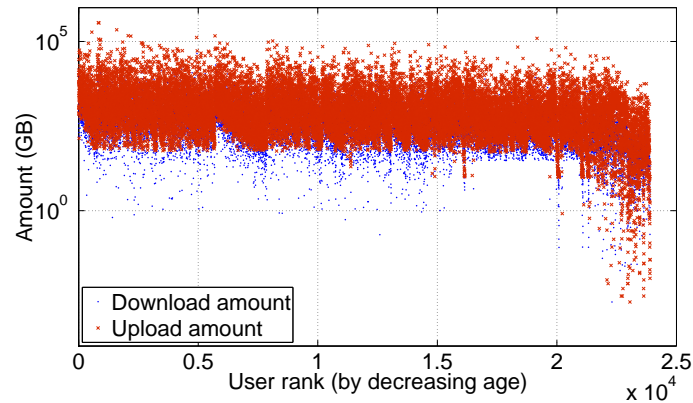


Figure 5.15: The upload and the download amount versus the age. The vertical axis is in log scale.

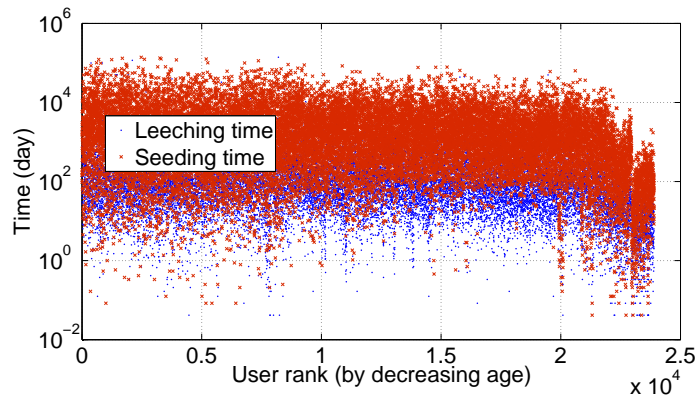


Figure 5.16: The seeding and the leeching time versus the age. The vertical axis is in log scale.

of seeders. As a consequence, peers who join late have to compete with a large number of seeders for uploading, which leads to a low upload speed, and hence a small upload amount. Therefore, peers who intend to become richer should join the swarm in its early stage, when it is still not extremely oversupplied.

### 5.3 The old and the new: how users evolve

The first behavior of any user is to register as a member and join the community. In this section, we explore the behavioral differences between users of different ages.

### 5.3.1 A general view

After joining the community, users gradually build up a history of uploads and downloads, in terms of the amount they consume and contribute, and the time they spend in leeching and in seeding. Fig. 5.15 shows the scatter plot of the upload and the download amount of each user, with the users ranked in the reverse order of their ages. We see that, in general, the upload and the download amount of the first 20,000 users are stable with their ages, while the remaining ones demonstrate a decreasing trend when the age decreases. We observe a similar correlation between the seeding and the leeching time of users and their ages, as shown in Fig. 5.16. With these clear behavioral differences we divide users into two groups, the first 20,000 users as *the old* and the other ones as *the new*. Next, we explore more behavioral differences between these two groups of users.

### 5.3.2 Level of commitment: on the way to become more committed

The most straightforward measure for a user's commitment to the community is its upload amount, which is exclusively decided by the seeding time and the average upload speed. As discussed in Section 5.2.4, the average upload speed is not a simple reflection of user's bandwidth, but an outcome of the swarm status like the number of seeders and leechers in the swarm. As it is difficult for users to control their average upload speeds, we use only the seeding time to reflect their level of dedication. Further, to avoid the cumulative effect of age, i.e., that older users have stayed in the community longer and therefore have higher opportunities to seed and leech longer, we use the ratio between the seeding and the leeching time, which is defined as the *effort ratio*, as the metric to measure user dedication to the community.

In Fig. 5.17 we show the CDF of the effort ratio of old and new users, respectively. We see that most users (both the old and the new) achieve very high effort ratios, indicating that with SRE, users in private communities are highly committed (or forced to be). More interestingly, in general, new users achieve lower effort ratios than old users, indicating that as users evolve in the community, they become more committed, which implies a deeper (psychological) effect of SRE on the old users.

### 5.3.3 Average upload speed: the new have not yet suffered

Under SRE, users will be expelled if they cannot upload enough to meet the SRE requirement. Intuitively, similar to the effect of Group Selection in biology, users with relatively low bandwidth will gradually become extinct, leaving only the "strong" ones to survive in the community. Nevertheless, Fig. 5.18 shows the CDF of the average upload speed achieved by old and new users, respectively, from which we see that in general new users achieve higher upload speeds. This counter-intuitive observation confirms our analysis in

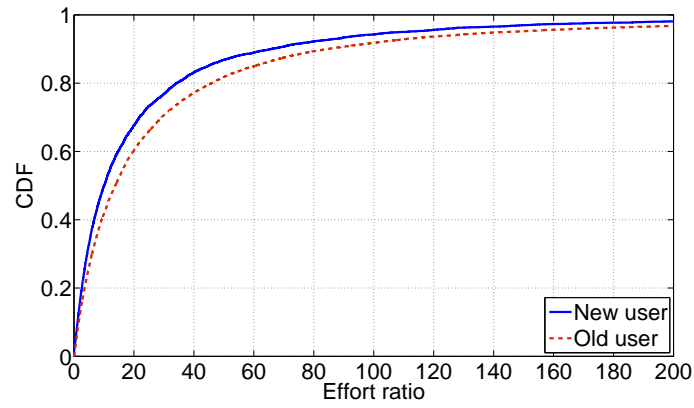


Figure 5.17: The CDF of the effort ratio of old and new users.

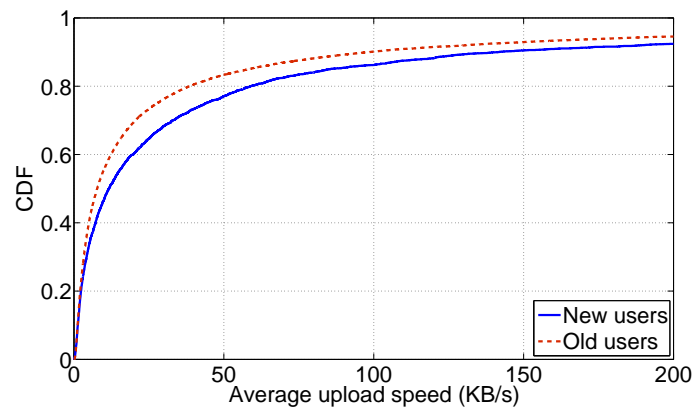


Figure 5.18: The CDF of the average upload speed of old and new users.

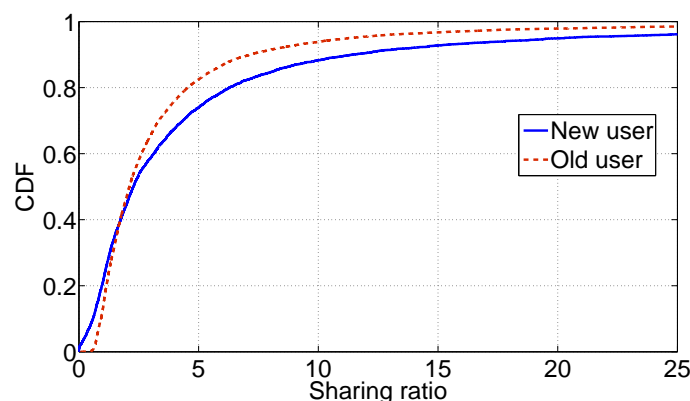


Figure 5.19: The CDF of the sharing ratio of old and new users.

Section. 5.2.4 that a user's average upload speed highly depends on the status of swarms it has participated in. When a user with a high bandwidth spends long time in unproductive seeding, it can achieve a very low upload speed. We have shown in Fig. 5.17 that new users are less dedicated to the community. Therefore, they avoid long unproductive seeding time and achieve higher upload speeds.

### 5.3.4 Sharing ratio: the spreading new and the conservative old

As discussed in Section 5.2.4, there are both internal and external reasons for users being poor or rich. In Fig. 5.19 we show the CDF of the sharing ratio achieved by new and old users, respectively. We see that the line for new users is more skewed, indicating that new users achieve a larger range of sharing ratios than old users. The lower end is due to the fact that new users are often given start-up time to increase their sharing ratios, while the higher end is normally due to their small download amounts. On the other hand, old users behave conservatively, without many risky sharing ratios below the SRE threshold or excessive sharing ratios requiring huge contribution and little consumption.

## 5.4 The big and the small consumer: active users active in all

How much to consume is a fundamental user behavior in private communities, since the main goal for users to join is to download the contents they are interested in. In this section, we explore the behavioral differences between users with different download amounts. As discussed in Section 5.3, a user's age has a cumulative effect on his download and upload amounts, and on his seeding and leeching times. This implies that as users evolve in the system, these four metrics gradually increase. To avoid this cumulative

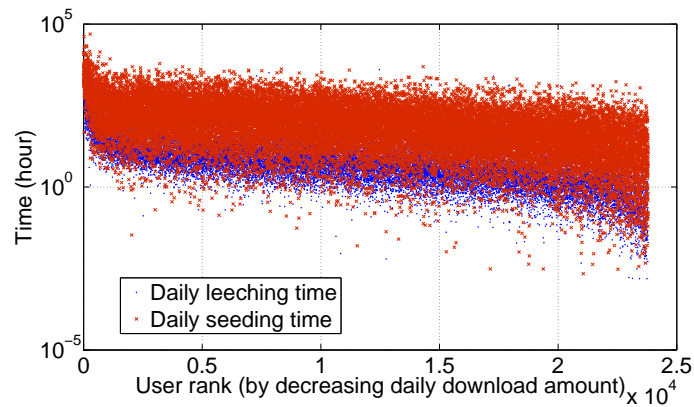


Figure 5.20: The daily seeding and leeching time versus the daily download amount. The vertical axis is in log scale.

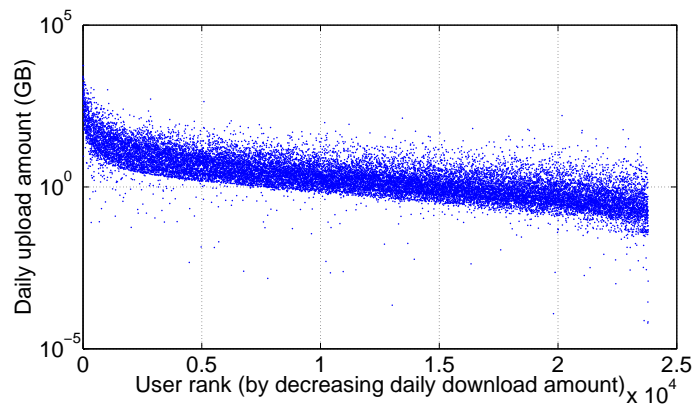


Figure 5.21: The daily upload amount versus the daily download amount. The vertical axis is in log scale.

effect, we consider the daily values for these metrics.

Figs. 5.20 and 5.21 show the daily seeding time, the daily leeching time, and the daily upload amount of each user, with users ranked according to the decreasing order of their daily download amounts. We see a clear decreasing trend of these three metrics when the daily download amount is decreased, indicating that users with larger consumptions normally contribute more, in terms of both the time and the amount they contribute. This also implies that active users are active in both downloading and uploading.

In Figs. 5.22 and 5.23 we show the sharing ratio and the average upload speed for each user, with users again ranked according to the decreasing order of their daily download amounts. This time we observe no clear correlation between these metrics, except that some users at the right end achieve extremely large sharing ratios. We believe this is due to their extremely small daily download amount.



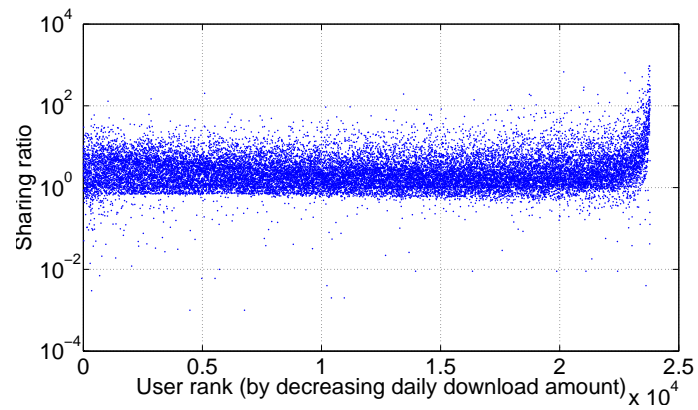


Figure 5.22: The sharing ratio versus the daily download amount. The vertical axis is in log scale.

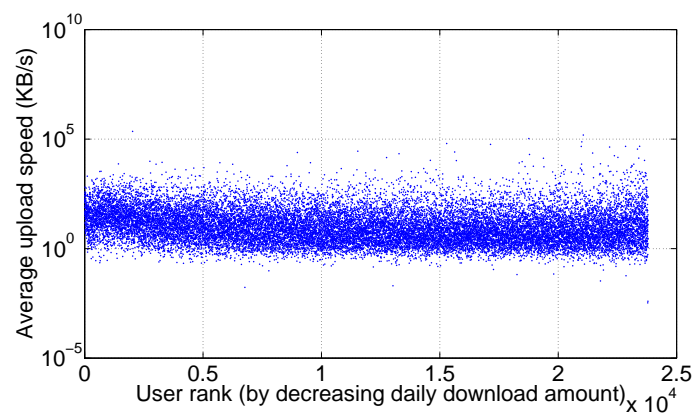


Figure 5.23: The average upload speed versus the daily download amount. The vertical axis is in log scale.

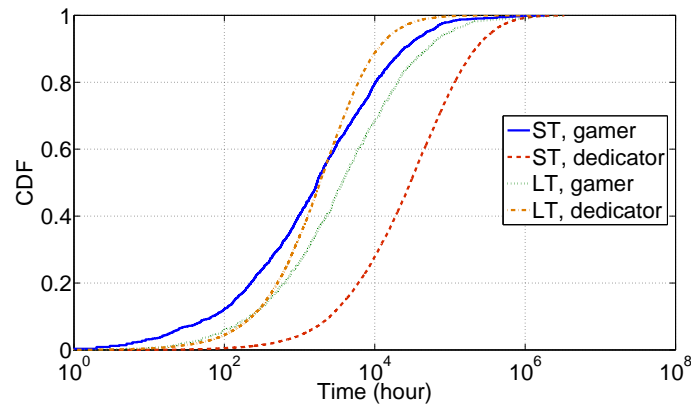


Figure 5.24: The CDFs of the seeding and the leeching time of gamers and dedicators. The horizontal axis is in log scale.

## 5.5 The gamer and the dedicator: Gemini of the private community

How much to contribute is an important choice users make in private communities. Based on the level of contribution, a user can be a *gamer* who games the system, explores the potential benefits, and avoids providing much contribution, or a *dedicator* who dedicates himself to the system and provides high contribution—two opposite user behaviors evolved in the same private community, just like the Gemini.

The upload amount is often used to measure a user’s contribution level. Nevertheless, while users can decide the time they contribute, they do not have full control of the upload speed. Therefore, as in Section 5.3.2, we again use the effort ratio as the metric for deciding a user’s contribution level, and we define users with effort ratio less than one as gamers and the rest as dedicators. Next, we explore more behavioral differences between these two groups of users.

### 5.5.1 Seeding and leeching time: which decides a gamer?

The reasons for a low effort ratio could be a short seeding time and/or a long leeching time. In Fig. 5.24 we show the CDFs of the seeding and the leeching time of gamers and dedicators, respectively. We see that, in general, gamers leech longer while they also seed shorter than dedicators, indicating that gamers not only contribute less but also potentially consume more.

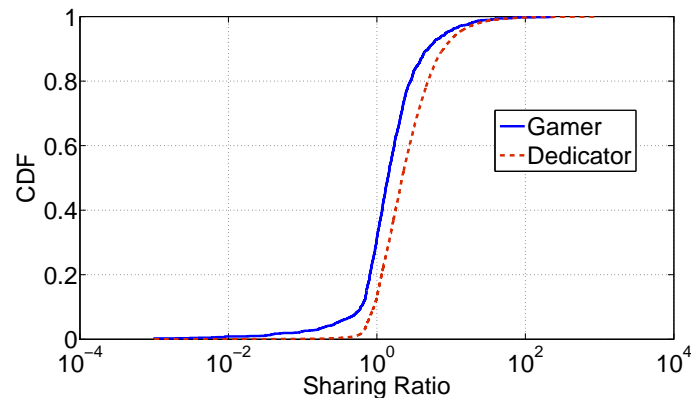


Figure 5.25: The CDFs of the sharing ratio of gamers and dedicators. The horizontal axis is in log scale.

### 5.5.2 Sharing ratio: gamers seize the day

In Fig. 5.25 we show the CDF of the sharing ratio achieved by gamers and dedicators, respectively. We see that in general, gamers achieve much lower sharing ratios. As the sharing ratio in a private community serves as virtual credit that can be spent in future downloads, the difference in sharing ratio implies that gamers do not hoard sharing ratios for future downloads as much as dedicators do. In other words, they are more the seize-the-day type: they keep sharing ratios that are just enough for them to stay in the community, so that they could explore the benefit of downloading (reflected by the long leeching time) and in the mean time provide little contribution (reflected by the short seeding time).

### 5.5.3 Average upload speed: gamers know the way

Though gamers achieve lower sharing ratios than dedicators, they still need to meet the SRE threshold to stay in the community. With the short seeding time and the long leeching time, gamers have to increase their upload speed efficiently so as to maintain their sharing ratios above the SRE threshold. Consistent with our intuition, they do achieve higher upload speed than dedicators. As shown in Fig. 5.26, apparently, gamers can achieve upload speeds an order of magnitude higher than dedicators. As discussed in Section 5.2.4, achieving high upload speed often requires users to be strategic such as joining swarms early, which implies that gamers not only intend to, but also know the way, to game the system.

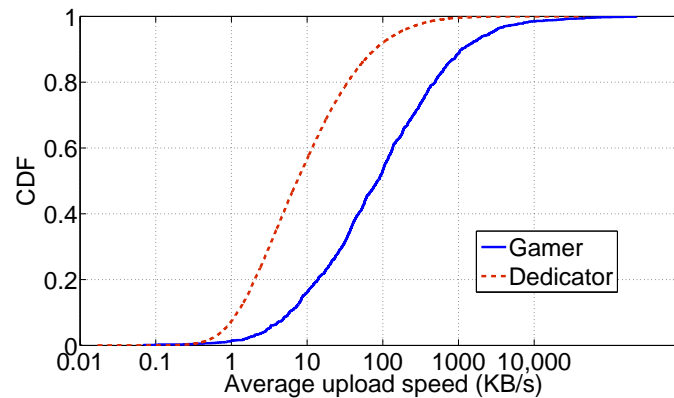


Figure 5.26: The CDFs of the average upload speed of gamers and dedicators. The horizontal axis is in log scale.

## 5.6 Discussion

Though altruistic users always exist, we conjecture that most users in private communities are *selfish*. Their initial goal in a community is to download all the contents they are interested in. To achieve this, they try to maintain the required sharing ratio while not limiting their download needs. The strategies they apply mainly optimize their own benefit, without considering the *social welfare*, i.e., the performance of other users. For example, users may seed all the files they have downloaded to increase the opportunity of performing some actual uploading during seeding. However, this directly increases the bandwidth supply and makes the upload competition even more severe [70]. As we discussed in Section 5.2.4, joining swarms earlier helps users gain sharing ratios more efficiently. However, if a majority of users strategically join a swarm immediately after a new content is published, then 1) many users will download something they don't want, only for gaining sharing ratios; 2) the download speed in the early stage of a swarm will be very low, because a large number of strategic users joining simultaneously makes the swarm heavily *flash-crowded*<sup>1</sup>; and 3) it will be more difficult to perform any actual uploads after the early stage, since only a few non-strategic users will join the swarm during that period.

Private community administrators that intend to adopt strategies, or remedies, to alleviate the side-effects of SRE, should take the potential strategic user behaviors into account. For example, some private communities try to further incentivize contribution beyond SRE by giving rich peers priority to access newly published contents [10]. However, as discussed previously, joining early in a new swarm will help the users, especially the poor users, gain sharing ratios more efficiently. By giving priority to rich peers, administrators are basically taking the opportunities away from the poor peers for gaining

<sup>1</sup>We refer a swarm to be flash-crowded when it has a sudden increase in the number of leechers.

---

sharing ratios. Unless the administrators intend to let the rich be richer and the poor be poorer (which will lead to a more intense competition and a potential deterioration of performance as discussed previously), we suggest administrators to remove these restrictions.

Another example of existing remedies for SRE adopted by private communities would be *free-leech* and *seeding-bonus*. Some communities [12, 15, 41, 81] temporarily adopt free-leech and/or seeding-bonus for certain swarms, which means that a user can download the file for free and/or get extra bonus for seeding. In free-leech periods, users are attracted to those swarms because of the low price for downloading. In this way, the bandwidth demand is increased and the oversupply in the system is alleviated. Meanwhile, when free-leech is applied to a relatively old swarm, the benefit of joining early is also reduced. The same argument is also applicable to seeding-bonus. In seeding-bonus periods, peers are attracted to the swarms to seed. Hence, when seeding-bonus is applied to old swarms, the file availability is improved. However, administrators should be careful and not adopt free-leech or seeding-bonus for a long time, otherwise strategic users might wait and not download anything until the files are for free, or only seed in swarms with seeding bonus.

In our previous work [49], we propose a self-organizing strategy named *SRE with supply-based price* that prevents this potential manipulation of strategic users. Instead of manually adopting free-leech (i.e., zero price), this strategy inversely relates the price for downloading one unit of data to the seeder-to-leecher ratio in the swarm. With a larger seeder-to-leecher ratio, i.e., an increasing supply, SRE with supply-based price automatically decreases the price. Once the supply goes tight again, it will automatically increase the price. In this way, the demand and supply are automatically balanced and reasonable downloading and seeding times are achieved.

## 5.7 Related work

To date, only few works have analyzed private communities. Zhang *et al.* [124] investigate hundreds of private trackers and depict a broad and clear picture of the private community landscape. Chen *et al.* [13] compare system behaviors among 13 private trackers and 2 public trackers, and they show their differences regarding user viscosity, single torrent evolution, user behaviors, and content distribution. Liu *et al.* [64] also perform measurement studies and further develop a model to show that SRE indeed provides effective incentives, but is vulnerable to collusion.

While these studies all focus on demonstrating the high seeding level achieved by private communities, there have been a few preliminary works that show the adverse effects. Andrade *et al.* [5] focus on the dynamics of resource demand and supply, and they show that users typically try to increase their contribution levels by seeding for longer and

not by providing more bandwidth to the system. However, our work shows that providing limited bandwidth is not the will of users, but it is a consequence of the oversupply in private communities. Chen *et al.* [14] also notice the oversupply problem and provide a model to identify the optimal stable SLR range. However, they didn't analyze the reason or propose strategies to solve the problem of oversupply. Kash *et al.* [57] demonstrate that there are significant disparities in the cost of new and old files in a private community named DIME, and users compensate for the high cost of older files by downloading more copies of newer files or by preferentially consuming older files during free-leech periods. Particularly, they have shown that after a period of free-leech, there are more download activities in the community. This is consistent with our result that during free-leech, there is more demand and the oversupply is alleviated. Besides analyzing positive and negative effects of SRE, we also extensively explore the user behaviors and argue the reasons for these behaviors. Further, we analyze the performance of well-adopted community strategies, their effects against strategic user behavior, and the remedies we proposed.

## 5.8 Conclusion

While previous work only focuses on showing the effectiveness of SRE in incentivizing users to contribute, in this chapter we provide a better understanding of private communities by exploring the user behaviors and demonstrating both the positive and the negative effects of these behaviors. We show that swarms in private communities are greatly oversupplied. Users achieve very high download speeds, but at significant expense including excessively long seeding times and very low upload speeds. Meanwhile, as users evolve in the community, some users become more committed, in terms of higher ratios of the seeding and the leeching time, and some users game the system by keeping risky low sharing ratios while they leech more often than they seed. For users who intend to increase their sharing ratios, we show that seeding for longer durations is not as effective as increasing the upload speed. If it is not realistic for users to upgrade their internet access, we suggest them to join swarms early or to join undersupplied swarms.

## Chapter 6

# Estimating user interaction strength in online networks

Online networks have become popular and powerful infrastructures for communication and they provide various mechanisms for users to interact. In this chapter, we present a new view that sees *user interactions* as the most basic underpinning of online networks such as Facebook, Wikipedia, and BitTorrent. The key research question we address is whether we can devise a framework for expressing user interactions and their strengths that is both generic and can be applied to a wide range of systems and applications.

The patterns and strengths of user interactions in online networks are prominent. For example, in BitTorrent, user interactions can be used as the foundation for designing incentive policies to promote contribution. Through estimating the interaction strengths between users in terms of the amounts or durations of uploads, system designers can make users favor the highly ranked users for future uploads. As another example, a number of applications [102, 103, 120, 121] leverage *online friendships* to enhance security, to promote cooperation, to improve item recommendation, etc. However, it has long been observed that low-interaction friendships, as exemplified by the “Familiar Stranger” [106], are prevalent, and it has been shown that the dynamics of user interactions is more representative for inferring user relationships [108, 114] than simple, statically established “binary” friendships. Distributed systems often rely on importing trust relationships from social networks such as Facebook to improve security [121]. Instead, users would be much better off by estimating their interaction strengths with others and by trusting the ones with whom they have interacted frequently. Last but not least, sociologists often rely on user interactions for identifying social ties [54, 116], and therefore a proper estimation of user interaction strength is essential.

The importance of user interactions in online networks leads to the question: *How can we estimate user interaction strength?* Previous work addressing this issue [18, 108, 114, 116] has focused only on online social networks like Facebook, and has only con-

sidered *binary* and *direct* user interactions, simply indicating whether a user has directly interacted with another user or not. To remedy this, in this chapter we propose a User Interaction Strength Estimation scheme called UISE that has a much more fine-grained notion of user interaction and that is applicable to a more general category of online networks. Further, UISE can be easily applied to distributed systems and therefore achieves a scalable design. Specifically, we make the following contributions.

1. As a model for representing user interaction histories, we introduce the *bitmap-based user interaction graph*, based on which UISE estimates user interaction strengths. UISE captures the frequencies of both direct and indirect interactions, provides disincentives for malicious user behaviors, and can be easily incorporated into distributed systems (Section 6.3).
2. We apply UISE to detect user interaction patterns in online networks. We take Facebook as an example and we derive patterns resembling those often observed in the offline human society—users in Facebook tend to interact frequently and stably with relatively small groups, occasionally with persons outside those groups, and they make new friends while in the meantime losing touch with some old friends (Section 6.4).
3. We further apply UISE to derive a decentralized scheme for estimating the time users are online, which is an important aspect of user activity and has yet not been studied before. We have implemented this application into the Tribler [84] online network and we demonstrate the scalability and the accuracy of UISE through simulations, emulations, and Internet deployment (Sections 6.5-6.8).

As it turns out, UISE achieves accurate estimations even when users hold only 5% of the global information related to user interactions. In fact, in order to maintain the accuracy of UISE, its requirement for the coverage of global information decreases with the population size, thus allowing UISE to achieve good scalability in a self-organized manner. Furthermore, although a user only possesses a partial view of the system, with UISE he can derive a ranking of users according to his estimations of their online time that highly resembles the ranking derived from the global view. Thus, UISE achieves the most important application of estimating user interaction strength, i.e., differentiating users with different levels of activity.

## 6.1 User interactions

Online networks provide various mechanisms for users to interact. In Facebook, users exchange messages, post on each other's walls, and comment on photos. In BitTorrent,



where the main application is file sharing, users meet (*rendezvous*), and if possible, upload to and download from each other. As users evolve in online networks, they gradually build up a long history of user interactions. Some of these interactions are directed, such as uploads and downloads, while others are undirected, such as *rendezvous* and a chat over a photo. Online networks provide abundant user interactions, which, however, change rapidly over time: only 30% of Facebook users consistently interact from one month to the next [108], and in BitTorrent, users who directly download from each other for one file rarely meet again—the so-called *problem of low rendezvous*.

Fortunately, users not only interact directly, but also indirectly. An indirect interaction is formed when users are linked through a sequence of interactions, such as in Facebook when a user posts on another user's wall who in turn posts on a third user's wall, and so on, or in BitTorrent when a user uploads to another user who further uploads to a third user, and so forth. When direct interactions are relatively scarce, such as in BitTorrent-like P2P systems where the problem of low *rendezvous* exists, indirect interactions provide supplementary information for inferring user relationships. Moreover, a group of direct and indirect interactions that happen within a short time frame may be caused by offline interactions. In the Facebook example, the corresponding users could have participated in some offline event together and are sharing their experiences. Indirect interactions that happen widely apart in time, however, are of limited use. For example, a user may have exchanged messages with a high-school friend three years ago and with a college friend one hour ago—these interactions hardly indicate any meaningful user relationships between the user's two friends.

To achieve a meaningful estimation of user interaction strength, the above aspects of user interaction should be considered. In the following section, we give the problem statement and discuss the challenges that need to be addressed.

## 6.2 Problem statement

User interaction strength is reflected by two aspects: the frequency of interactions and the intensity of each interaction. In this chapter, we do not consider the latter aspect in order to avoid evaluating the strengths of words, such as to decide which comment should get a higher weight, "Happy birthday" or "You look nice". Rather, in this chapter we define user interaction strength as the *frequency that two users interact*, and we propose a model for estimating user interaction strength based on this definition. In this context, the following four issues are addressed.

**Partial history versus full history of user interactions.** A properly selected partial history of user interactions is more suitable for estimating user interaction strength than a full history. First, as user interactions may change rapidly over time, the stale ones are no longer useful for inferring meaningful relationships. Secondly, keeping a full history

of user interactions induces a “glass ceiling” for new users: they will always be evaluated as less active than old users who have already accumulated a long history. When user interaction strengths are used to decide on the service level a user can get, such an unfair playing field will discourage new users from joining and will eventually decay the system.

**Direct versus indirect interactions.** As stated in Section 6.1, direct and indirect interactions are both important and therefore should both be included. In our model, we provide two options to control the level of indirect interactions to be included in the estimation of user interaction strength. The first is a *distance limit* in terms of the number of hops between interacting users. The second is an *eligible period*. Only indirect interactions that happened together with related direct interactions within this eligible period are included, e.g., when a user exchanges messages with another user who further exchanges messages with a third user, the first and third users are only linked when these interactions happened within, say one week. The distance limit and the length of the eligible period are tunable parameters, so that our design will be applicable to different applications. For example, researchers analyzing social ties can leverage our design to include different levels of indirect interactions in their models [54, 116].

**Scalability.** As an online network evolves, users involve in a huge number of interactions. Somehow, user interaction records have to be collected and analyzed to estimate user interaction strengths, but doing so at central servers may be neither scalable nor practical. We propose a decentralized approach in which the collection of interaction records and the estimation of interaction strengths are performed by the end users, where the estimations are actually used. When applied in such a distributed way, the design should be lightweight so that it will not impose a high computational load on the end users.

**Security.** Online networks are subject to security concerns. When user interaction strengths are used to decide on the service level a user can get, malicious users may disseminate false information to be better off than others, e.g., through Sybil attacks. Traditional defenses against these attacks rely on trusted identities provided by an authority or automatically imported from some other online social networks [55, 121]. However, requiring users to present trusted identities runs counter to the open membership that underlies the success of these systems in the first place, and inferring trustworthy relationships purely from online friendships has been proven to be insufficient [108, 114]. Without assuming secure super-nodes, we propose a design in which malicious users that disseminate false information will not gain any privileges, i.e., in which disincentives for malicious user behaviors are provided.

### 6.3 Design description

In this section, we introduce UISE, a user interaction strength estimation scheme that addresses the issues listed in Section 6.2. In UISE, users collect the whole or parts of the

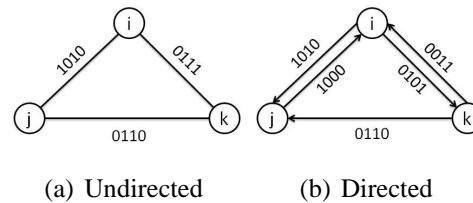


Figure 6.1: Examples of user interaction graphs

interaction histories of other user pairs through central servers or distributed information dissemination, based on which they estimate their user interaction strengths with other users. UISE consists of four parts: representing the user interaction history, estimating the user interaction strength, maintaining security against malicious user behaviors, and incorporating it into distributed systems. In the following sections, we discuss them in turn.

### 6.3.1 Representing user interaction history

In UISE, the interaction histories received by a user are incorporated into its *bitmap-based user interaction graph* (UIG), a model that we introduce to represent user relationships based on their interactions. Different users build different UIGs unless they can obtain full knowledge of the system, for example, through central servers. Therefore, a UIG reflects a user’s local view of the system.

In a UIG, a vertex represents a user and the edge between two vertices represent the interaction history of the users it connects. The interaction history of two connected users is reflected by a label of the corresponding edge, which is a string called the *interaction bitmap*, or simply *bitmap*. To capture the interaction frequency, we abstract time into cycles where one cycle represents a certain unit of time such as 30 minutes. We keep the interaction history in a time-based (cycle-based) sliding window fashion, with the window size being equal to the length of the kept history. When two users have directly interacted in a particular cycle, the corresponding bit in their bitmap is set to 1 (otherwise it is set to 0). As time evolves, their interaction bitmap becomes a binary string and the number of “1”s shows how frequently they have recently interacted.

Fig. 6.1 shows examples of UIGs for undirected and directed user interactions, respectively, with a window size of 4. For example, when Fig. 6.1(a) is derived from wall posts in Facebook, it specifies that users  $i$  and  $j$  have chatted on each other’s wall in cycles 1 and 3. When the example is derived from the upload and download interactions in BitTorrent, Fig. 6.1(b) specifies that user  $i$  has uploaded to user  $k$  in cycles 2 and 4.

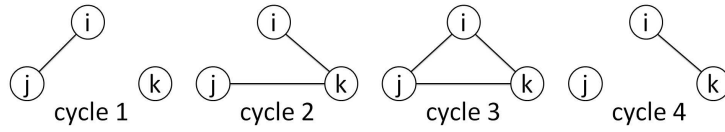


Figure 6.2: Per-cycle user interaction graphs generated from Fig. 6.1(a).

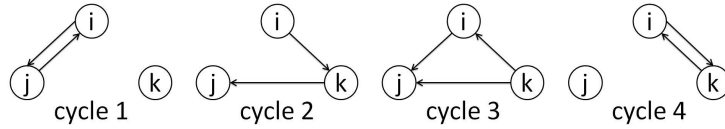


Figure 6.3: Per-cycle user interaction graphs generated from Fig. 6.1(b).

### 6.3.2 Estimating user interaction strength

To calculate the interaction strength, i.e., the frequency that two users interact (directly or indirectly), we perform cycle by cycle examinations. First, a UIG is divided into a number of *per-cycle* UIGs. Then, for each of these per-cycle UIGs, an algorithm for finding *connected components* or *reachability* is applied when the interactions are undirected or directed, respectively: if two users are connected (or one is reachable from the other) in a per-cycle UIG, they are considered as having interacted in that cycle. Finally, the ratio between the total number of these recognized cycles and the window size gives their interaction strength. In this way, UISE captures the frequency of both direct and indirect interactions.

Figs. 6.2 and 6.3 show the per-cycle UIGs derived from Figs. 6.1(a) and 6.1(b), respectively. In Fig. 6.2, an algorithm for finding connected components is applied. Here, users  $i$  and  $j$  have interacted directly in cycles 1 and 3, and indirectly in cycle 2. Therefore, their user interaction strength is estimated to be 0.75. Similarly, in Fig. 6.3 an algorithm for finding the reachability is applied and user  $j$  is reachable from user  $i$  in cycles 1, 2, and 3. Therefore, the user interaction strength from user  $i$  to  $j$  is estimated to be 0.75.

The requirement for being in the same connected component or being reachable serves two purposes. First, it specifies that only indirect interactions that happened together with related direct interactions within the same cycle are included in the estimation. Secondly, when UISE is applied in a distributed system, it alleviates the potential manipulations of malicious users, since users only trust the bitmaps that can link back to themselves in their UIGs. During the calculation of user interaction strength, we also provide the option of a distance limit in terms of the number of hops between interacting users. This limit is a tunable parameter for specifying the range of indirect interactions to be included in the calculation. With a limit of one hop, only direct interactions are included.

The connected components in an undirected graph and the reachability in a directed graph can be computed in linear time (in terms of the numbers of the vertices and edges of

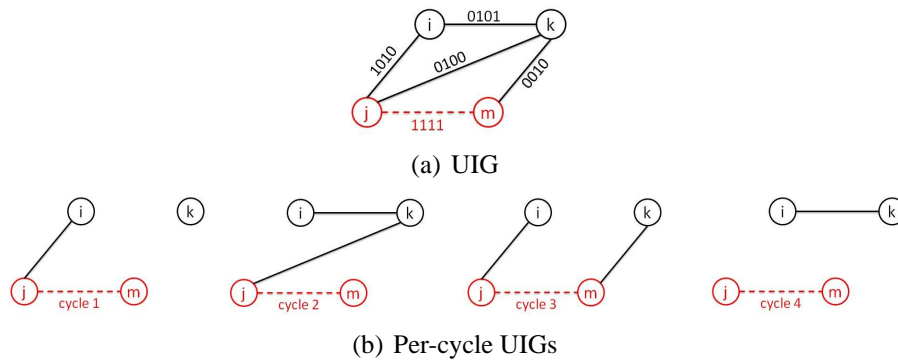


Figure 6.4: An example of user interaction graph (UIG) with false information from user  $j$  and user  $m$

the graph) using either breadth-first search or depth-first search [74]. Thus, UISE achieves linear time complexity for estimating user interaction strength.

### 6.3.3 Maintaining security

Through UISE a user can rank other users based on its interaction strengths with them. When some privilege is give to the high-ranked ones, malicious users are incentivized to spread false information to be better off than others. One prevalent way to do so is through a *Sybil attack* [28, 109]. Under a Sybil attack, the attacker generates multiple *sybils* with fake identities, and together they distribute false information about strong interactions among them. These forged interaction histories cause false edges to be added to the per-cycle UIGs of other users, where the attacker manages to connect to its sybils and to further connect to more users (victim users) through its sybils. In this way, the attacker potentially achieves higher estimated interaction strengths at victim users than without the sybil attack. Nevertheless, the false edges not only connect the attacker and victim users, but also victim users themselves (indirectly). Similarly, they also potentially increase the estimated user interaction strength between victim users. So, spreading false information does not make the malicious users be better off than others. In this way, UISE actually provides disincentives for malicious user behaviors.

Fig. 6.4 shows an example of a Sybil attack. Here, the attacker uses its real identity,  $j$ , to interact with user  $i$  in cycles 1 and 3, and with user  $k$  in cycle 2; it uses its fake identity,  $m$ , to interact with user  $k$  in cycle 3. Further,  $j$  and  $m$  claim that they have interacted in all 4 cycles. Because of the false edge added between  $j$  and  $m$  (represented by the dashed line), the attacker is connected to the victim user (user  $k$ ) through its Sybil  $m$  in cycle 3 (Fig. 6.4(b)). As a consequence, user  $k$  is considered to have interacted with the attacker in cycles 2 and 3, while originally only in cycle 2. On the other hand, user  $k$  is now connected to user  $i$  (another victim user) through  $m$  in cycle 3. It is considered to

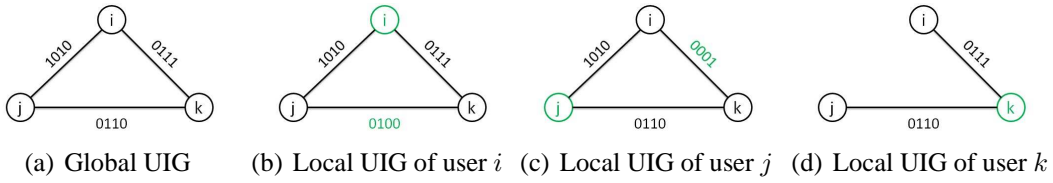


Figure 6.5: An example of the global and local user interaction graphs (UIGs)

have interacted with  $i$  in cycles 2, 3 and 4, while originally only in cycles 2 and 4.

### 6.3.4 Incorporating into distributed systems

We now show how UISE can be incorporated into a distributed system without central servers, which comes down to the question of how a user obtains interaction histories of other user pairs and builds its local UIG, and to what extent that a local UIG resembles the global one.

We assume that in a distributed system, users can obtain information through dissemination. In UISE, after two users have directly interacted with each other for the first time in a particular cycle, they generate an *interaction record*, and disseminate this record into the system through the dissemination protocol provided by the system. Based on the interaction records obtained through dissemination, each user builds its own *local UIG*, which represents its local view of the system. Further dividing a local UIG based on cycles gives the *local per-cycle UIGs*. A local UIG is a subset of the *global UIG*, in terms of the vertices, the edges, and the interaction bitmaps. The global UIG can only be obtained when the underlying dissemination protocol achieves a 100% coverage. In this chapter, the 100% coverage case is used as the baseline for performance evaluation in Sections 6.6, 6.7, and 6.8.

Fig. 6.5 shows an example of the global and local UIGs. Notice that user  $i$  did not receive the interaction record between  $j$  and  $k$  for cycle 3. Therefore, in its local UIG (Fig. 6.5(b)), the interaction bitmap between  $j$  and  $k$  is 0100, instead of the ground truth 0110. A similar loss of information also happens to  $j$  (Fig. 6.5(c)) and  $k$  (Fig. 6.5(d)).

Until now, we have introduced the basic design of UISE. In the following sections, we demonstrate and evaluate two examples of its application. In Section 6.4, we apply it to detect user interaction patterns in Facebook, a centralized online social network. In Sections 6.5-6.8, we apply it to derive a decentralized scheme for estimating the online times of users and we evaluate its performance through simulations, emulations, and Internet deployment.

---

## 6.4 Interaction pattern detection

In this section, we apply UISE to detect user interaction patterns in online networks. There have been several works addressing this issue [18, 108, 114, 116], but they only consider direct and binary user interactions. Instead, UISE captures the frequency of both direct and indirect interactions, and therefore provides a more thorough indication of the nature of user interactions. We take Facebook as an example and we apply UISE to detect its user interaction patterns. As a result, we have derived patterns that highly resemble the ones often observed in the offline human society.

### 6.4.1 Applying to interaction pattern detection

We take *wall posts* in Facebook as the example of user interactions. In Facebook, each user can post messages on the walls of his friends. Wall posts are visible to others who visit the user profiles. A previous measurement study [108] contains a snapshot taken on Jan 22, 2009, of the entire wall post histories of 60,290 users in the New Orleans network. We select their data of the last year—from Jan 23, 2008 to Jan 22, 2009—as the user behavior imported into our experiment. In total, 44,397 users and 876,993 posts are included.

We set the cycle size to one week and we divide the data into two parts: we take the first 80% as the training data (weeks 1-43), and the latter 20% as the testing data (weeks 44-54). When a message is posted on a wall in some week, the two friends involved are considered as having directly interacted, and the corresponding bit in their interaction bitmap is set from 0 to 1. As Facebook is centralized, we can obtain the interaction bitmaps of all user pairs. For every user pair, we evaluate their user interaction strength (UIS) based on the two parts of the data, and we refer to the results as UIS80 and UIS20, respectively. We use UIS80 to demonstrate the user interaction pattern and we use the ratio between UIS20 and UIS80 (represented as  $UIS20/UIS80$ ) to demonstrate the evolution of the user interaction. The results are shown in the following section.

### 6.4.2 Results

We first show the UISs of a highly active and a medium active user with all users with whom they have interacted, either directly or indirectly. The highly active user has been active in 53 out of 54 weeks and has exchanged 2,083 messages with 25 of his friends; the medium active user has been active in 26 out of 54 weeks and has exchanged 84 messages with 8 of his friends. For each of these users (called the evaluating user) we calculate his UIS80 and UIS20 with all other users (called the evaluated users), including the friends he has interacted with directly. We group the evaluated users based on the value of their UIS80.

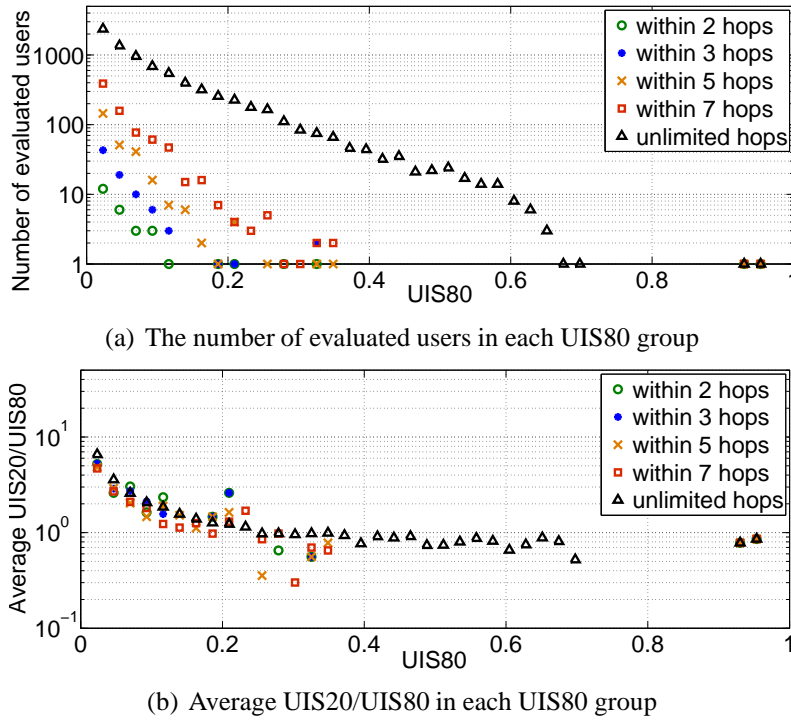
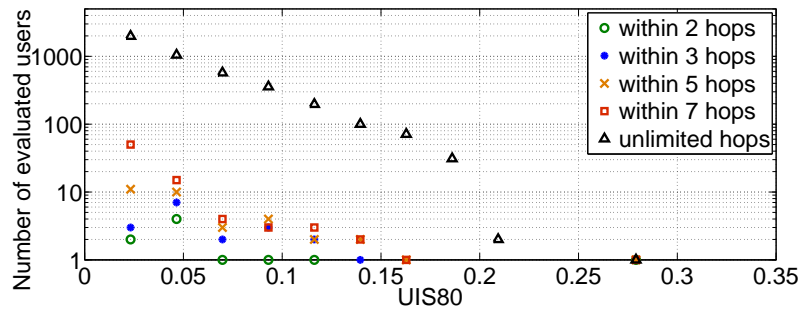


Figure 6.6: Interaction pattern of a highly active Facebook user (the vertical axes are in log-scale)

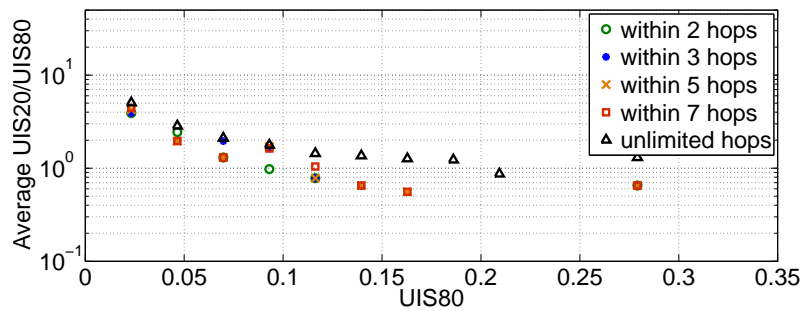
For the highly active user, we show in Fig. 6.6 the number of users and the average UIS20/UIS80 for each UIS80 group (represented by the black triangles, we do not consider a limit to the distance between interacting users here). While this evaluating user has only interacted directly with 25 friends, UISE links him to more than one thousand users through indirect interactions, among which, obviously, it interacts intensively with only a small group: as shown in Fig. 6.6(a), the number of users in the UIS80 groups decreases dramatically with increasing values of UIS80. A similar phenomenon is often observed in human society where people tend to interact frequently with relatively small groups and occasionally with the people outside those groups [67]. The small group could be friends, with whom people interact directly, or friends of friends, with whom people build bonds through, for example, sharing gossips with friends.

Another interesting observation is that, as shown in Fig. 6.6(b), the average value of UIS20/UIS80 decreases with increasing values of UIS80 until UIS80 equals 0.25, and stays stable (at a little bit less than 1) afterwards. This indicates that (1) the interactions between the evaluating user and the users with whom it has high interaction strengths in the first 80% of the year tend to stay stable, with slightly decreased interaction strengths in the latter 20% of the year; and (2) the interactions between the evaluating user and the users with whom it has low interaction strengths in the first 80% of the year tend to



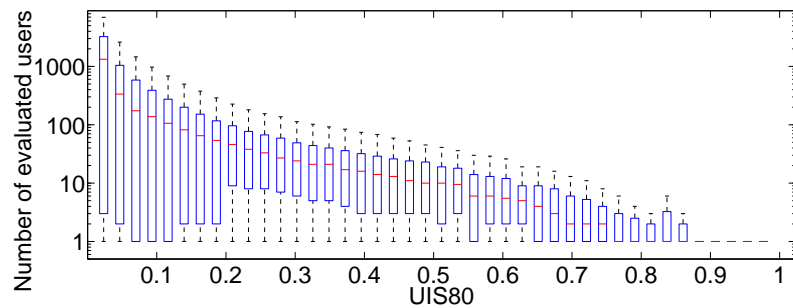


(a) The number of evaluated users in each UIS80 group

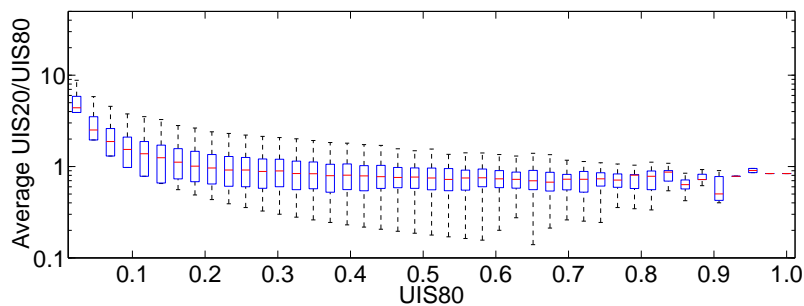


(b) Average UIS20/UIS80 in each UIS80 group

Figure 6.7: Interaction pattern of a medium active Facebook user. The vertical axes are in log-scale.



(a) Statistics of the number of evaluated users in each UIS80 group



(b) Statistics of the average UIS20/UIS80 in each UIS80 group

Figure 6.8: Interaction pattern of all Facebook users. The vertical axes are in log-scale.

become more intense. It can be conjectured that the same dynamic holds in offline social relationships.

To verify whether the above observations are due to indirect interactions linking too many strangers to the evaluating user, we have also performed tests where we consider a limited distance in terms of the number of hops between interacting users. For example, for the result of “within 2 hops” as shown in Fig. 6.6, we have only considered the evaluating user’s friends and friends of friends. We show the results for different numbers of hops, and we observe a similar tendency as for our original design without a hop limit.

For the medium active user we show in Fig. 6.7 the number of users and the average UIS20/UIS80 for each UIS80 group. In general, this user achieves a similar interaction pattern as the highly active user, except that his interaction strengths with other users are always less than 0.4—indicating that he is indeed a user with medium activity.

We have tested all users as in the above two examples and in Fig. 6.8 we show the minimum, the maximum, the median, and the 25th and 75th percentiles of the number of evaluated users and of the average value of UIS20/UIS80 of all users (there is no limit to the distance between interacting users here). We find similar user interaction patterns in these results as in the two examples we gave earlier.

## 6.5 Distributed online time estimation

In this section, we introduce another application of UISE, which is distributed online time estimation. Online time directly reflects user activity and is therefore important for online networks. A potential utilization in Facebook (especially in a distributed version) is evaluating user’s stickiness by estimating the time users spend being online. And in BitTorrent, as users upload when they download, online time implies a user’s contribution level and therefore can be used to design incentive policies [49]. The advantage of online time is that it is a metric with a ground truth—by comparing the real and the estimated online times, we can assess the accuracy of UISE. In this section, we introduce how to apply UISE to derive a decentralized scheme for online time estimation, and how to implement this application into Tribler. In Sections 6.6-6.8, we evaluate its performance by means of simulation, emulation, and real world deployment.

### 6.5.1 Applying to distributed online time estimation

Similarly as we previously took wall posts in Facebook as the example of user interactions, in this application we take *rendezvous* as the example of user interactions. By applying UISE, a user can now estimate the online time of another user through calculating their interaction strength, i.e., the frequency of their direct and indirect interactions. Here, when two users meet (*rendezvous*) in a particular cycle, they generate an interaction

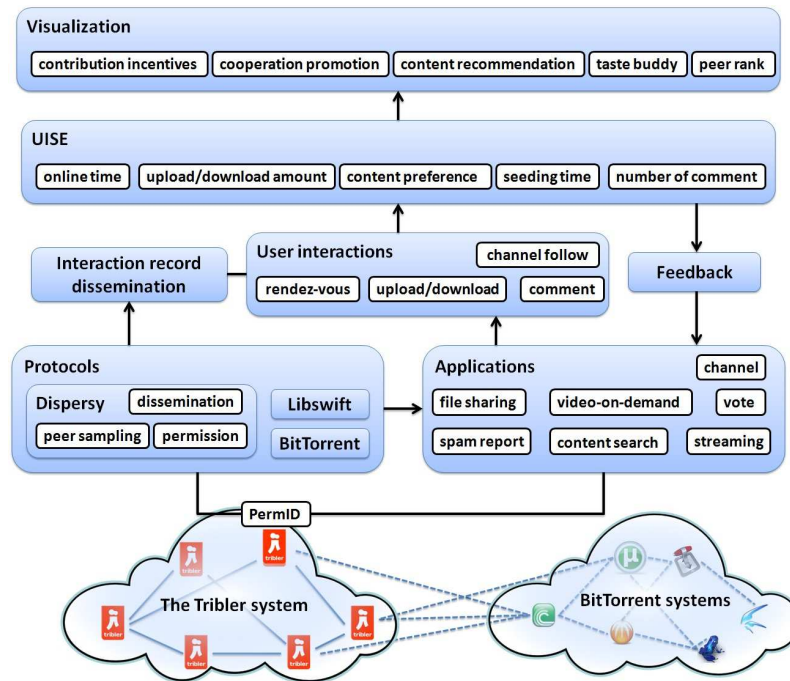


Figure 6.9: The Tribler system.

record and disseminate it into the system. Based on the records received through dissemination, a user  $i$  builds its local per-cycle UIGs, which are undirected since rendezvous is undirected, and if in any of these a user  $j$  is in the same connected component as  $i$ ,  $i$  recognizes  $j$  to be online in that cycle. The number of these recognized cycles gives  $j$ 's online time as estimated by  $i$ , and the user interaction strength between  $i$  and  $j$  computed as the number of these cycles divided by the length of the interaction history, gives  $j$ 's fraction of online time as estimated by  $i$ .

The requirement for being in the same connected component is for maintaining the security against malicious users, as in the example shown in Fig. 6.4, which generates a side effect of UISE—the accuracy of an evaluating user's estimations is limited by its own online time: as we will see, the more active a user is, the more accurate his estimations of the online times of others will be.

## 6.5.2 Implementing into Tribler

We have implemented the distributed online time estimation application with UISE into Tribler [84], which is a fully distributed open-source online network for media and social applications like file sharing, live streaming, video-on-demand, content searching, voting, and interest-based channels. Users in Tribler interact in various ways including rendezvous, upload, download, and additionally, commenting, replying, voting, and re-

porting spam in the channels they join. Tribler serves well as a general framework for implementing, testing, and analyzing user interaction related studies. It has been instrumented with monitoring capabilities to measure both system and component specific performance for design improvements. In this section we give a brief overview of the design features of Tribler relevant for this chapter.

Fig. 6.9 shows the general architecture of the Tribler system. To capture user behavior across sessions, Tribler assigns each user a permanent identifier. It uses the BitTorrent protocol for P2P file sharing, and it uses the Libswift protocol [77], which is an IETF (Internet Engineering Task Force) standard protocol for streaming proposed by the Tribler group, for P2P streaming. We have implemented UISE into Tribler as a separate component for estimating user interaction strength. The estimations can be fed back to applications for policy design, such as in file sharing to reciprocate active users with priority for future downloads; they can also be visualized in the user interface to psychologically motivate users to contribute.

Tribler uses the following protocol to discover new users and to disseminate interaction records. Every 5 seconds, a user  $i$  contacts one of its neighbors, for example,  $j$ . First,  $i$  and  $j$  generate an interaction record for this rendezvous. Secondly,  $j$  introduces one of its own neighbors to  $i$  for later contacts. Finally,  $i$  sends a Bloom filter expressing the interaction records it currently possesses and fetches the ones it is missing from  $j$ . In this way, each Tribler user can build its local UIG and estimate the online time of other users.

## 6.6 Simulation

In order to evaluate the performance of UISE, we take distributed online time estimation as the example and we address three questions: how well does UISE perform for different information dissemination protocols; how well does it perform for different user behaviors; and how well does it perform in the real world.

To answer the first question, in this section we run simulations with generic information dissemination, which allow us to explore the accuracy and scalability of UISE under different dissemination protocols by tuning the coverage. To answer the second question, in Section 6.7 we run emulations of UISE under various synthetic and real-world user behaviors. To answer the last question, in Section 6.8 we report measurement results derived from the Internet-deployed Tribler system.

### 6.6.1 Basic simulation model

**Synthetic user behavior:** We consider synthetic user behavior in our simulation. At any time, a peer<sup>1</sup> can be either online or offline. When an online session ends, it starts

---

<sup>1</sup>From here, we use *user* and *peer* alternatively to refer to the functioning agent in our experiments.

an offline session immediately, and vice versa. The online and offline session lengths follow exponential distributions, as observed in many distributed systems [36]. We do not consider population turnover in synthetic user behavior. Instead, later in Sections 6.7 and 6.8 we use user behavior generated from measurements where population turnover is naturally included.

Let  $(S_{on}, S_{off})$  represent the average online and offline session lengths. We consider two classes of peers: (i) active peers, class A, with (8, 2) cycles, and (ii) less active peers, class B, with (2, 8) cycles.

**Synthetic peer discovery and record dissemination:** In our simulation, we abstract peer discovery into a constant probability,  $P_1$ , and we apply it in every cycle to specify the probability that any two online peers meet and generate an interaction record. Similarly, we abstract record dissemination into another constant probability,  $P_2$ , and we apply it in every cycle to specify the probability of an interaction record being received by a third peer. Real-world protocols are more complicated and they normally result in dynamic probabilities. Nevertheless, by assuming a constant probability and setting different values to it, we can analyze the performance of UISE for different peer discovery and dissemination protocols.

**Simulation setup:** We run each simulation for 336 cycles, i.e., 168 hours (7 days) when one cycle represents 30 minutes in the real implementation<sup>2</sup>. Unless otherwise stated, we consider 250 peers in class A and 250 in class B, and we set  $P_1$  to 20% and  $P_2$  to 50%.

Based on the synthetic user behavior and record dissemination, each peer gradually collects interaction records and builds its local UIG. At the end of the simulation, it estimates its user interaction strength with every other peer, which, as specified in Section 6.5.1, is equal to its estimation of the fraction of online time of another peer. By comparing this estimation with the real fraction of online time, we evaluate the accuracy and scalability of UISE. The results are presented in the following sections.

## 6.6.2 Accuracy

We first show in Fig. 6.10 the comparison between the real and estimated fractions of online times, where the latter is represented by the user interaction strengths (UISs) between the evaluating and evaluated peers. In Figs. 6.10(a) and 6.10(b), the evaluated peers are ranked according to their UISs with an evaluating peer in class A and in class B, respectively. We see that the estimation is improved when the evaluating peer is more active: the peer in class A achieves more accurate estimations than the peer in class B. Peers in class B only stay online for  $S_{on}^B / (S_{on}^B + S_{off}^B) = 20\%$  of the time and therefore they meet few peers to build their local views. As stated in Section 6.5.1, this is a compromise for

<sup>2</sup>We will give the reason later in Section 6.7.3

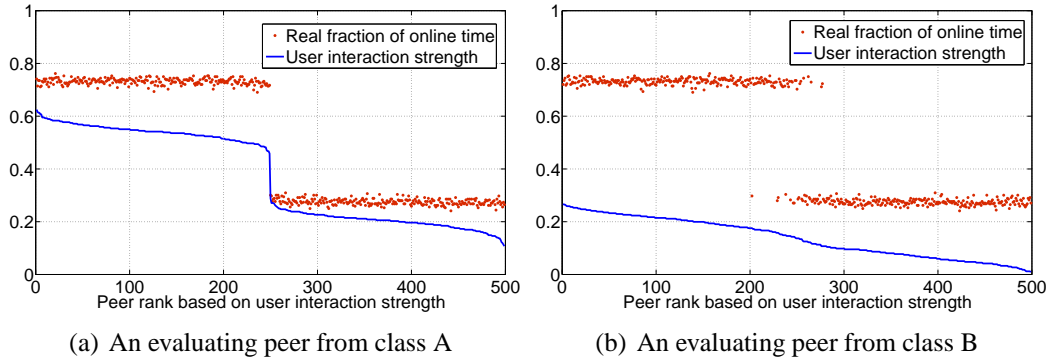


Figure 6.10: Comparison between the real and estimated fractions of online time.

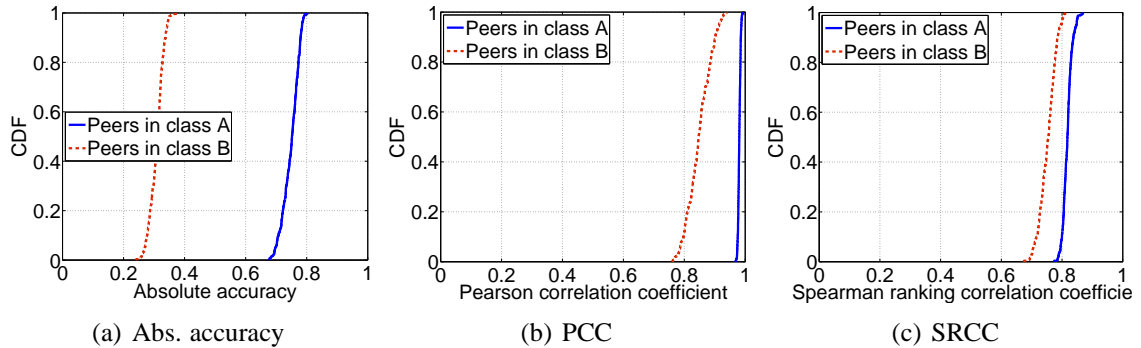


Figure 6.11: CDFs of the absolute accuracy, PCC, and SRCC in a simulation with 500 peers.

maintaining the security.

**Absolute accuracy:** We define the absolute accuracy achieved by a peer as the ratio between its estimation and the real online time of another peer, averaged over all evaluated peers. In Fig. 6.11(a) we show the CDF of the absolute accuracies achieved by peers in class A and in class B, respectively. In general, peers in class A achieve better absolute accuracies than peers in class B.

**Relative accuracy:** We use the Pearson Correlation Coefficient (PCC) [94] and the Spearman Ranking Correlation Coefficient (SRCC) [99] to assess the relative accuracy achieved by each peer. In brief, PCC and SRCC measure the linear and the monotonic dependence between two variables, respectively. For each evaluating peer, first, we rank its evaluated peers based on its online-time estimations for them; then, we generate two variables, a list of the real online times and a list of the online-time estimations, in the order of this ranking; finally, we calculate the PCC and the SRCC of these two variables. In this way, we can assess the correlation between the local rank of peers at the evaluating peer and the global rank of peers based on their real online times. For the two examples shown in Figs. 6.10(a) and 6.10(b), the PCCs are 0.9848 and 0.9026, and the SRCCs are

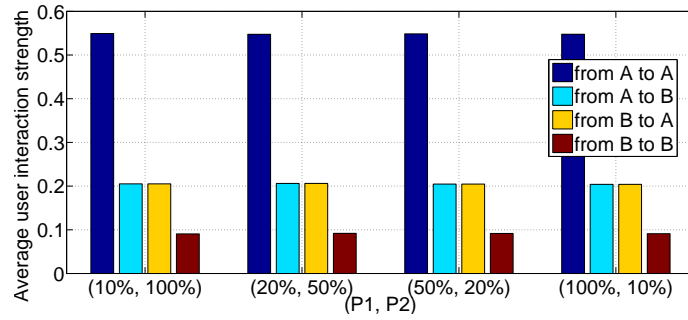


Figure 6.12: Average user interaction strengths for different values of  $P_1$  and  $P_2$ .

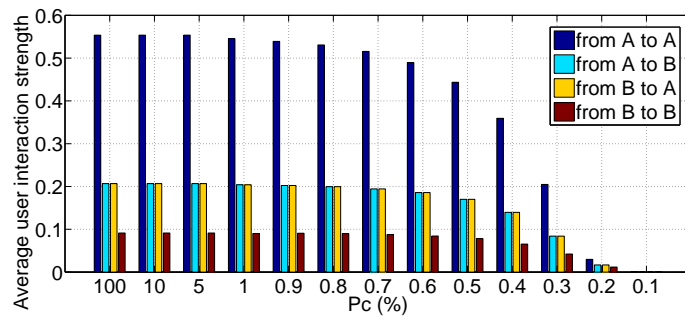


Figure 6.13: Average user interaction strengths for different values of  $P_C$ .

0.7968 and 0.7812, respectively. Figs. 6.11(b) and 6.11(c) show the CDFs of the PCC and the SRCC achieved by peers in class A and in class B. We see that peers achieve decent relative accuracies, i.e., their local ranks of peers resemble the global one.

**Absolute accuracy versus relative accuracy:** One important application of online time estimation is to differentiate users with different levels of activity. Then, only the rank of users is needed and a design with high relative accuracy, like UISE, will be suitable. Further, as we will show in the next section, UISE gives accurate estimations even under low coverages of peer discovery and information dissemination.

### 6.6.3 Accuracy under partial information

To test the accuracy of UISE under partial information, we first vary  $P_1$  and  $P_2$  in such a way that  $P_C = P_1 \times P_2$  is constant (equal to 10%).  $P_C$  represents the probability of establishing an edge between two peers in a local per-cycle UIG of a third peer. Intuitively, it decides the third peer's estimations for others. In Fig. 6.12 we show the user interaction strengths (estimated fractions of online time) averaged over the classes of evaluating (“from” in the figure) and evaluated (“to” in the figure) peers. We find that, consistent with our intuition, the estimations stay stable for different values of  $P_1$  and of  $P_2$  while  $P_C$  is constant. This allows us to analyze the influence of  $P_C$  without exploring exten-

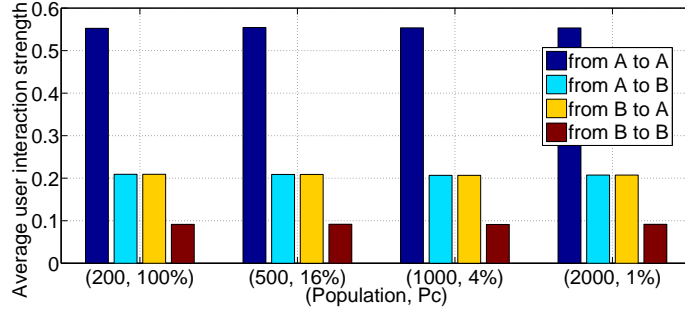


Figure 6.14: Average user interaction strengths for different populations and values of  $P_C$ .

sive combinations of  $P_1$  and  $P_2$ . We keep  $P_1$  constant (equal to 1) and vary  $P_2$  in such a way that  $P_C$  is decreased from 100% to 0.1%. In Fig. 6.13 we show the user interaction strengths averaged over classes. We find that when  $P_C$  is at least equal to 5%, i.e., when peers hold at least 5% of the total information, the estimations stay stable. Further reducing  $P_C$  results in noticeable decreases of estimations, nevertheless, UISE still achieves decent relative accuracies: on average, peers in class A are correctly estimated as more active than peers in class B.

### 6.6.4 Scalability

In this section we test the scalability of UISE under different populations. Let  $N(t)$  represent the number of online peers at cycle  $t$ . At the end of the simulation, for cycle  $t$ , each peer will receive  $N(t)(N(t) - 1)P_C$  records and will generate  $N(t)(N(t) - 1)P_C$  edges in its local per-cycle UIG. To test the scalability, when the population increases, we decrease  $P_C$  in such a way that  $N(t)(N(t) - 1)P_C$  is constant.

In UISE, an evaluating peer recognizes another peer to be online at cycle  $t$  if they are in the same connected component in its local per-cycle UIG for cycle  $t$ . In graph theory, for a random graph with  $n$  vertices to be connected, the expected number of edges needed is less than  $n \ln n$  [74]. Therefore, the basic condition for a peer to correctly recognize all the peers online at cycle  $t$  is:

$$N(t)(N(t) - 1)P_C \geq N(t) \ln(N(t)) \Rightarrow P_C \geq \frac{\ln(N(t))}{N(t) - 1}.$$

As  $\ln(N(t))$  increases very slowly with  $N(t)$ , the required value for  $P_C$  decreases strongly with the population. Thus, UISE achieves good scalability in a self-organized manner.

The simulation result confirms the above analysis. Fig. 6.14 shows the average user interaction strengths (estimated fractions of online time). We see that while we increase the population and decrease  $P_C$  accordingly, UISE achieves stable estimations.



## 6.7 Emulation

In this section we evaluate the performance of UISE for different user behaviors. As we have tested different dissemination protocols in Section 6.6, we now release the assumption of a generic information dissemination and we test UISE under Tribler’s dissemination protocol. As we intend to test different user behaviors that we cannot control in real world experiments, we use emulation in this section. Based on synthetic user behaviors, we test the performance of UISE for different online patterns. Based on user behaviors generated from measurement traces, we test its performance across different online networks.

### 6.7.1 Emulation setup

The emulation is performed on the DAS-4 supercomputer [24], a distributed six-cluster system for computer science research. All our experiments are run on the TUDelft cluster that contains 23 nodes, each of which has two 2.4 Ghz quad-core processors and 24 GBytes of memory. The nodes are connected by a 10 Gb/s QDR Infiniband interconnect. Unless otherwise stated, we deploy 500 peers evenly on 20 nodes. Peers run Tribler’s dissemination protocol: they meet, generate, and disseminate interaction records, and store the records received from dissemination in their local SQLite databases. At the end of each emulation, they estimate their user interaction strengths with others, which give their estimations of the fractions of online time of others.

We set the cycle size to 2 minutes and we run each emulation for 10 hours, resulting in interaction bitmaps of  $10 \times 60/2 = 300$  bits. The small cycle size and short emulation time are compromises for the time consumption of the cluster. This parameter setting represents a running time of 7 days when the cycle size is set to 30 minutes in the real world implementation.

### 6.7.2 Synthetic user behavior

In this section we evaluate the performance of UISE under different mixtures of active peers, less active peers, peers with heavy churn, and peers that always stay online. Here, we use the synthetic user behaviors as introduced in Section 6.6.1 and we test four scenarios in our emulation: (i) peers always online with  $A(\infty, 0)$  and  $B(\infty, 0)$ ; (ii) active peers  $A(20, 5)$  versus less active peers  $B(5, 20)$ ; (iii) active peers  $A(20, 5)$  versus heavy churn peers  $B(5, 5)$ ; and (iv) heavy churn peers with  $A(5, 5)$  and  $B(5, 5)$ . All the session lengths are in minutes. The results are shown in Fig. 6.15.

Under scenario 1, peers have the highest chance to meet and hence generate a large number of records. Fig. 6.15 shows that UISE achieves accurate estimations under this scenario: on average, 93.33% of the online time is successfully identified. Under scenario

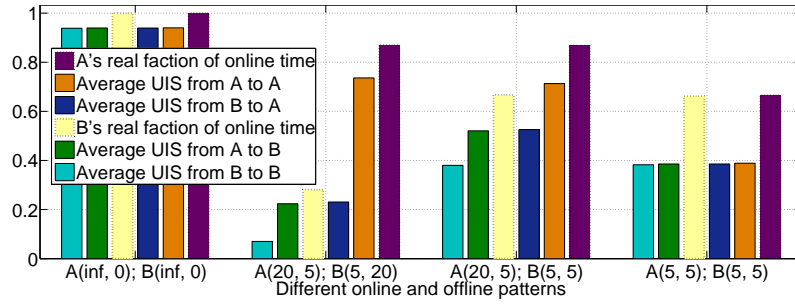


Figure 6.15: Comparison between real and estimated fractions of online time for different churn patterns ( $S_{on}$ ,  $S_{off}$ ).

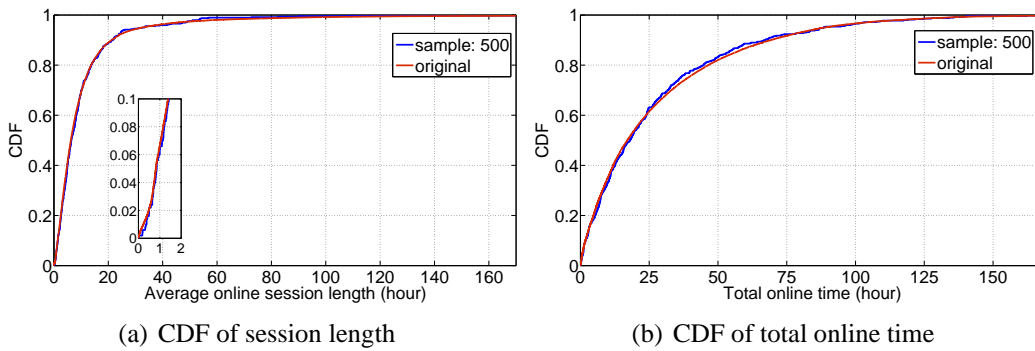


Figure 6.16: Comparison between the original FileList trace and our sample.

2, the active peers achieve higher accuracies than the less active peers. Interestingly, comparing scenarios 2 and 3, when the less active peers are changed to peers with heavy churn, their estimations for active users (“from B to A” in the figure) become more accurate. Under scenario 4, though peers are with heavy churn, they can still identify  $0.4/0.6 = 67\%$  of the online time. These results indicate that the absolute accuracy depends on the *availability*, i.e., the fraction of online time, of the evaluating peer, rather than its churn pattern.

### 6.7.3 Trace-based user behavior

To test the performance of UISE across different online networks, we run emulations based on measurement traces generated from the private BitTorrent community FileList [6]. These traces contain uptime and downtime of every user that was online at least once during the measurement period. In total, we captured 63,548 users in 7 days, from which we randomly select 500 users for our emulations. Fig. 6.16 shows the CDFs of the average online session length and the total online time for the original trace and our sample. We see that our sample represents the original trace very well. Further, as 94% of the online sessions are longer than one hour, we set the cycle size to 30 minutes in the real implementation, so as to capture most of the online sessions.

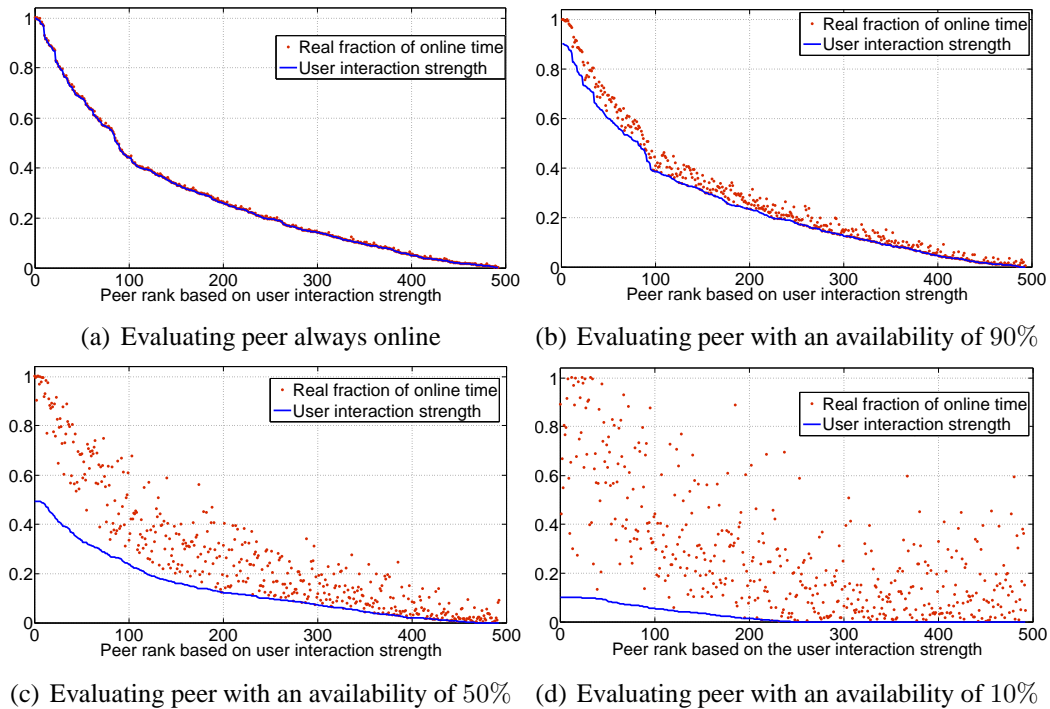


Figure 6.17: Comparison between real and estimated fractions of online time for FileList trace.

Fig. 6.17 shows four examples of comparisons between the real and estimated fractions of online time, where the latter one is represented by the user interaction strengths between the evaluated and evaluating peers. In Figs. 6.17(a), 6.17(b), 6.17(c), and 6.17(d), the evaluated peers are ranked based on their online times estimated by a peer with an availability of 100%, 90%, 50%, and 10%, respectively. The first three peers achieve estimations very close to the real fractions of online time, with an SRCC equal to 0.9998, 0.9945, and 0.9388, respectively. We can see a clear decrease of the real fractions of online time when the evaluated peers are ranked based on the estimations from these three evaluating users, indicating that their local ranks of peers closely resemble the global one.

Evaluating peer 4 (Fig. 6.17(d)), however, only achieves an SRCC equal to 0.6853. The reason is that, as stated in Section 6.5.1, in UISE an evaluating peer only trusts the interaction records that can link back to itself (reflected by being in the same connected component in a local UIG). Therefore, its estimation for another user in fact reflects their *concurrent* online time. As evaluating peer 4 is only online for 10% of the time, it achieves a low accuracy. Nevertheless, its estimation for another user can be used to assess their availability to each other—an important issue for distributed online networks where users collaborate and only the ones online simultaneously can help each other. In Fig. 6.18 we show evaluating peer 4’s estimations and its fraction of concurrent online time with other peers, where we observe very accurate estimations with an SRCC equal to 0.9973.

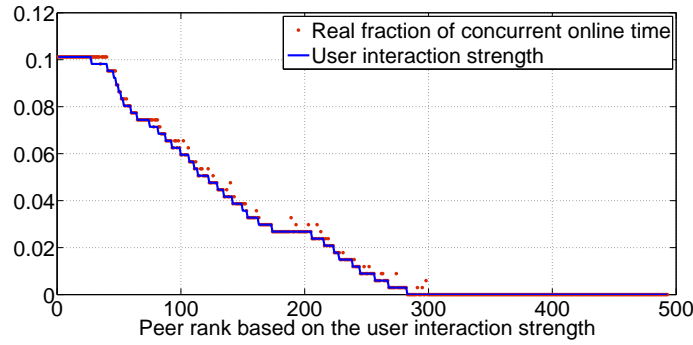


Figure 6.18: Comparison between fraction of concurrent online time and user interaction strength.

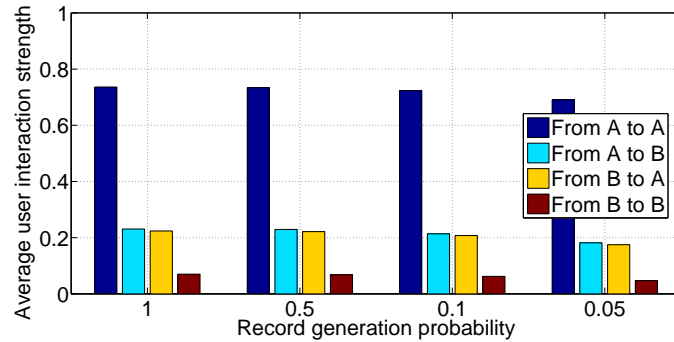


Figure 6.19: User interaction strength under random-generation for different values of  $P_G$ .

#### 6.7.4 Preparing for the real world: strategies for reducing the workload

As we found in Section 6.6.3, UISE achieves good estimations even with limited information. This desirable feature allows us to introduce two practical strategies, *random-generation* and *targeted-generation*, to reduce the workload imposed on the system, in terms of the number of interaction records to be disseminated. In random-generation, when two online peers meet for the first time in some cycle, with a probability  $P_G$  they generate an interaction record (in the original design they will for sure generate a record). In targeted-generation, for each cycle, a user only generates records with the  $N_G$  users that it has observed to be online the longest during the past  $M$  cycles, resulting a constant number of records being generated per user per cycle.

We run emulations to test the performance of these two strategies, where we use the same parameter settings as in Section 6.7.2. Figs. 6.19 and 6.20 show the user interaction strengths (estimated fractions of online time) averaged over classes for different values of  $P_G$  and  $N_G$  (we set  $M = 1$  in our emulations). We see that UISE performs stably

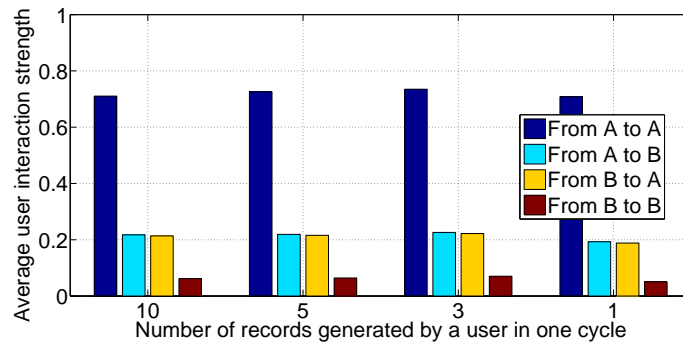


Figure 6.20: User interaction strength under targeted-generation for different values of  $N_G$ .

when  $P_G$  is decreased from 1 to 0.05 and when  $N$  decreases from 10 to 1. This implies that we can decrease the workload dramatically without deteriorating the accuracy of the estimation.

## 6.8 Real-word deployment

We have implemented UISE into Tribler. In addition to its qualitative impact on design, Tribler’s user community serves as the basis for our real-world performance evaluation. In this section, we report measurements of the user behavior, the overlay structure, the performance of interaction record generation, and the accuracy of online time estimation.

### 6.8.1 Deployment techniques

In the real world deployment, the cycle size is set to 30 minutes and the interaction history is kept in a sliding window fashion with a window size of 7 days, resulting in interaction bitmaps of  $7 \times 24 \times 60/30 = 336$  bits. We adopt the *targeted-generation* version of UISE as introduced in Section 6.7.4, with  $N_G = 5$  and  $M = 1$ . In addition, we specify that two users generate their first interaction record only until they have seen each other online for at least two cycles. This effectively prevents “hit-and-run” users generating records that are of limited use.

Tribler is fully distributed, containing no central servers and hence no records of user behaviors from the global view. To obtain the ground truth for our experiment, we deploy log servers and every 5 minutes, each user reports its online activity to one of them, including its identifier, its timestamp, the number of interaction records it generated successfully, and the updated information about interaction records of other user pairs it received since last report. In total, for the first week of Tribler’s new release, we obtain 2,874 active users with unique identifiers, among which there are 1,713 users that have

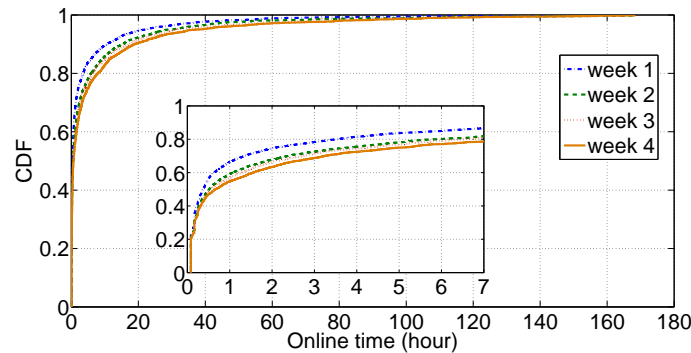


Figure 6.21: CDF of online time of Tribler users.

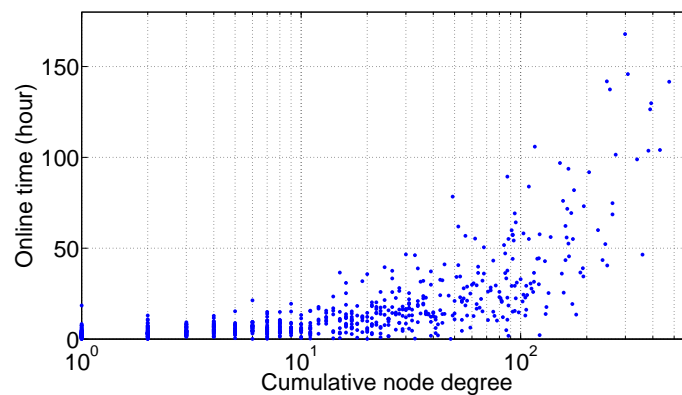


Figure 6.22: Online time versus cumulative node degree.

generated at least one interaction record. For weeks 2, 3, and 4, we obtain 2,673, 2,905, and 2,884 active users with unique identifiers, respectively. We use the data from weeks 1-4 to analyze the evolution of Tribler user's online time; and we use the data of week 1 to evaluate the overlay structure, the performance of record generation, and the accuracy of online time estimation. Results are shown in the following sections.

## 6.8.2 Evaluation

**Online time:** The dashed blue line in Fig. 6.21 shows the CDF of online times of Tribler users obtained from log servers during the first week. Around 15% users are online for more than 7 hours, resulting in an average of more than one hour per day. Nevertheless, 60% users are online for less than one hour in total. Comparing Figs. 6.21 and 6.16(b), clearly users in FileList are more active than users in Tribler. FileList specifies that users with high contribution levels will be rewarded with the preference for future downloads, and therefore users are incentivized to stay online longer. To do so, FileList constantly monitors user activities through central servers—UISE achieves exactly the same goal,

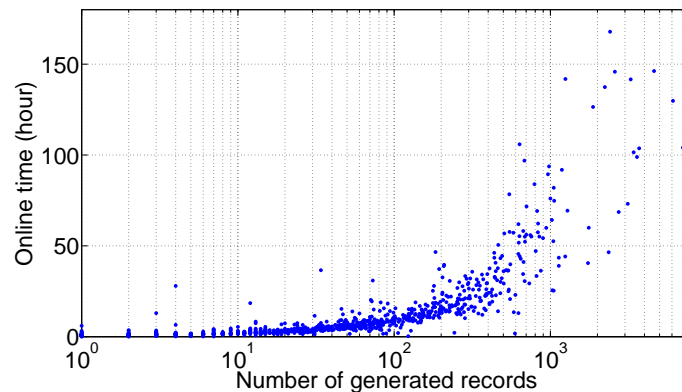


Figure 6.23: Online time versus the number of generated records.

and moreover, is performed in a distributed manner. As a matter of fact, in the current release of Tribler, users are educated with the fact that their activities will be evaluated through a distributed algorithm (i.e., UISE). Though at this moment the estimated online times are not utilized explicitly as in FileList, we can already observe a gradual increase of user's online time from week 1 to week 4 (Fig. 6.21). This promising observation indicates that, being aware of their activities being evaluated, users in Tribler are becoming more committed—a behavioral change that has been observed by many sociologist and psychologists under similar circumstances in human society [44].

**Overlay structure:** When users are online, they gradually meet more users and generate more interaction records. This effect is further amplified by targeted-generation where users that stay online longer are preferred by other users to generate interaction records with. In Fig. 6.22 we show the scatter plot of each user's online time and its cumulative node degree, which is defined as the number of unique users that it has generated interaction records with. We see a clear positive monotony trend between them, achieving an SRCC of 0.7883. In Fig. 6.23 we show the scatter plot of each user's online time and the number of interaction records it generated successfully, where we observe a positive monotony trend with an SRCC of 0.9312.

**Interaction record generation:** In Fig. 6.24 we show the CDF of record generation success rate, which is defined as the ratio between the number of generated records and the number of record generation attempts, for each user. We see that in general, 80% of the users achieve success rates larger than 0.7. The failures may come from several circumstances including the churn of users, the recipients being saturated by requests, and the NATs between users that prevent them to connect. One may argue that with the preference to generate records with highly active users, those users can be fully occupied and therefore resulting in low record generation success rates. Nevertheless, we show in Fig. 6.25 the scatter plot of each user's online time and its record generation success rate. The corresponding SRCC is only -0.1591 and therefore shows no correlation be-

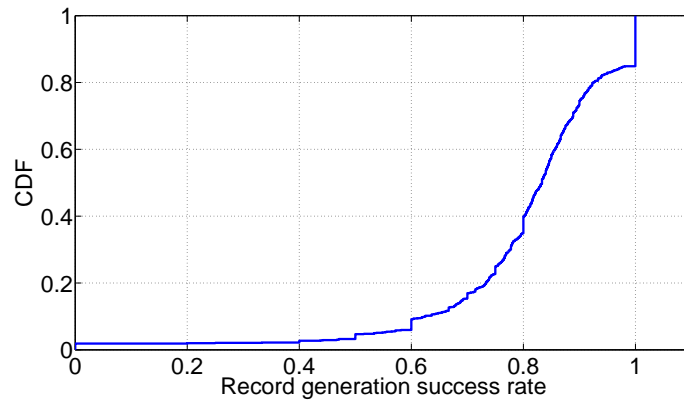


Figure 6.24: CDF of record generation success rate.

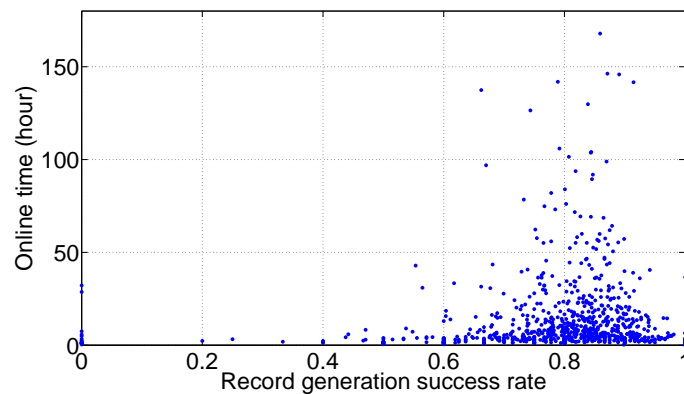
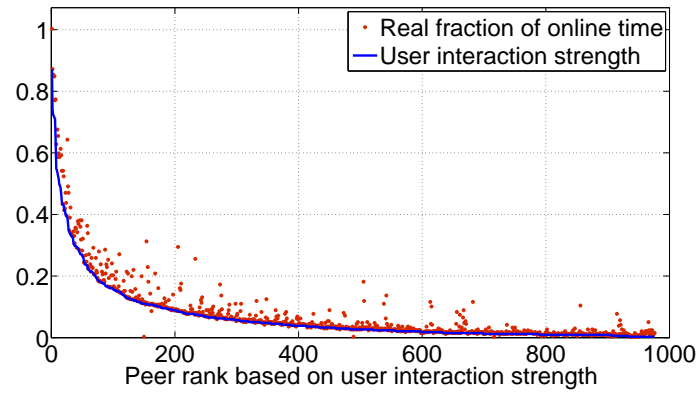


Figure 6.25: Online time versus record generation success rate.

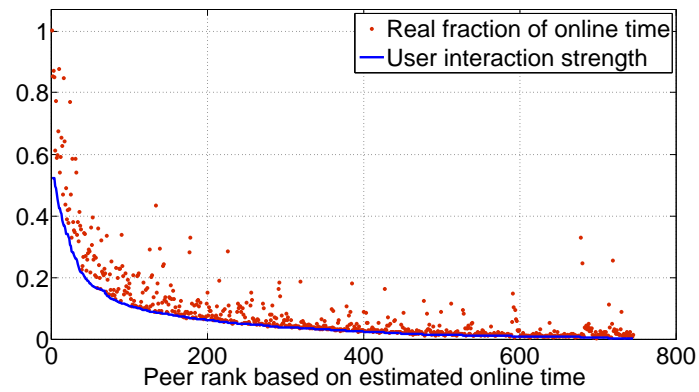
tween these two metrics. Particularly, a highly active user that stays online for 151 hours generates 2,410 records, achieving a success rate of 85.92%.

**The accuracy:** Fig. 6.26 shows the comparisons between the real and estimated fractions of online time of each Tribler user, where the latter is represented by the user interaction strengths between the evaluating and the evaluated users. In Figs. 6.26(a), 6.26(b), and 6.26(c), the evaluated users are ranked based on their user interaction strengths with a user with an availability of 90%, 50%, and 10%, respectively. Similar to the results of Filelist trace relay in Section 6.7.3, the accuracy of the estimation is limited by the availability of the evaluating user: the more active it is, the more accurate its estimations will be. In total, the three evaluating users have successfully identified 976, 745, and 217 users to be online for at least one cycle; and the SRCCs between their estimations and real online times are equal to 0.9325, 0.8635, and 0.6446, respectively. Though evaluating user 3 (Fig. 6.26(c)) achieves a low accuracy, it can accurately estimate its concurrent online time with other peers. In Fig. 6.27 we show its estimations (represented by user interaction strengths) and its real fraction of concurrent online time with others, where we

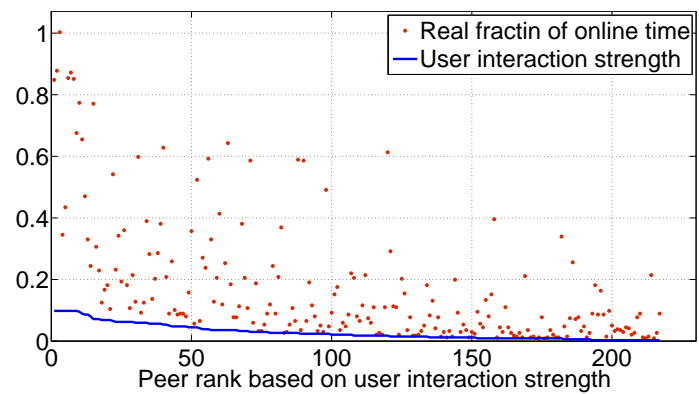




(a) Evaluating user with an availability of 90%



(b) Evaluating user with an availability of 50%



(c) Evaluating user with an availability of 10%

Figure 6.26: Comparison between real and estimated fractions of online time.

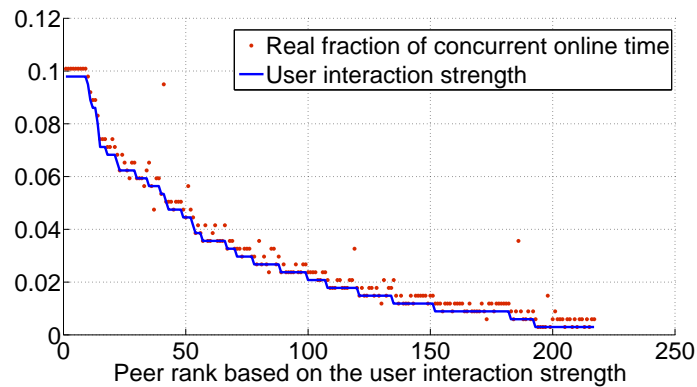


Figure 6.27: Fraction of concurrent online time versus user interaction strength.

observe very accurate estimations with an SRCC equal to 0.9785. Thus, it can successfully identify the users with whom it is online simultaneously, i.e., the users with whom it has been and potentially will be collaborating.

## 6.9 Related work

To date, a few works have focused on understanding user interactions in online social networks. Moon *et al.* [18] investigate the guestbook logs of Cyworld and they show that interactions between friends are highly reciprocated. Viswanath *et al.* [108] study the evolution of user interactions in Facebook and they find that user interactions change rapidly over time. These observations provide the foundation for designing UISE that only considers recent user interactions. Wilson *et al.* [114] introduce an interaction graph. They show that interaction links exhibit different properties than social links (friendships) and are more representative for inferring meaningful user relationships. Nevertheless, their interaction graph is unweighted and does not take the interaction frequency into account as we do.

Another direction of related research is identifying social ties. Kahanda *et al.* [54] propose an approach for identifying the weak and the strong ties. They focus on supervised learning models that require human annotation of link strength such as top friend nomination. Xiang *et al.* [116] develop an unsupervised model that represents a range of tie strengths based on user interactions and profile similarity. However, they consider only direct and binary interactions, simply indicating whether a user has interacted with another user or not. Instead, we propose UISE that captures the frequency of both direct and indirect interactions. Moreover, while all the above related works are centralized, UISE is applicable in distributed systems.

There are also studies on leveraging user interactions in distributed online networks for policy design. BitTorrent [19] clients constantly monitor their direct interactions (up-

loads) with others and reciprocate the ones from whom they download the fastest. However, in BitTorrent systems the problem of low rendezvous exits and direct interactions are insufficient for inferring user relationships [49]. Meulpolder *et al.* introduce BarterCast, a distributed reputation system that ranks users based on their upload and download activity in P2P file sharing. BarterCast captures both direct and indirect user interactions, however, it adopts a MaxFlow-based algorithm with a heavy complexity. Instead, UISE adopts an connect-component-based algorithm and achieves a linear time complexity in terms of the number of user pairs. We have also applied UISE to derive a decentralized scheme for online time estimation. To the best of our knowledge, this is the first work that sheds lights on this topic.

## 6.10 Conclusion

User interaction is the most important underpinning of online networks, in which hundreds of millions of users communicate, interact, and share their online lives. In this chapter we propose UISE, a scalable scheme for estimating user interaction strength in both centralized and distributed online networks. We have applied UISE to detect user interaction patterns in Facebook based on wall posts, and we have derived patterns that resemble the ones often observed in the offline human society. We have further applied UISE to design and deploy a decentralized scheme for online time estimation based on rendezvous. In the latter application we shown that UISE is scalable and stable for different dissemination protocols and for different user behaviors. We have incorporated this application into Tribler, and we have shown through measurements that UISE effectively differentiates users with different levels of activity, and thus, accomplishes the most important goal of estimating user interaction strength.



# Chapter 7

## Conclusion

In this thesis we have explored the correlation between user behaviors and contribution incentives in online networks. In order to gain a systematic insight, we have investigated both barter schemes and monetary schemes for contribution incentives. We have examined their limitations and risks, we have proposed remedies to revise them, and we have proposed a general framework for estimating user interaction strength that underpins a wide range of systems and applications including contribution incentives. Below we present our conclusions and suggestions for future work.

### 7.1 Conclusions

The main conclusions of this thesis are as follows:

1. Enhancing reciprocity or reducing inequality is a key principle in the design of BitTorrent and its variations. Different applications often require different principles, and therefore the balance between reciprocity and equality is essential. Several factors in the BitTorrent protocol have significant influences on this balance. For example, reciprocity is enhanced when users assign more bandwidth to their *regular unchoke slots* and when seeders favor the fast users. On the other hand, inequality is reduced when users assign more bandwidth to their *optimistic unchoke slots* and when seeders favor the relatively slow users. Overall, reducing inequality leads to a better system-wide performance in terms of a higher download speed averaged over all users (Chapter 2).
2. Sharing Ratio Enforcement (SRE), prevalently adopted in private BitTorrent communities, has been demonstrated through measurements to be effective in incentivizing contributions. Based on a fluid model, we have theoretically proven its effectiveness. By assuming that users are rational and that they only upload the minimum amount required by SRE, we have analyzed two typical scenarios where

SRE is applied within one swarm and within multiple swarms, respectively. As it turns out, the former case induces enhanced reciprocity, whereas the latter case improves the system-wide performance. Above all, our model provides a lower bound of the download performance that would be achieved in the real world (Chapter 3).

3. Due to a number of limitations induced by barter schemes, monetary schemes can be used as effective alternatives for contribution incentives. We have provided a long-missing risk analysis of these monetary schemes. As it turns out, under these schemes, the supply and the demand of contributions need a delicate balance, without which the system is prone to systemic risk that may cause the collapse of the entire system due to too little or, counter-intuitively, too much contributions. Even when these extreme situations do not occur, the user-level performance can be deteriorated due to an oversupply or an undersupply of contributions. To tackle these problems, we propose four strategies that have been inspired by ideas in social sciences and economics. The primary idea behind them is to adaptively adjust the supply and the demand based on the credit dynamics in the system. As the simulation results show, all four strategies are very effective in alleviating the negative effects of monetary schemes while still providing strong incentives for contributions (Chapter 4).
4. In online networks users do not always behave and contribution incentives do not always work well. Taking private BitTorrent communities as the example, we have explored the behaviors of their users and have found that user's dedication is not universal. As users evolve in the community, some of them become more committed, in terms of higher ratios of the seeding and the leeching time, and some game the system by keeping risky low sharing ratios while they leech more often than they seed. Meanwhile, even though swarms in private communities are greatly oversupplied, users achieve very high download speeds at significant expense including excessively long seeding times and very low upload speeds. For users who intend to increase their sharing ratios, we have shown that seeding for longer durations is not as effective as increasing the upload speed. If it is not realistic for the users to upgrade their internet access, we suggest them to join swarms early or to join undersupplied swarms (Chapter 5).
5. With the User Interaction Strength Estimation (UISE) scheme that we have presented in this thesis, we have devised a general framework for expressing user interactions and their strengths that is both generic and can be applied to a wide range of online networks. Among the strong points of UISE is that it captures direct and indirect user interactions, that it scales with only partial information dissemination in decentralized systems, and that it provides disincentives for malicious user be-

haviors. We have applied UISE to detect user interaction patterns based on wall posts in Facebook and we have found patterns that resemble those observed in the offline human society: users in Facebook tend to interact frequently and stably with relatively small groups, occasionally with persons outside those groups, and they make new friends while in the meantime losing touch with some old friends. We have further applied UISE to online time estimation based on rendezvous as user interactions in Tribler, and we have demonstrated the accuracy and scalability of UISE with different information dissemination protocols and user behaviors using simulations, emulations, and a real-world deployment. Particularly, we have observed a desired user behavioral change in the real-world deployment: with the awareness that their interactions are being monitored, users are becoming more committed. This observation indicates that UISE is effective in, among other applications, incentivizing contributions (Chapter 6).

## 7.2 Suggestions for future work

There are a number of promising directions for future work related to the topic of this thesis, including:

1. Our model for BitTorrent is based on its original Tit-For-Tat. Since its first release, several strategic variations of BitTorrent have been proposed, for example BitTyrant [78] and BitThief [65], which aim at maximizing their download speeds while minimizing their contributions. It will be interesting to apply our model to these strategic variations and to investigate their effects in bandwidth allocation, in reciprocity enhancement, and in inequality reduction.
2. We have performed a measurement of user behaviors in private BitTorrent communities that adopt Sharing Ratio Enforcement (SRE). There are some private communities that apply further policies to enhance the positive effects and to alleviate the negative effects of SRE, for example, the occasional *free-leech* and *free-seed*. A more detailed measurement of user behavioral changes under these special scenarios will help community organizers to understand the possible manipulations from users, which in turn will lead to better policy designs.
3. We have used models and simulations to provide a straightforward demonstration of the effectiveness and risks of monetary schemes for contribution incentives. In this analysis, we have only considered static behaviors. A more realistic approach will be including dynamic behaviors where users can change their strategies given the current situation of the system. For example, a game theory approach, as the one

often adopted in analyzing scrip systems [56], will provide a more theoretical foundation and therefore more insights into the understanding of monetary schemes.

4. The UISE scheme we propose is a general framework for representing user relationships with their interactions. It will be promising to combine it with social science studies that evaluate how well online interaction represents real world relationships.

### 7.3 Epilogue

For decades, sociologists and psychologists have been putting effort in analyzing human behaviors, and particularly, in analyzing policies to incentivize humans to contribute and cooperate. However, traditional offline approaches such as surveys and controlled experiments suffer from the common flaw that the data derived from these approaches were comparably scarce and often of poor quality [61, 113]. The needed quantity and quality of data on human societies was simply impossible to obtain. Coincidentally or naturally, online networks arose, together with the societies formed within them.

With the similarities between online and offline societies, researchers now are able to collect behavior data and to evaluate their theories at a scale that can never be reached by traditional offline methods. This is what we have done in this thesis. For example, in this thesis we have, from a computer scientist's point of view, explored the *Tragedy of the Commons* and the strategies to avoid this tragedy in the context of barter-based BitTorrent with millions of users, we have evaluated credit dynamics and systemic risk of monetary systems embedded in some P2P systems in which credit is earned and spent through contributions and consumptions, we have demonstrated on a user base of over ten thousands that one single incentive policy can trigger a variety of user behaviors in the same online community, and with the detailed logs of user behaviors in Facebook and Tribler, we have analyzed user interaction patterns and we have inferred possible user relationships.

From the large user base and the detailed records of user behavior that are made possible only by online networks, we have obtained several interesting and important insights in user behaviors and contributions incentives. Above all is that users do not always behave and contribution incentives do not always incentivize users to contribute. As in the real world, online users can be lazy, selfish, strategic, or even malicious. Moreover, adaption is hardwired in human's nature and this applies to online users as well. When a certain incentive policy is designed and applied, users learn and adapt fast, and devise new strategies to exploit the policy. Thus, there is hardly any static incentive policy that is effective for all systems at all times. Instead, our analysis shows that policies that can adapt to the context are preferable.

Another major insight is the universal tradeoff in system and policy design. Barter-



based incentive policies have their limitations in indirect reciprocity, but remedies dealing with these limitations will introduce security concerns. Monetary schemes are effective in increasing the supply of contribution, but sometimes they may lead to a state of oversupply that eventually crashes the entire system. In the context of estimating user interaction strength, excluding all indirect interactions surely filters out a large amount of useful information, but considering blindly all interactions including even the very remote ones adds a lot of noise in the analysis. Thus, finding the critical point in the tradeoffs is crucial for successful system and policy design, and this often requires a thorough understanding of the system that can only be obtained through parameter exploration, design space analysis, and user behavior modeling.

Above all, the human society has evolved for thousands of years, but there are still many areas of human behavior left unknown or unexplored. In that sense, online societies that emerged only two decades ago are still in a primitive state, yet with their ever-improving technologies we have already obtained many exciting results. This points the way to a promising future for the study of online networks, not only in analyzing online behaviors, but also in cross reference with offline societies.



## Bibliography

- [1] A. Abdul-Rahman and S. Hailes. A distributed trust model. In *New Security Paradigms Workshop (NSPW'97)*, 1997.
- [2] E. Adar and B. A. Huberman. Free riding on Gnutella. *First Monday*, September 2000.
- [3] M. Albert. *Parecon: Life After Capitalism*. Verso, 2003.
- [4] K. Anagnostakis, F. Harmantzis, S. Ioannidis, and M. Zghaibeh. On the impact of practical P2P incentive mechanisms on user behavior. In *the 2nd Workshop on the Economics of Networks, Systems, and Computation (NetEcon'07)*, 2007.
- [5] N. Andrade, E. Santos-Neto, F. Brasileiro, and M. Ripeanu. Resource demand and supply in BitTorrent content-sharing communities. *Computer Networks*, 53(4):515-527, 2008.
- [6] P2P Trace Archieve. <http://p2pta.ewi.tudelft.nl/pmwiki/>.
- [7] D. Besanko, R. Braeutigam, R.R. Braeutigam, and J. Michael. *Microeconomics*. Wiley, 2010.
- [8] Bitcoin. P2P virtual currency. <http://www.bitcoin.org>.
- [9] BitHQ. <http://www.bithq.org/>.
- [10] BitSoup. <http://bitsoup.org>.
- [11] M. Capotă, N. Andrade, T. Vinkó, F.R. Santos, J.A. Pouwelse, and D.H.J. Epema. Inter-swarm resource allocation in BitTorrent communities. In *Proceedings of the 11th IEEE International Conference on Peer-to-Peer Computing (P2P'11)*, 2011.
- [12] CHDBits. <http://chdbits.org/>.
- [13] X. Chen and X. Chu. Measurements, analysis and modeling of private trackers. In *Proceedings of the 10th IEEE International Conference on Peer-to-Peer Computing (P2P'10)*, 2010.

- [14] X. Chen, X.W. Chu, and Z. Li. Improving sustainability of private P2P communities. In *Proceeding of the International Conference on Computer Communications and Networks (ICCCN'11)*, 2011.
- [15] ChinaHDTV. <http://www.chinahdvtv.org>.
- [16] A.L.H. Chow, L. Golubchik, and V. Nisra. BitTorrent: An extensible heterogeneous model. In *Proceedings of the 29th IEEE International Conference on Computer Communications (INFOCOM'09)*, 2009.
- [17] B. Chun, P. Buonadonna, A. AuYung, C. Ng, D. Parkes, J. Schneidman, A. Snoeren, and A. Vahdat. Mirage: A microeconomic resource allocation system for sensor network testbeds. In *Proceedings of the IEEE Workshop on Embedded Networked Sensors*, 2005.
- [18] H. Chun, H. Kwak, Y.H. Eom, Y.Y. Ahn, S. Moon, and H. Jeong. Comparison of online social relations in column vs interaction: a case study of cyworld. In *Proceeding of the 8th Internet Measurement Conference (IMC'08)*, 2008.
- [19] B. Cohen. Incentives build robustness in BitTorrent. In *the 1st Workshop on Economics of Peer-to-Peer Systems (P2PECON'03)*, Berkeley, USA, 2003.
- [20] N.S. Cohen. The valorization of surveillance: Towards a political economy of Facebook. *Democratic Communiqué*, 22(1):5–22, 2007.
- [21] L. D'Acunto, N. Andrade, J.A. Pouwelse, and H.J. Sips. Peer selection strategies for improved QoS in heterogeneous BitTorrent-like VoD systems. In *Proceedings of the IEEE International Symposium on Multimedia (ISM'10)*, 2010.
- [22] L. D'Acunto, T. Vinkó, and H.J. Sips. Bandwidth allocation in BitTorrent-like vod systems under flashcrowds. In *Proceedings of the 11th IEEE International Conference on Peer-to-Peer Computing (P2P'11)*, 2011.
- [23] Dalesa. <http://www.dalesa.lk>.
- [24] DAS4. <http://www.cs.vu.nl/das4/>.
- [25] R. Delaviz, N. Andrade, and J.A. Pouwelse. Improving accuracy and coverage in an Internet-deployed reputation mechanism. In *Proceedings of the 10th IEEE International Conference on Peer-to-Peer Computing (P2P'10)*, 2010.
- [26] R. Delaviz, N. Andrade, J.A. Pouwelse, and D.H.J. Epema. Sybilres: a sybil-resilient flow-based decentralized reputation mechanism. In *Proceedings of the 32th IEEE International Conference on Distributed Computing Systems (ICDCS'12)*, 2012.

- 
- [27] P. Dhungel, D. Wu, B. Schonhorst, and K.W. Ross. A measurement study of attacks on BitTorrent leechers. In *the 7th International Workshop on Peer-to-Peer Systems (IPTPS'08)*, 2008.
- [28] J. Douceur. The sybil attack. In *the 1st International Workshop on Peer-to-Peer Systems (IPTPS'02)*, 2002.
- [29] N.B. Ellison and C. Steinfield and C. Lampe. The benefits of Facebook friends: Social capital and college students' use of online social network sites. *Journal of Computer-Mediated Communication*, 12(4):1143–1168, 2007.
- [30] B. Fan, D.M. Chiu, and J.C. Lui. The delicate tradeoffs in BitTorrent-like file sharing protocol design. In *Proceeding of the 14th IEEE International Conference on Network Protocols (ICNP'06)*, 2006.
- [31] B.M. Friedman and F.H. Hahn. *Handbook of Monetary Economics*. North Holland, 1990.
- [32] M. Friedman. *Capitalism and Freedom: Fortieth Anniversary Edition*. University of Chicago Press, 2002.
- [33] H. Gao, J. Hu, C. Wilson, Z. Li, Y. Chen, and B.Y. Zhao. Detecting and characterizing social spam campaigns. In *Proceedings of the 10th Internet Measurement Conference (IMC'10)*, 2010.
- [34] D. Gkorou, T. Vinkó, N. Chiluka, F.R. Santos, J.A. Pouwelse, and D.H.J. Epema. Reducing the history in decentralized interaction-based reputation systems. In *Proceedings of IFIP Networking*, 2012.
- [35] J. Golbeck and J.A. Hendler. Inferring binary trust relationships in web-based social networks. *ACM Transactions on Internet Technology*, 6(4):497-529, 2006.
- [36] L. Guo, S. Chen, Z. Xiao, E. Tan, X. Ding, and X. Zhang. Measurements, analysis, and modeling of BitTorrent-like systems. In *Proceedings of the 5th Internet Measurement Conference (IMC'05)*, 2005.
- [37] D. Hales, R. Rahman, B. Zhang, M. Meulpolder, and J.A. Pouwelse. BitTorrent or BitCrunch: Evidence of a credit squeeze in BitTorrent? In *the 18th IEEE International Workshops on Enabling Technologies: Infrastructures for Collaborative Enterprises (WETICE'09)*, 2009.
- [38] G. Hardin. Tragedy of the commons. *Science*, 162:1243–1248, 1968.

- [39] D. Hausheer, N.C. Liebau, A. Mauthe, R. Steinmetz, and B. Stiller. Token-based accounting and distributed pricing to introduce market mechanisms in a peer-to-peer file sharing scenario. In *Proceedings of the 3rd International Conference on Peer-to-Peer Computing (P2P'03)*.
- [40] HDChina. <http://hdchina.org/>.
- [41] HDStar. <http://www.hdstar.org/>.
- [42] O. Heckmann and A. Bock. The eDonkey2000 protocol. Technical Report 08/2002, Darmstadt University of Technology, 2002.
- [43] J. Ioannidis, S. Ioannidis, A. D. Keromytis, and V. Prevelakis. Fileteller: Paying and getting paid for file storage. In *Proceedings of the 6th International Conference on Financial Cryptography, 2002*.
- [44] B.A. Jacob. Accountability, incentives and behavior: The impact of high-stakes testing in the chicago public schools. *Journal of public Economics*, 89(5):761–796, 2005.
- [45] A. L. Jia, X. Chen, X. Chu, and J.A. Pouwelse. From user experience to strategies: how to survive in a private community. Technical Report PDS-2011-004, Delft University of Technology, 2011.
- [46] A.L. Jia, X. Chen, X. Chu, J.A. Pouwelse, and D.H.J. Epema. How to survive and thrive in a private BitTorrent community. In *the 14th International Conference on Distributed Computing and Networking (ICDCN'13)*, Jan 2013.
- [47] A.L. Jia, L. D'Acunto, M. Meulpolder, and J.A. Pouwelse. Modeling and analysis of sharing ratio enforcement in private BitTorrent networks. In *Proceedings of the IEEE International Communications Conference (ICC'11)*, 2011.
- [48] A.L. Jia, L. D'Acunto, M. Meulpolder, J.A. Pouwelse, and D.H.J. Epema. BitTorrent's dilemma: Enhancing reciprocity or reducing inequity. In *Proceedings of the 7th IEEE Consumer Communications and Networking Conference (CCNC'11)*, 2011.
- [49] A.L. Jia, R. Rahman, T. Vinkó, J.A. Pouwelse, and D.H.J. Epema. Fast download but eternal seeding: The reward and punishment of sharing ratio enforcement. In *Proceedings of the 11th International Conference on Peer-to-Peer Computing (P2P'11)*.
- [50] A.L. Jia, R. Rahman, T. Vinkó, J.A. Pouwelse, and D.H.J. Epema. User behaviors in private BitTorrent communities. *Submitted to Computer Networks*.

- 
- [51] A.L. Jia, R. Rahman, T. Vinkó, J.A. Pouwelse, and D.H.J. Epema. Systemic risk and user-level performance in private P2P communities. *IEEE Transactions on Parallel and Distributed Systems*, Available online, 2012.
- [52] A.L. Jia, B. Schoon, J.A. Pouwelse, and D.H.J. Epema. Estimating user interaction strength in online networks. In *Submitted to INFOCOM 2014*.
- [53] J.J.D. Mol, J.A. Pouwelse, M. Meulpolder, D.H.J. Epema and H.J. Sips. Give-to-Get: Free-riding-resilient Video-on-Demand in P2P systems. In *Proceeding of Multimedia Computing and Networking (MMCN'08)*, 2008.
- [54] I. Kahanda and J. Neville. Using transactional information to predict link strength in online social networks. In *Proceeding of the 3rd International AAI Conference on Weblogs and Social Media (ICWSM'09)*.
- [55] S.D. Kamvar, M.T. Schlosser, and H. Garcia-Molina. The Eigentrust algorithm for reputation management in P2P networks. In *Proceeding of the 12th International World Wide Web Conference (WWW'03)*.
- [56] I.A. Kash, E.J. Friedman, and J.Y. Halpern. Optimizing scrip systems: Efficiency, crashes, hoarders, and altruists. In *Proceeding of the 8th ACM Conference on Electronic Commerce (EC'07)*, 2007.
- [57] I.A. Kash, J.K. Lai, H. Zhang, and A. Zohar. Economics of BitTorrent communities. In *the 6th Workshop on the Economics of Networks, Systems, and Computation (NetEcon'11)*, 2011.
- [58] G. Kaufman. Banking and currency crises and systemic risk: A taxonomy and review. *Financial Markets, Institutions and Instruments*, 9(2):69-131, 2000.
- [59] Kazaa. <http://www.kazaa.com>.
- [60] R. Landa, D. Griffin, R.G. Clegg, E. Mykoniati, and M. Rio. A sybilproof indirect reciprocity mechanism for Peer-to-Peer networks. In *Proceedings of the 29th IEEE International Conference on Computer Communications (INFOCOM'09)*, 2009.
- [61] D. Lazer, A. Pentland, L. Adami, S. Aral, A. Barabasi, D. Brewer, N. Christakis, N. Contractor, J. Fowler, M. Gutmann, T. Jebara, G. King, M. Macy, D. Roy, and M. Van Alstyne. Computational social science. *Science*, 323:721–723, 2009.
- [62] D. Levin, K. Lacurts, N. Spring, and B. Bhattacharjee. BitTorrent is an auction: Analyzing and improving BitTorrent's incentives. In *Proceeding of ACM Special Interest Group on Data Communication (SIGCOMM'08)*, 2008.

- [63] D. Lewis. *Louder voices: The corporate welfare bums*. Lorimer, 1972.
- [64] Z. Liu, P. Dhungel, D. Wu, C. Zhang, and K.W. Ross. Understanding and improving incentives in private P2P communities. In *Proceedings of the 30th International Conference on Distributed Computing Systems (ICDCS'10)*, 2010.
- [65] T. Locher, P. Moor, S. Schmid, and R. Wattenhofer. Free riding in BitTorrent is cheap. In *the 5th Workshop on Hot Topics in Networks (HotNets'06)*, 2006.
- [66] M. Sirivianos, J. H. Park, R. Chen and X. Yang. Free-riding in BitTorrent networks with the large view exploit. In *the 6th International Workshop on Peer-to-Peer Systems (IPTPS'07)*, 2007.
- [67] P.V. Marsden. Core discussion networks of Americans. *American Sociological Review*, 52(1):122–131, 1987.
- [68] M. Medo, G. Cimini, and S. Gualdi. Temporal effects in the growth of networks. *Physical Review Letters*, 107(23):87–101, 2011.
- [69] M. Meulpolder, L. D'Acunto, M. Capotă, M. Wojciechowski, J.A. Pouwelse, D.H.J. Epema, and H.J. Sips. Public and private BitTorrent communities: A measurement study. In *the 9th International Workshop on Peer-to-Peer Systems (IPTPS'10)*, 2010.
- [70] M. Meulpolder, L.E. Meester, and D.H.J. Epema. The problem of upload competition in Peer-to-Peer systems with incentive mechanisms. *Concurrency and Computation: Practice and Experience.*, Available online, 2012.
- [71] M. Meulpolder, J.A. Pouwelse, D.H.J. Epema, and H.J. Sips. Bartercast: A practical approach to prevent lazy freeriding in P2P networks. In *the 6th International Workshop on Hot Topics in Peer-to-Peer Systems (HotP2P'09)*, 2009.
- [72] M. Meulpolder, J.A. Pouwelse, D.H.J. Epema, and H.J. Sips. Modeling and analysis of bandwidth-inhomogeneous swarms in BitTorrent. In *9th International Conference on Peer-to-Peer Systems (P2P'09)*, 2009.
- [73] T. Moreton and A. Twigg. Trading in trust, tokens, and stamps. In *the 1st Workshop on Economics of Peer-to-Peer Systems (P2PECON'03)*.
- [74] R. Motwani and P. Raghavan. *Randomized Algorithms*. Cambridge University Press, 1995.
- [75] Napster. <http://www.napster.com>.



- 
- [76] A. O'Sullivan and S.M. Sheffrin. *Economics: Principles in Action*. Pearson Prentice Hall, 2003.
- [77] R. Petrocco, J.A. Pouwelse, and D.H.J. Epema. Performance analysis of the Libswift P2P streaming protocol. In *Proceedings of the 12th IEEE International Conference on Peer-to-Peer Computing (P2P'12)*, 2012.
- [78] M. Piatek, T. Isdal, T. Anderson, A. Krishnamurthy, and A. Venkataramani. Do incentives build robustness in BitTorrent? In *Proceedings of the 4th USENIX Symposium on Networked Systems Design and Implementation (NSDI'07)*, 2007.
- [79] M. Piatek, T. Isdal, A. Krishnamurthy, and T. 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'08)*, 2008.
- [80] The PirateBay. <http://www.thepiratebay.se>.
- [81] PolishTracker. <http://polishtracker.net/>.
- [82] W. Poundstone. *Prisoner's Dilemma*. Doubleday, January 1992.
- [83] J.A. Pouwelse, P. Garbacki, D.H.J. Epema, and H.J. Sips. The BitTorrent P2P file-sharing system: Measurements and analysis. In *the 4th International Workshop on Peer-to-Peer Systems (IPTPS'05)*, 2005.
- [84] J.A. Pouwelse, P. Garbacki, J. Wang, A. Bakker, J. Yang, A. Iosup, D.H.J. Epema, M. Reinders, M. van Steen, and H.J. Sips. Tribler: A social-based Peer-to-Peer system. *Concurrency and Computation: Practice and Experience*, 20:127–138, 2008.
- [85] D. Qiu and R. Srikant. Modeling and performance analysis of BitTorrent-like Peer-to-Peer networks. In *Proceeding of ACM Special Interest Group on Data Communication (SIGCOMM'04)*, 2004.
- [86] Qlectives. <http://www.qlectives.eu/>.
- [87] Y. Qu, G.Y. Keung, C. Lin, J. Liu, B. Li, S. Xie, and X. Zhang. Inside the new Coolstreaming: Principles, measurements and performance implications. In *Proceedings of the 28th IEEE International Conference on Computer Communications (INFOCOM'08)*, 2008.
- [88] J. Raacke and J. Bonds-Raacke. MySpace and Facebook: Applying the uses and gratifications theory to exploring friend-networking sites. *CyberPsychology & Behavior*, 11:169–174, 2008.

- [89] R. Rahman, D. Hales, T. Vinkó, J.A. Pouwelse, and H.J. Sips. No more crash or crunch: Sustainable credit dynamics in a P2P community. In *Proceeding of International Conference on High Performance Computing & Simulation (HPCS'10)*, 2010.
- [90] R. Rahman, M. Meulpolder, D. Hales, J.A. Pouwelse, D.H.J. Epema, and H.J. Sips. Improving efficiency and fairness in P2P systems with effort-based incentives. In *Proceedings of the IEEE International Communications Conference (ICC'10)*, 2010.
- [91] R. Rahman, T. Vinkó, D. Hales, J.A. Pouwelse, and H.J. Sips. Design space analysis for modelling incentives in distributed systems. In *Proceedings of ACM Special Interest Group on Data Communication (SIGCOMM'11)*, 2011.
- [92] C.P. Roca, C. Efferson, S. Vogt, and D. Helbing. Group choice with limited information allows cooperation to persist. *Submitted to Science*.
- [93] C.P. Roca and D. Helbing. Emergence of social cohesion in a model society of greedy, mobile individuals. *Proceedings of the National Academy of Sciences (PNAS)*, 108:11370–11374, 2011.
- [94] J.L. Rodgers and W.A. Nicewander. Thirteen ways to look at the correlation coefficient. *The American Statistician*, 42(1):59-66, 1988.
- [95] S. Saroiu, P.K. Gummadi, and S.D. Gribble. A measurement study of peer-to-peer file sharing systems. In *Proceedings of the Multimedia Computing and Networking (MMCN'02)*, 2002.
- [96] S. Saroiu, P.K. Gummadi, and S.D. Gribble. Measuring and analysis the characteristics of Napster and Gnutella hosts. *Multimedia Systems*, 9:170–184, 2003.
- [97] R. Snader and N. Borisov. EigenSpeed: Secure Peer-to-peer Bandwidth Evaluation. In *the 8th International Workshop on Peer-to-Peer Systems (IPTPS'09)*, 2009.
- [98] G. Soros. The worst market crisis in 60 years. *Financial Times*, January 22, 2008.
- [99] C. Spearman. The proof and measurement of association between two things. *American Journal of Psychology*, 15:72-101, 1904.
- [100] Spotify. A world of music. <http://www.spotify.com>.
- [101] K. Thomas, C. Grier, D. Song, and V. Paxso. Uncovering social network sybils in the wild. In *Proceedings of the 11th Internet Measurement Conference (IMC'11)*, 2011.

- 
- [102] N. Tran, J. Li, L. Subramanian, and S. S. Chow. Optimal sybil-resilient node admission control. In *Proceedings of the 31th IEEE International Conference on Computer Communications (INFOCOM'11)*, 2011.
- [103] N. Tran, B. Min, J. Li, and L. Subramanian. Sybil-resilient online content voting. In *Proceeding of the 6th USENIX Symposium on Networked Systems Design and Implementation (NSDI'09)*, 2009.
- [104] TVTorrents. <http://www.tvtorrents.com/>.
- [105] uTorrent. <http://www.utorrent.com>.
- [106] A. Vespignani. The familiar stranger: An aspect of urban anonymity. *The Individual in a Social World: Essays and Experiments*, 1977.
- [107] V. Vishnumurthy, S. Chandrakumar, and E.G. Sirer. Karma: A secure economic framework for peer-to-peer resource sharing. In *the 1st Workshop on Economics of Peer-to-Peer Systems (P2PECON'03)*, 2003.
- [108] B. Viswanath, A. Mislove, M. Cha, and K. P. Gummadi. On the evolution of user interaction in Facebook. In *the Second ACM SIGCOMM Workshop on Online Social Networks (WOSN'09)*, 2009.
- [109] B. Viswanath and A. Post. An analysis of social network-based sybil defenses. In *Proceeding of ACM Special Interest Group on Data Communication (SIGCOMM'10)*, 2010.
- [110] A. Vlavianos, M. Iliofotou, and M. Faloutsos. BiToS: Enhancing BitTorrent for supporting streaming applications. In *Proceedings the 9th IEEE Global Internet Symposium (GIS'06)*, 2006.
- [111] Vuze. <http://www.vuze.com>.
- [112] W. C. Liao, F. Papadopoulos and K. Psounis. Performance analysis of BitTorrent-like systems with heterogeneous users. In *Proceedings of the 26th International Symposium on Computer Performance, Modeling, Measurements, and Evaluation (PERFORMANCE'07)*, 2007.
- [113] D. Watts. A twenty-first century science. *Nature*, 445:489, 2007.
- [114] C. Wilson, B. Boe, A. Sala, K. P. N. Puttaswamy, and B. Y. Zhao. User interactions in social networks and their implications. In *Proceeding of EuroSys*, 2009.
- [115] J. Wu, S. Lozano, and D. Helbing. Empirical study of the growth dynamics in real career hindex sequences. *Journal of Informetrics*, 5:489–497, 2011.

- [116] R. Xiang, J. Neville, and M. Rogati. Modeling relationship strength in online social networks. In *Proceeding of the 19th International World Wide Web Conference (WWW'10)*, 2010.
- [117] Y. Tian, D. Wu and K. W. Ng. Modeling, analysis and improvement for BitTorrent-like file sharing networks. In *Proceedings of the 26th IEEE International Conference on Computer Communications (INFOCOM'06)*, 2006.
- [118] Yacy. <http://www.yacy.net>.
- [119] Z. Yang, C. Wilson, X. Wang, T. Gao, B.Y. Zhao, and Y. Dai. Uncovering social network sybils in the wild. In *Proceedings of the 11th Internet Measurement Conference (IMC'11)*, 2011.
- [120] H. Yu, P. B. Gibbons, M. Kaminsky, and F. Xiao. Sybillimit: A near-optimal social network defense against sybil attacks. In *IEEE Symposium on Security and Privacy*, 2008.
- [121] H. Yu, M. Kaminsky, P.B. Gibbons, and A. Flaxman. Sybilguard: Defending against sybil attacks via social networks. In *Proceeding of ACM Special Interest Group on Data Communication (SIGCOMM'06)*, 2006.
- [122] N. Zeilemaker, M. Capotă, A. Bakker, and J.A. Pouwelse. Tribler: P2P media search and sharing. In *Poster presented at ACM Multimedia*, 2011.
- [123] N. Zeilemaker, B. Schoon, and J.A. Pouwelse. Dispersy bundle synchronization. Technical Report PDS-2013-002, Delft University of Technology, 2013.
- [124] C. Zhang, P. Dhungel, Z. Liu Di Wu, and K.W. Ross. BitTorrent darknets. In *Proceedings of the 30th IEEE International Conference on Computer Communications (INFOCOM'10)*, 2010.

# Summary

## Online networks as societies: user behaviors and contribution incentives

Online networks like Facebook and BitTorrent have become popular and powerful infrastructures for users to communicate, to interact, and to share social lives with each other. These networks often rely on the cooperation and the contribution of their users. Nevertheless, users in online networks are often found to be selfish, lazy, or even malicious, rather than cooperative, and therefore need to be incentivized for contributions. To date, great effort has been put into designing effective contribution incentive policies, which range from *barter schemes* to *monetary schemes*. In this thesis, we conduct an analysis of user behaviors and contribution incentives in online networks. We approach online networks as both computer systems and societies, hoping that this approach will, on the one hand, motivate computer scientists to think about the similarities between their artificial computer systems and the natural world, and on the other hand, help people outside the field understand online networks more smoothly.

In Chapter 2, we investigate the characteristics and principles of BitTorrent's Tit-For-Tat incentive policy and its variations that are based on barter. We propose a fluid model that captures the bandwidth allocation in BitTorrent. We explore several strategies that influence the balance between reciprocity and inequality. Our study shows that (i) reducing inequality leads to a better overall system performance, and (ii) the behavior of seeders (i.e., users that hold a complete copy of the file and upload it for free) influences whether reciprocity is enhanced or inequality is reduced.

In Chapter 3, we provide a theoretical model to analyze a contribution incentive named Sharing Ratio Enforcement (SRE). We aim to provide an understanding of how SRE provides contribution incentives and how SRE influences the download performance in the system. Specifically, we study the influence of the SRE threshold (i.e., the minimum sharing ratio requirement) and the bandwidth heterogeneity of users in the system. In our analysis, we assume users to be rational, i.e., they seed only the minimum amount required by SRE, and we show that the download performance as predicted by our model represents a lower bound for the actual performance that can be reached in a BitTorrent private community. Hence, following our model, community administrators can predict the minimum performance level their systems will be able to reach.

In Chapter 4, we analyze the performance of online networks that adopt monetary

policies as contribution incentives, from both the system-level and the user-level perspectives. Taking private BitTorrent communities as an example, we show that monetary policies can lead to system-wide *crunches* or *crashes* where the system completely seizes due to too little or to too much credit, respectively. We explore the conditions that lead to these system pathologies and we present a theoretical model that predicts if a system will eventually crunch or crash. We apply this analysis to design an adaptive monetary policy to maintain the system sustainability. Given private communities that are sustainable, we further analyze their user-level performance by studying the effects of their oversupply, in terms of excessively high seeder-to-leecher ratios. We show that although achieving an increase in the average downloading speed, the phenomenon of oversupply has three undesired effects: long seeding times, low upload capacity utilizations, and an unfair playing field for late entrants into swarms. To tackle these problems, we propose four different strategies, which have been inspired by ideas in social sciences and economics. We evaluate these strategies through simulations and demonstrate their positive effects.

In Chapter 5, we explore the user behaviors in private BitTorrent communities, we argue the reasons for these behaviors, and we demonstrate both the positive and the negative effects of these behaviors. We show that in these communities, as predicted by our model, users seed for excessively long times to maintain required sharing ratios, but that their seedings are often not very productive (in terms of low upload speeds) and that their long seeding times do not necessarily lead to large upload amounts. We find that as users evolve in the community, some users become more committed, in terms of increasing ratios between their seeding and leeching times. In the mean time, some users game the system by keeping risky and low sharing ratios while leeching more often than seeding. Based on these observations, we analyze strategies that alleviate the negative effects of these user behaviors from both the user's and the community administrator's perspective.

In Chapter 6, we present the design, deployment, and analysis of the UISE scheme for User Interaction Strength Estimation for both centralized and decentralized online networks. Among the strong points of UISE is that it captures direct and indirect user interactions, that it scales with only partial information dissemination in decentralized systems, and that it provides disincentives for malicious user behaviors. We apply UISE to detect user interaction patterns based on wall posts in Facebook, and our results resemble the patterns often observed in the offline human society. We further apply UISE to online time estimation based on rendezvous as user interactions in Tribler, an online network for media and social applications like file sharing, streaming, and voting. We demonstrate the accuracy and scalability of UISE with different information dissemination protocols and user behaviors using simulations, emulations, and a real-world deployment.

To summarize, in this thesis we provide theoretical and practical insights into the correlation between user behaviors and contribution incentives in online networks. We demonstrate user behaviors and their consequences at both the system and the individual

level, we analyze barter schemes and their limitations in incentivizing users to contribute, we evaluate monetary schemes and their risks in causing the collapse of the entire system, and we examine user interactions and their implications in inferring user relationships. Above all, unlike the offline human society that has evolved for thousands of years, online networks only emerged two decades ago and are still in a primitive state. Yet with their ever-improving technologies we have already obtained many exciting results. This points the way to a promising future for the study of online networks, not only in analyzing online behaviors, but also in cross reference with offline societies.





# Samenvatting

## Online networks as societies: user behaviors and contribution incentives

*Online* netwerken zoals Facebook en BitTorrent zijn populaire en krachtige infrastructures geworden voor mensen die met elkaar willen communiceren, contact willen hebben, en hun sociale leven willen delen. Deze netwerken zijn vaak afhankelijk van de samenwerking en de bijdragen van hun gebruikers. Echter, in plaats van samenwerkend zijn gebruikers in *online* netwerken vaak egoïstisch, lui of zelfs kwaadaardig, en daarom moeten zij geprikkeld worden om bijdragen te leveren. Er is al veel moeite gestoken in het ontwerpen van effectieve *policies* om tot bijdragen te prikkelen, die lopen van *ruilhandel* tot *monetaire* systemen. In dit proefschrift voeren we een analyse uit van gebruikersgedrag en prikkels tot bijdragen in *online* netwerken. We benaderen *online* netwerken als computersystemen en als gemeenschappen, hopend dat deze benadering aan de ene kant informatici kan motiveren om na te denken over de overeenkomsten tussen kunstmatige computersystemen en de werkelijke wereld, en aan de andere kant mensen buiten dit onderzoeksveld gemakkelijker *online* netwerken laat begrijpen.

In hoofdstuk 2 onderzoeken we de karakteristieken en principes van BitTorrent's Tit-For-Tat mechanisme en variaties daarop die op ruilhandel gebaseerd zijn. We stellen een vloeiend model voor waarmee de allocatie van bandbreedte in BitTorrent wordt beschreven. We onderzoeken verschillende strategieën die de balans tussen wederkerigheid en ongelijkheid beïnvloeden. Onze studie toont aan dat (i) het verminderen van ongelijkheid leidt tot betere algehele systeemprestaties, en (ii) het gedrag van *seeders* (d.w.z. gebruikers die een volledige kopie van het bestand hebben en dat gratis weggeven) bepaalt of de wederkerigheid wordt versterkt of de ongelijkheid wordt gereduceerd.

In hoofdstuk 3 presenteren we een theoretisch model voor het analyseren van *Sharing Ratio Enforcement* (SRE) als prikkel tot bijdragen. Ons doel is om duidelijk te maken hoe SRE dergelijke prikkels levert en hoe SRE de *download*-prestaties in het systeem beïnvloedt. Meer in het bijzonder onderzoeken we de invloed van de SRE-drempel (d.w.z. de minimum vereiste *sharing ratio*) en van de heterogeniteit van de bandbreedte van gebruikers in het systeem. In onze analyse gaan we uit van rationele gebruikers, wat wil zeggen dat ze slechts de minimum-bijdrage geven die SRE vereist, en we laten zien dat de *download*-prestaties zoals voorspeld door ons model een ondergrens vormen voor de werkelijke prestaties die in een gesloten BitTorrent-gemeenschap bereikt kunnen worden.

Derhalve kunnen de beheerders van een gemeenschap op grond van ons model het minimale prestatieniveau van hun systeem voorspellen.

In hoofdstuk 4 analyseren we de prestaties van *online* netwerken die monetaire prikkels tot bijdragen gebruiken, vanuit zowel systeem- als gebruikersperspectief. Aan de hand van gesloten BitTorrent-gemeenschappen laten we zien dat monetaire prikkels kunnen leiden tot systeemwijde *crunches* of *crashes* waarbij het systeem volledig vastloopt door te weinig respectievelijk te veel krediet. We onderzoeken de omstandigheden die leiden tot deze systeempathologieën en presenteren een theoretisch model dat voorspelt of een systeem uiteindelijk zal vastlopen. We passen deze analyse toe bij het ontwerp van een adaptief monetair mechanisme om de bestendigheid van het systeem te garanderen. Voor gesloten gemeenschappen die bestendig zijn analyseren we hun prestaties voor gebruikers door de effecten van hun overaanbod in termen van uitzonderlijk hoge *seeder-leecher* verhoudingen te bestuderen. We tonen aan dat hoewel een stijging van de gemiddelde *download*-snelheid wordt bereikt, het fenomeen van overaanbod drie ongewenste effecten heeft: lange *seeding*-tijden, laag gebruik van *upload*-capaciteit en een ongelijk speelveld voor de late toetreders tot het systeem. Om deze problemen aan te pakken, stellen we vier verschillende strategieën voor die zijn geïnspireerd door ideeën uit de sociale wetenschappen en economie. We evalueren deze strategieën door middel van simulaties en demonstreren hun positieve effecten.

In hoofdstuk 5 verkennen we het gedrag van gebruikers in gesloten BitTorrent-gemeenschappen, gaan we de redenen voor dit gedrag na, en demonstreren we zowel de positieve als de negatieve gevolgen van dit gedrag. We laten zien dat in deze gemeenschappen, zoals voorspeld door ons model, gebruikers zeer lange tijd *seed*-tijden hebben om de benodigde *sharing ratio* te behouden, maar dat deze tijd vaak niet erg productief gebruikt wordt (in termen van lage *upload*-snelheden) en dat de lange *seed*-tijden niet noodzakelijkerwijs leiden tot grote hoeveelheden ontvangen data. Onze bevinding is dat naarmate gebruikers evolueren in een gemeenschap, sommige van hen meer toegewijd worden in termen van het verhogen van de verhouding tussen hun *seed*- en *leech*-tijd. Tegelijkertijd zijn er ook gebruikers die het systeem bespelen door een risicovolle, lage *sharing ratio* te onderhouden door relatief meer aan *leeching* dan aan *seeding* te doen. Gebaseerd op deze observaties analyseren we strategieën die de negatieve effecten van dit gebruikersgedrag verlichten vanuit het perspectief van zowel de gebruiker als de beheerder van de gemeenschap.

In Hoofdstuk 6 presenteren we het ontwerp, de implementatie en de analyse van UISE voor *User Interaction Strength Estimation* voor zowel gecentraliseerde als gedecentraliseerde *online* netwerken. De sterke punten van UISE zijn dat het directe en indirecte interacties tussen gebruikers vangt, dat het schaalbaar is met slechts gedeeltelijke informatieverspreiding in gedecentraliseerde systemen, en dat het kwaadwillende gebruikers ontmoedigt. We passen UISE toe om patronen van gebruikersinteracties te detecteren op

basis van *wall-posts* in Facebook, en de resultaten lijken op vaak waargenomen patronen in de *offline* menselijke samenleving. Verder passen we UISE toe op het schatten van de *online* tijden op basis van rendez-vous als gebruikersinteracties in Tribler, een *online* netwerk voor media- en sociale toepassingen zoals *file sharing*, *streaming*, en stemmen. Met behulp van simulaties, emulaties, en door het in het Internet toe te passen tonen we de nauwkeurigheid en schaalbaarheid van UISE aan onder verschillende protocollen voor informatieverspreiding en verschillend gebruikersgedrag.

Samenvattend verschaffen we in dit proefschrift theoretische en praktische inzichten in de relatie tussen gebruikersgedrag en prikkels tot bijdragen in *online* netwerken. We presenteren gebruikersgedrag en de consequenties daarvan zowel op systeem- als individueel niveau, we analyseren ruilhandelmechanismen en hun beperkingen om gebruikers te prikkelen tot bijdragen, we evalueren monetaire mechanismen en hun risico's om de ineenstorting van het gehele systeem te veroorzaken, en we onderzoeken gebruikersinteracties en hun implicaties bij het afleiden van de relaties tussen gebruikers. Bovenal zijn *online* netwerken pas gedurende de laatste twee decennia opgekomen en zijn ze nog in een primitieve staat, in tegenstelling tot de *offline* menselijke samenleving die zich gedurende duizenden jaren ontwikkeld heeft. Toch hebben we met hun aldoor verbeterende technologieën al veel opwindende resultaten bereikt. Dit wijst de weg naar een veelbelovende toekomst voor de studie van *online* netwerken, niet alleen wat betreft het analyseren van *online* gedrag, maar ook in vergelijking met *offline* gemeenschappen.



# Biography

Lu Jia (Adele), was born in Harbin, China, on March 7th, 1983. After she graduated from Harbin Institute of Technology with a BSc and an MSc degree in Communication Engineering, in 2007 she attended The Chinese University of Hong Kong (CUHK) for an MPhil degree in Information Engineering. During her stay in CUHK, she worked and earned an early enlightenment on modeling and analysis of Peer-to-Peer systems, after which she joined the Parallel & Distributed Systems Group of Delft University of Technology in 2009 as a researcher and a PhD candidate. The research carried out between 2009 to 2013 has eventually led to this thesis.

Adele is currently looking for a research position in her home country. Her free time is dedicated to her family and friends and a number of avocations including badminton and horseback riding. Along with this thesis writing that began in April, 2013, she started and maintained a meditation-like project of one-hour piano everyday, to resume a childhood routine. Above all, she wishes to explore the world and the mind with her own vision and tenderness.

## Teaching Experience

Teaching Assistant at *Delft University of Technology*, The Netherlands.

### Course

Seminar Peer-to-Peer Systems

### Responsibility

Presentation Assistance, Grading

Teaching Assistant at *The Chinese University of Hong Kong*, Hong Kong.

### Course

Engineering Mathematics

Computer Networks

### Responsibility

Exercise lesson, Grading

Exercise lesson, Grading

## Review Experience

### Journals

- *IEEE Transactions on Parallel and Distributed Systems*
- *Computer Networks*
- *Computer Communications*
- *Journal of Network and Computer Applications*

## Publications

### Journals

- A.L. Jia, R. Rahman, T. Vinkó, J.A. Pouwelse, and D.H.J. Epema. Systemic risk and user-level performance in private P2P communities. In *IEEE Transactions on Parallel and Distributed Systems*. Available online, 2012.
- A.L. Jia, X. Chen, X. Chu, J.A. Pouwelse, and D.H.J. Epema. User behaviors in private BitTorrent communities. *Under review*.
- X. Chen, X. Chu, A.L. Jia, and J.A. Pouwelse. Dissecting darknets: Measurement and performance analysis. *Under review*.

### Conference Proceedings

- A.L. Jia, B. Schoon, J.A. Pouwelse, and D.H.J. Epema. Estimating user interaction strength in online networks. *Under review*.
- A.L. Jia, X. Chen, X. Chu, J.A. Pouwelse, and D.H.J. Epema. How to survive and thrive in a private BitTorrent community. In *The 14th International Conference on Distributed Computing and Networking (ICDCN'13)*, 2013.
- X. Chen, X. Chu, A.L. Jia, and J.A. Pouwelse. Inequity of Sharing Ratio Enhancement in Darknet: Measurement and improvement. In *Proceedings of the 14th IEEE International Conference on High Performance Computing and Communications (HPCC'12)*, 2012.
- A.L. Jia, R. Rahman, T. Vinkó, J.A. Pouwelse, and D.H.J. Epema. Fast download but eternal seeding: The reward and punishment of sharing ratio enforcement. In *Proceedings of the 11th International Conference on Peer-to-Peer Computing (P2P'11)*, 2011.

- A.L. Jia, R. Rahman, T. Vinkó, J.A. Pouwelse, and D.H.J. Epema. Sharing Ratio Enforcement: The ultimate solution for BitTorrent? In *Proceedings of ICT.Open 2011*. Best paper award.
- A.L. Jia, L. D'Acunto, M. Meulpolder, J.A. Pouwelse, and D.H.J. Epema. BitTorrent's dilemma: Enhancing reciprocity or reducing inequity. In *Proceedings of the 7th IEEE Consumer Communications and Networking Conference (CCNC'11)*, 2011.
- A.L. Jia, L. D'Acunto, M. Meulpolder, and J.A. Pouwelse. Modeling and analysis of Sharing Ratio Enforcement in private BitTorrent communities. In *Proceedings of the IEEE International Communications Conference (ICC'11)*, 2011.
- A.L. Jia and D.M. Chiu. Designs and evaluation of a tracker in P2P networks. In *Proceedings of the 8th International Conference on Peer-to-Peer Computing (P2P'08)*, 2008.